

Finite-Difference Schemes for Reaction–Diffusion Equations Modeling Predator–Prey Interactions in MATLAB

Marcus R. Garvie

School of Computational Science, Florida State University, Tallahassee, FL 32306-4120, USA

Received: 9 August 2005 / Accepted: 6 December 2005 / Published online: 1 February 2007
© Society for Mathematical Biology 2007

Abstract We present two finite-difference algorithms for studying the dynamics of spatially extended predator–prey interactions with the Holling type II functional response and logistic growth of the prey. The algorithms are stable and convergent provided the time step is below a (non-restrictive) critical value. This is advantageous as it is well-known that the dynamics of approximations of differential equations (DEs) can differ significantly from that of the underlying DEs themselves. This is particularly important for the spatially extended systems that are studied in this paper as they display a wide spectrum of ecologically relevant behavior, including chaos. Furthermore, there are implementational advantages of the methods. For example, due to the structure of the resulting linear systems, standard direct, and iterative solvers are guaranteed to converge. We also present the results of numerical experiments in one and two space dimensions and illustrate the simplicity of the numerical methods with short programs in MATLAB. Users can download, edit, and run the codes from <http://www.uoguelph.ca/~mgarvie/>, to investigate the key dynamical properties of spatially extended predator–prey interactions.

Keywords Reaction-diffusion system · Predator-prey interaction · Finite difference method · MATLAB

1. Introduction

1.1. Model equations

In this paper, we study the numerical solutions of 2-component reaction–diffusion systems with the following general form (cf. May, 1974, p. 84; Murray, 1993, p. 71;

Sherratt, 2001)

$$\begin{cases} \frac{\partial u}{\partial t} = \delta_1 \Delta u + r u \left(1 - \frac{u}{w}\right) - p v h(ku), \\ \frac{\partial v}{\partial t} = \delta_2 \Delta v + q v h(ku) - s v, \end{cases} \tag{1}$$

where $u(\vec{x}, t)$ and $v(\vec{x}, t)$ are the population densities of prey and predators at time t and (vector) position \vec{x} . Δ is the usual Laplacian operator in $d \leq 3$ space dimensions and the parameters $\delta_1, \delta_2, r, w, p, k, q,$ and s are strictly positive. The ‘functional response’ $h(\cdot)$ is assumed to be a C^2 function satisfying the following conditions:

- (i) $h(0) = 0,$
- (ii) $\lim_{x \rightarrow \infty} h(x) = 1,$
- (iii) $h(\cdot)$ is strictly increasing on $[0, \infty).$

The functional response represents the prey consumption rate per predator as a fraction of the maximal consumption rate p . The constant k determines how fast the consumption rate saturates as the prey density increases. q and r denote maximal per capita birth rates of predator and prey, respectively, s is the per capita predator death rate, and w the prey-carrying capacity. In the above model, the local growth of the prey is logistic and the predator shows the ‘Holling type II functional response’ (Holling, 1959). Type II functional responses are the most frequently studied functional responses and well documented in empirical studies (for review, see Gentleman et al., 2003; Jeschke et al., 2002; Skalski and Gilliam, 2001). It is important to note that the above model takes into account the invasion of the prey species by predators but does not include stochastic effects or any influences from the environment. Nevertheless, reaction–diffusion equations modeling predator–prey interactions show a wide spectrum of ecologically relevant behavior resulting from intrinsic factors alone, and is an intensive area of research. For an introduction to research in the application of reaction–diffusion equations to population dynamics, see Holmes et al. (1994) and the references therein.

It is much easier to work with equations that have been scaled to nondimensional form. Thus, in (1), we take

$$\tilde{u} = \frac{u}{w}, \quad \tilde{v} = v \left(\frac{p}{r w}\right), \quad \tilde{t} = r t, \quad \tilde{x}_i = x_i \left(\frac{r}{\delta_1}\right)^{1/2},$$

and rescale the parameters via

$$a = k w, \quad b = \frac{q}{r}, \quad c = \frac{s}{r}, \quad \delta = \frac{\delta_2}{\delta_1}.$$

This leads to (after dropping the tildes) the nondimensional system:

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u + u(1 - u) - v h(au), \\ \frac{\partial v}{\partial t} = \delta \Delta v + b v h(au) - c v, \end{cases} \tag{2}$$

where the parameters $a, b, c,$ and δ are strictly positive. We assume the system is defined on a bounded domain (habitat), denoted by $\Omega,$ and augmented with appropriate initial conditions and the zero-flux boundary conditions. The particular choice of boundary conditions reflects the assumption that the individual species cannot leave the domain.

The aim of this paper is to present two stable finite-difference schemes for the numerical solution of (2) in one and two space dimensions and illustrate the simplicity of the numerical methods with short programs in MATLAB. A detailed numerical analysis of the model equations was undertaken by Garvie and Trenchea (2005b), and the current work provides the computational and implementational details needed to study the key dynamical properties of these equations. We also give the results of some numerical experiments that have ecological and numerical implications.

We focus on the following specific type II functional responses with positive parameters $\alpha, \beta,$ and γ

$$h(\eta) = h_1(\eta) = \frac{\eta}{1 + \eta} \quad (\eta = au), \quad \text{with } a = 1/\alpha, b = \beta, c = \gamma, \tag{3a}$$

$$h(\eta) = h_2(\eta) = 1 - e^{-\eta} \quad (\eta = au), \quad \text{with } a = \gamma, c = \beta, b = \alpha\beta, \tag{3b}$$

due originally to Holling (1965) and Ivlev (1961), respectively. Thus, the two types of kinetics covered by this work are

- (i) $f(u, v) = u(1 - u) - \frac{uv}{u + \alpha}, \quad g(u, v) = \frac{\beta uv}{u + \alpha} - \gamma v,$
- (ii) $f(u, v) = u(1 - u) - v(1 - e^{-\gamma u}), \quad g(u, v) = \beta v(\alpha - 1 - \alpha e^{-\gamma u}).$

In order to provide guidelines on the appropriate choice of parameters for numerical simulation of the full reaction–diffusion system, it is important to consider the local dynamics of the system (Medvinsky et al., 2002). It is, naturally, the dynamics in the biologically meaningful region $u \geq 0, v \geq 0$ that are of interest. By considering the ‘nullclines’ $f = 0, g = 0,$ and the intersection of these curves in phase space, linear stability analysis reveals that we have saddle points at $(0, 0)$ and $(1, 0)$ for both types of kinetics. Furthermore, there is a stationary point at (u^*, v^*) corresponding to the coexistence of prey and predators, given by

$$u^* = \frac{\alpha\gamma}{\beta - \gamma}, \quad v^* = (1 - u^*)(u^* + \alpha), \quad \text{with } \beta > \gamma, \quad \alpha < \frac{\beta - \gamma}{\gamma},$$

$$u^* = -\frac{1}{\gamma} \ln\left(\frac{\alpha - 1}{\alpha}\right), \quad v^* = \frac{u^*(1 - u^*)}{1 - e^{-\gamma u^*}}, \quad \text{with } \alpha > 1, \quad \gamma > -\ln\left(\frac{\alpha - 1}{\alpha}\right),$$

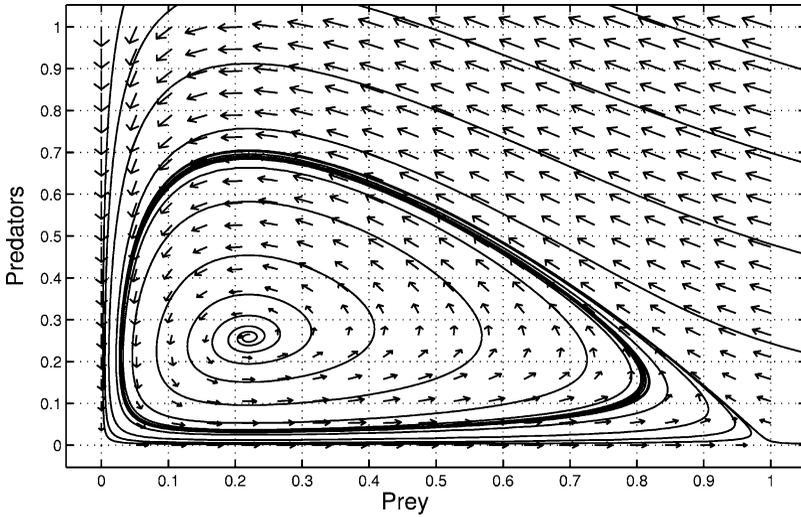


Fig. 1 Phase plane for 2 with Kinetics (ii). Parameter values: $\alpha = 1.5$, $\beta = 1.0$, $\gamma = 5.0$.

for Kinetics (i) and (ii), respectively. In order for the stationary point to be in the positive quadrant, we must have $0 < u^* < 1$, The result leads to the stated restrictions on the parameters (assumed throughout). Notice that in both cases $b > c$. For certain parameter choices (see Garvie and Trenchea, 2005a for additional details), the kinetics have a stable limit cycle surrounding the unstable stationary point (u^*, v^*) , i.e., the densities of predators and prey cycle periodically in time. An example of kinetics with a limit cycle is illustrated in Fig. 1.

1.2. Background

For the remainder of this section, we give a brief review of previous work on (2) with either the Holling or the Ivlev functional response. It was proved in Garvie and Trenchea (2005a) that the predator–prey systems are well posed mathematically, meaning that there is a unique solution for all time, depending continuously on the initial data. Furthermore, with bounded nonnegative initial data solutions remain nonnegative for all time. This is obviously necessary for any realistic model involving biological or chemical ‘species.’

The system of ordinary differential equations (ODEs), i.e., the spatially homogeneous system, corresponding to (2) has been well studied (Freedman, 1980; May, 1974; Murray, 1993). The ODE system corresponding to Kinetics (i) is sometimes called the Rosenzweig–MacArthur model (Rosenzweig and MacArthur, 1963), and has been used in many studies to fit ecological data. However, there are fewer papers, concerning the (‘spatially extended’) reaction–diffusion system, which takes into account both spatial and temporal dynamics of predators and prey.

A work that partly motivated this study is a SIAM Review paper (Medvinsky et al., 2002) that considers the reaction–diffusion system (2–3a) as a model for

marine plankton dynamics. The paper has an extensive reference list, and the authors numerically show that, for various initial conditions, the evolution of the system leads to the formation of spiral patterns, followed by irregular patches over the whole domain (spatiotemporal chaos), which are in qualitative agreement with field observations.

For additional studies of the reaction–diffusion systems considered here see [Malchow and Petrovskii \(2002\)](#), [Petrovskii and Malchow \(2001\)](#), [Petrovskii and Malchow \(1999\)](#), [Petrovskii and Malchow \(2002\)](#), [Sherratt et al. \(1997\)](#), [Sherratt et al. \(2002\)](#), [Sherratt et al. \(1995\)](#), [Alonso et al. \(2002\)](#), [Gurney et al. \(1998\)](#), [Rai and Jayaraman \(2003\)](#), and [Savill and Hogeweg \(1999\)](#).

In addition to works on spatiotemporal phenomena such as waves and chaos, there is a large group of papers on stationary spatial patterns in predator–prey systems. These arise via diffusion driven instability (discovered by Turing, [1952](#)) and rely on significant differences between predator and prey diffusion coefficients (see for example [Segel and Jackson, 1972](#), and the references in [Murray, 1993](#); [Neubert et al., 2002](#)). For two studies of diffusion-induced chaos for (2) with Kinetics (i) see [Pascual \(1993\)](#), [Rai and Jayaraman \(2003\)](#). In some situations, one can assume that the diffusion coefficients of predators and prey are equal, excluding the possibility of ‘Turing patterns.’ For example, this assumption is valid for plankton communities, where turbulent diffusivity is usually much greater than the diffusivity of the plankton species ([Medvinsky et al., 2002](#); [Petrovskii and Malchow, 2002](#)), but less valid for terrestrial communities, where the predator population is typically more motile than the prey population.

The remainder of the paper is organized in the following way. In Section 2, we present two semi-implicit (in time) finite-difference schemes for approximating the solutions of (2) in one and two space dimension. In Section 3, we present the results of numerical experiments in one and two dimensions, and sample MATLAB codes are described in Section 4. In Section 5, some concluding comments are made. The sample MATLAB codes are listed in Appendix D.

2. Numerical solution of the model equations

2.1. Motivation

Almost all of the realistic models in biology are nonlinear ([Murray, 1993](#)), without closed form solutions, thus numerical methods have an important part to play in investigating the behavior of their solutions. The rigorous numerical analysis of (2) (with either kinetic forms) was undertaken by the author in [Garvie and Trenchea \(2005b\)](#) using the standard Galerkin finite-element method ([Brenner and Scott, 1994](#); [Ciarlet, 1979](#)) with piecewise linear basis functions. With appropriate conditions on the discretization parameters and initial data, the finite-element methods possess several theoretical and implementational advantages, namely

- (i) they are stable;
- (ii) they are convergent;
- (iii) the resulting linear systems have a simple structure;

- (iv) the schemes have equivalent finite-difference representations on rectangular domains.

As a consequence of (iii), standard direct and iterative solvers can be employed to solve the resulting linear systems, and due to the last property, the stability, convergence, and implementational features of the finite-element methods can be carried over to the less theoretical finite-difference setting on rectangular domains, which is the focus of this paper. In this setting, by ‘equivalent’ we mean that the linear systems obtained from the finite-element and finite-difference methods are identical, thus the respective numerical results will be identical. With appropriate modification of the boundary conditions more general domains can be handled as well.

The finite-difference method (Hildebrand, 1968; Morton and Mayers, 1996; Richtmyer and Morton, 1967) is widely used by the scientific community for the numerical solution of reaction–diffusion equations; however, there are comparatively few studies that give stability and convergence results (see for example Ascher et al., 1995; Beckett and Mackenzie, 2001; Hoff, 1978; Jerome, 1984; Li et al., 1994; Mickens, 2003; Pao, 1998, 1999, 2002; Pujol and Grimalt, 2002). The need for a rigorous numerical analysis is due to the well-known (but often neglected) fact that the dynamics of the approximations of nonlinear differential equations (DEs) can differ significantly from that of the original DEs themselves. The literature abounds with examples of spurious behavior of numerical solutions, e.g., spurious fixed points, numerical ‘chaos,’ and spurious periodicity, which does not reflect the behavior of the underlying continuous model. The use of approximation methods with known stability and convergence properties is particularly important for the predator–prey models studied in this paper as the solutions are inherently chaotic (Medvinsky et al., 2002; Sherratt et al., 1997, 1995). Thus, with careful application of the finite-difference methods presented in this paper, the scientist can distinguish between the onset of numerical instability and chaos that is a true feature of the continuous model. For a unified treatment of how and when the finite-difference method for reaction–diffusion equations breaks down see Stuart (1989), Elliott and Stuart (1993), and Ruuth (1995); for a more general treatment of the area of spurious solutions of DEs see Stuart and Humphries (1998, Chapter 5), Yee and Sweby (1994, 1995), and the references therein.

2.2. Preliminaries

In order to construct the finite-difference methods in one space dimension, take a uniform subdivision of the interval $\Omega = [A, B]$, $0 \leq A < B$, with grid points $x_i = ih + A$, $i = 0, \dots, J$, and space step $h = (B - A)/J$. For the two-dimensional approximations, we use a uniform subdivision of the square $\Omega = [A, B] \times [A, B]$ with grid points $(x_i, y_j) = (ih + A, jh + A)$, $i, j = 0, \dots, J$. It will be convenient to define

$$\tilde{J} = \begin{cases} J & \text{if } d = 1, \\ (J + 1)^2 - 1 & \text{if } d = 2, \end{cases}$$

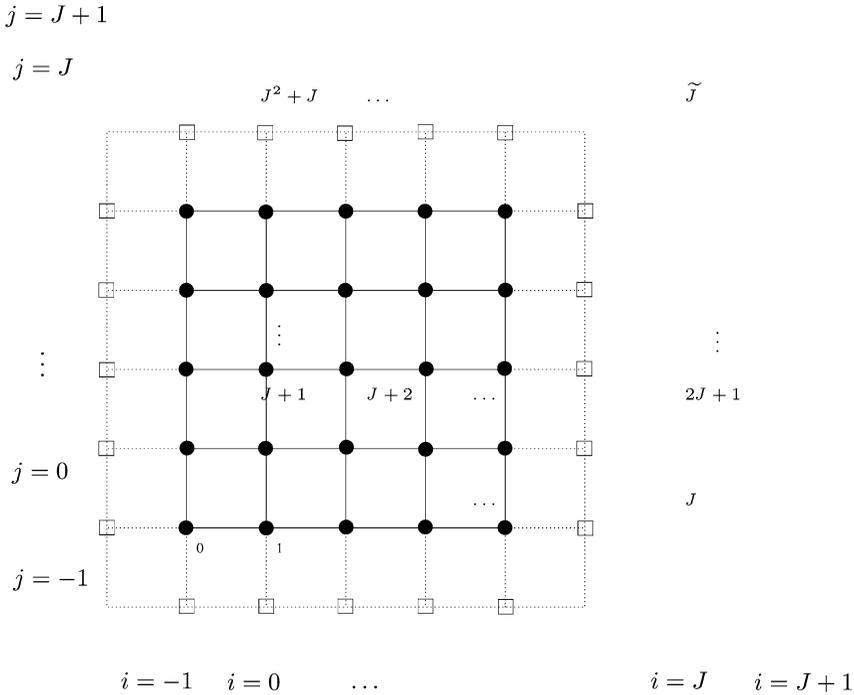


Fig. 2 Two-dimensional grid with nodes (•) and fictitious nodes (◻).

so that in both dimensions, we have $2(\tilde{J} + 1)$ unknowns to solve for. The computational grid in the two dimensional case is illustrated in Fig. 2. We also take a uniform subdivision of the time interval $[0, T]$ with time levels $t_n = n\Delta t$, $n = 1, \dots, N$, so the time step is $\Delta t = T/N$. The approximation to the solution $\vec{u} = (u, v)^T$ of (2) in one dimension at the point (x_i, t_n) is denoted by $\vec{U}_i^n = (U_i^n, V_i^n)^T$, while $\vec{U}_{i,j}^n = (U_{i,j}^n, V_{i,j}^n)^T$ denotes the two-dimensional approximation at the point (x_i, y_j, t_n) . Nodes on the two-dimensional grid are numbered in the natural way, i.e., numbered consecutively from left to right starting with the bottom row, from $k = 0, 1, \dots, \tilde{J}$. The relationship between the natural numbering of the nodes and the (i, j) indexing is given by

$$U_k^n = U_{i,j}^n \quad \text{where } k = i + j(J + 1) \quad \text{for } i, j = 0, \dots, J. \tag{4}$$

The (i, j) indexing of the approximations is used to express the finite-difference schemes, while indexing based on the natural numbering of the nodes is used in the resulting linear systems. These two indexing systems coincide in the one-dimensional case.

Two semi-implicit (in time) finite-difference methods are presented in Schemes 1 and 2. The methods are called semi-implicit as the right-hand side of the schemes involve approximations at the current time level t_n and at the previous time level t_{n-1} . Both methods lead to a sparse, banded, linear system of algebraic equations.

In order to approximate the predator–prey system with *stable* finite-difference methods (see Garvie and Trenchea, 2005b), we replace the functional responses (3a)–(3b) by the modified functional responses

$$\widehat{h}(\chi) = \widehat{h}_1(\chi) = \frac{\chi}{1 + |\chi|}, \tag{5}$$

$$\widehat{h}(\chi) = \widehat{h}_2(\chi) = 1 - e^{-|\chi|}, \tag{6}$$

and modify the logistic growth term via

$$\chi(1 - \chi) \longrightarrow \chi(1 - |\chi|).$$

The following notation is used to simplify the forward differences in time, the central difference approximation of the Laplacian in one dimension, and the five-point central difference approximation of the Laplacian in two dimensions:

$$\partial_n \psi^n = (\psi^n - \psi^{n-1})/\Delta t, \tag{7}$$

$$\Delta_h \eta_i = (\eta_{i+1} - 2\eta_i + \eta_{i-1})/h^2, \tag{8}$$

$$\Delta_h \chi_{i,j} = (\chi_{i,j+1} + \chi_{i,j-1} + \chi_{i+1,j} + \chi_{i-1,j} - 4\chi_{i,j})/h^2. \tag{9}$$

Note that we use the same symbol Δ_h for the discrete Laplacian operator in both the one- and two-dimensional cases, as the context will make it clear which one we refer to.

2.3. General form of difference schemes

In the following, we denote \widehat{f} and \widehat{g} to be the (“modified”) discrete kinetic functions corresponding to f and g . The one-dimensional linear schemes have the following general form.

For $n = 1, \dots, N$ and $i = 0, \dots, J$ find $\{U_i^n, V_i^n\}$ such that

$$\begin{cases} \partial_n U_i^n = \Delta_h U_i^n + \widehat{f}(\bar{U}_i^n, \bar{U}_i^{n-1}), \\ \partial_n V_i^n = \delta \Delta_h V_i^n + \widehat{g}(\bar{U}_i^n, \bar{U}_i^{n-1}), \end{cases} \tag{10}$$

with initial approximations given by

$$U_i^0 := u_0(x_i), \quad V_i^0 := v_0(x_i). \tag{11}$$

The two-dimensional linear schemes have the following general form.

For $n = 1, \dots, N$ and $i, j = 0, \dots, J$ find $\{U_{i,j}^n, V_{i,j}^n\}$ such that

$$\begin{cases} \partial_n U_{i,j}^n = \Delta_h U_{i,j}^n + \widehat{f}(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}), \\ \partial_n V_{i,j}^n = \delta \Delta_h V_{i,j}^n + \widehat{g}(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}), \end{cases} \tag{12}$$

with initial approximations given by

$$U_{i,j}^0 := u_0(x_i, y_j), \quad V_{i,j}^0 := v_0(x_i, y_j). \quad (13)$$

In both cases, we also need a number of auxiliary conditions that approximate the zero-flux boundary conditions of the continuous equations. For ease of exposition, these are described in Appendix A.

2.4. Scheme 1

The first scheme involves an approximation of the reaction kinetics with terms at the current time level t_n and the previous time level t_{n-1} . The kinetics in one dimension corresponding to (10) are

$$\hat{f}(\bar{U}_i^n, \bar{U}_i^{n-1}) = f_1(\bar{U}_i^n, \bar{U}_i^{n-1}) = U_i^n - U_i^n |U_i^{n-1}| - V_i^n \hat{h}(aU_i^{n-1}), \quad (14a)$$

$$\hat{g}(\bar{U}_i^n, \bar{U}_i^{n-1}) = g_1(\bar{U}_i^n, \bar{U}_i^{n-1}) = bV_i^n \hat{h}(aU_i^{n-1}) - cV_i^n, \quad (14b)$$

while in two dimensions, the kinetics corresponding to (12) are given analogously by

$$\hat{f}(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}) = f_1(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}) = U_{i,j}^n - U_{i,j}^n |U_{i,j}^{n-1}| - V_{i,j}^n \hat{h}(aU_{i,j}^{n-1}), \quad (15a)$$

$$\hat{g}(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}) = g_1(\bar{U}_{i,j}^n, \bar{U}_{i,j}^{n-1}) = bV_{i,j}^n \hat{h}(aU_{i,j}^{n-1}) - cV_{i,j}^n. \quad (15b)$$

Scheme 1 can be expressed as $2(\tilde{J} + 1)$ linear equations in the following block matrix form with the ‘natural numbering’ of the nodes:

$$\begin{pmatrix} A_{n-1} & B_{n-1} \\ 0 & C_{n-1} \end{pmatrix} \begin{pmatrix} \bar{U}^n \\ \bar{V}^n \end{pmatrix} = \begin{pmatrix} \bar{U}^{n-1} \\ \bar{V}^{n-1} \end{pmatrix}, \quad 1 \leq n \leq N, \quad (16)$$

where

$$\bar{U}^n = (U_0^n, \dots, U_J^n)^T, \quad \bar{V}^n = (V_0^n, \dots, V_J^n)^T.$$

The constant coefficient matrix L , and coefficient matrices A_{n-1} , B_{n-1} , and C_{n-1} , depending on the solution at time level t_{n-1} , are given in Appendix B.

2.5. Scheme 2

The second scheme is simpler than the first, as it involves an approximation of the reaction kinetics with terms entirely at the previous time level t_{n-1} . The kinetics in one dimension corresponding to (10) are

$$\hat{f}(\bar{U}_i^n, \bar{U}_i^{n-1}) = f_2(\bar{U}_i^{n-1}) = U_i^{n-1} - U_i^{n-1} |U_i^{n-1}| - V_i^{n-1} \hat{h}(aU_i^{n-1}), \quad (17a)$$

$$\widehat{g}(\vec{U}_i^n, \vec{U}_i^{n-1}) = g_2(\vec{U}_i^{n-1}) = bV_i^{n-1}\widehat{h}(aU_i^{n-1}) - cV_i^{n-1}, \tag{17b}$$

while in two dimensions, the kinetics corresponding to 12 are given analogously by

$$\widehat{f}(\vec{U}_{i,j}^n, \vec{U}_{i,j}^{n-1}) = f_2(\vec{U}_{i,j}^{n-1}) = U_{i,j}^{n-1} - U_{i,j}^{n-1}|U_{i,j}^{n-1}| - V_{i,j}^{n-1}\widehat{h}(aU_{i,j}^{n-1}), \tag{18a}$$

$$\widehat{g}(\vec{U}_{i,j}^n, \vec{U}_{i,j}^{n-1}) = g_2(\vec{U}_{i,j}^{n-1}) = bV_{i,j}^{n-1}\widehat{h}(aU_{i,j}^{n-1}) - cV_{i,j}^{n-1}. \tag{18b}$$

Scheme 2 can be expressed as $2(\tilde{J} + 1)$ linear equations in the following block matrix form with the ‘natural numbering’ of the nodes:

$$\begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} \vec{U}^n \\ \vec{V}^n \end{pmatrix} = \begin{pmatrix} \vec{U}^{n-1} + \Delta t \vec{F} \\ \vec{V}^{n-1} + \Delta t \vec{G} \end{pmatrix}, \quad 1 \leq n \leq N, \tag{19}$$

where

$$\{\vec{F}\}_k = f_2(\vec{U}_k^{n-1}), \quad \{\vec{G}\}_k = g_2(\vec{U}_k^{n-1}), \quad \text{for } k = 0, \dots, \tilde{J},$$

(see (4)). The constant coefficient matrices B_1 and B_2 are given in Appendix B.

2.6. Some implementational issues

We solve the linear system (16) at each time step in two stages. First, we solve $C_{n-1}\vec{V}^n = \vec{V}^{n-1}$ for \vec{V}^n , and then solve $A_{n-1}\vec{U}^n = \vec{U}^{n-1} - B_{n-1}\vec{V}^n$ for \vec{U}^n . In the case of linear system (19), due to the block diagonal form and the constant coefficients of B_1 and B_2 , we solve for \vec{U}^n and \vec{V}^n independently at each time level. Thus, in both cases, due to the simple structure of the linear systems, we effectively solve tri-diagonal systems in one dimension, and block tri-diagonal systems in two dimension, i.e., the situation is similar to approximating the solutions of a scalar reaction–diffusion system.

From Garvie and Trenchea (2005b) it follows that

- (H1) the initial data u_0, v_0 is bounded and nonnegative, with continuous first and second derivatives,
- (H2) $\Delta t < \min\{\frac{1}{7}, \frac{1}{1+4(b-c)}\}$, $(b > c)$,
- (H3) $h \leq 1$,

then Schemes 1 and 2 are stable finite-difference approximations and the coefficient matrices of the linear systems (19) and (16) are strictly (row) diagonally dominant. From the elementary numerical analysis, we recall that Gaussian elimination can be successfully applied to a strictly diagonally dominant matrix A , without the need for partial pivoting. With regard to Scheme 2, as B_1 and B_2 are constant coefficient matrices, in order to solve for \vec{U}^n and \vec{V}^n , we need only apply LU factorization *once* (in each case), followed by forward and backward substitution at each time step. Iterative linear solvers can also be employed. For example, given any

initial approximation, the Jacobi and Gauss–Seidel iterative methods give sequences that converge to the unique solution of $A\vec{x} = \vec{b}$ (Isaacson and Keller, 1966).

For higher dimensional problems, more sophisticated iterative solvers may be needed due to the extra storage and computational requirements, e.g., Krylov subspace methods. An example in this class is the GMRES algorithm with ‘restarts’ (Saad and Schultz, 1986). If the matrix A is strictly diagonally dominant, then this iterative solver converges for any restart value (Saad, 2003).

3. Numerical results

After numerically calculating the rates of convergence for the finite-difference methods, the results of experiments in one and two space dimensions are presented. The linear systems resulting from the solution of the one-dimensional problems were solved using LU factorization, while the linear systems resulting from the solution of the two-dimensional problems were solved using the GMRES algorithm (without preconditioning). Our criterion for deciding when numerical solutions converged was to reduce the time step, with the space kept sufficiently small to clearly see the qualitative features of solutions, until there was virtually no difference between approximations from Schemes 1 and 2.

All codes were run in MATLAB 6.5.1 (R13SP1). Note that there is currently a bug in MATLAB 7.0.1 (R14SP1) when using the GMRES function. MathWorks Inc. (support@mathworks.com) can send a corrected version of the GMRES function needed to run our codes with the latest version of MATLAB. Instructions, plus the problem description, are given at <http://www.mathworks.com/support/solutions/data/1-Z81WY.html?solution=1-Z81WY>.

In the numerical experiments, we chose parameter sets that guarantee a stable limit cycle in the reaction kinetics surrounding an unstable steady state (u^* , v^*). Thus, the densities of predators and prey are oscillatory, which is the situation of primary interest from an ecological point of view.

3.1. Rate of convergence

We numerically verify the (pointwise) convergence rates of Scheme 1 for the predator–prey system with Kinetics (i) in one dimension. As no exact solution of the predator–prey system (2) is known, the approximation on a fine mesh and a small time step was compared with the corresponding approximations on a sequence of coarse meshes and larger time steps. The fine solution from Scheme 1 was used in place of an ‘analytical’ solution. Similar results were obtained for Kinetics (ii) and with Scheme 2 (results omitted).

Let \vec{U}^n be the solution vector of (16) (for the prey) calculated on the coarse grid with space step h and time steps Δt at time T . Let \vec{u}^n be the solution vector of (16) (for the prey) calculated on the fine mesh with space step h_{fine} and time steps Δt_{fine} at time T . The l_∞ norm of the ‘error’ between the fine solution and the coarse solution is given by

Table 1 Rates of convergence of Scheme 1 with Kinetics (i).

h	$E^n(h, 1/57344)$	R^h (3 s.f.)
1	3.914e-03	4.31
1/2	9.081e-04	4.06
1/4	2.238e-04	4.01
1/8	5.577e-05	4.00
1/16	1.393e-05	–

Note. $\Delta t = (1/57344)$, $h = (1/2^j)$, $j = 0, 1, 2, 3, 4$.

$$E^n(h, \Delta t) := \|\vec{u}^n - \vec{U}^n\|_\infty = \max_{0 \leq j \leq J} |u_j^n - U_j^n|.$$

We fixed $h_{\text{fine}} = 1/1024$, $\Delta t_{\text{fine}} = 1/57344$, $A = 0$, $B = 50$, $T = 10/7$, $\alpha = 0.3$, $\beta = 2.0$, $\gamma = 0.8$, and chose the initial data

$$u_0(x) = 0.2 + 0.2 \cos(\pi x/16),$$

$$v_0(x) = 0.4 + 0.4 \sin(\pi x/16).$$

The following ratios were computed

$$R^h := \frac{E^n(h, \Delta t_{\text{fine}})}{E^n(h/2, \Delta t_{\text{fine}})}, \quad R^{\Delta t} := \frac{E^n(h_{\text{fine}}, \Delta t)}{E^n(h_{\text{fine}}, \Delta t/2)},$$

to obtain the results in Tables 1 and 2. The tabulated results indicate that the rates of convergence are first order in the time step, and second order in the space step, i.e.,

$$u_j^n - U_j^n = \mathcal{O}(\Delta t + h^2).$$

3.2. Experiments in one dimension

In Fig. 3, the numerical solutions of the predator–prey system are plotted for Scheme 1 with Kinetics (i). The parameter values and initial data are given in the caption. The initial data are small spatial perturbations of the stationary so-

Table 2 Rates of convergence of Scheme 1 with Kinetics (i).

Δt	$E^n(1/1024, \Delta t)$	$R^{\Delta t}$ (3 s.f.)
1/7	2.807e-02	2.09
1/14	1.340e-02	2.05
1/28	6.548e-03	2.02
1/56	3.237e-03	2.01
1/112	1.608e-03	–

Note. $h = (1/1024)$, $\Delta t = (1/\{7(2^j)\})$, $j = 0, 1, 2, 3, 4$.

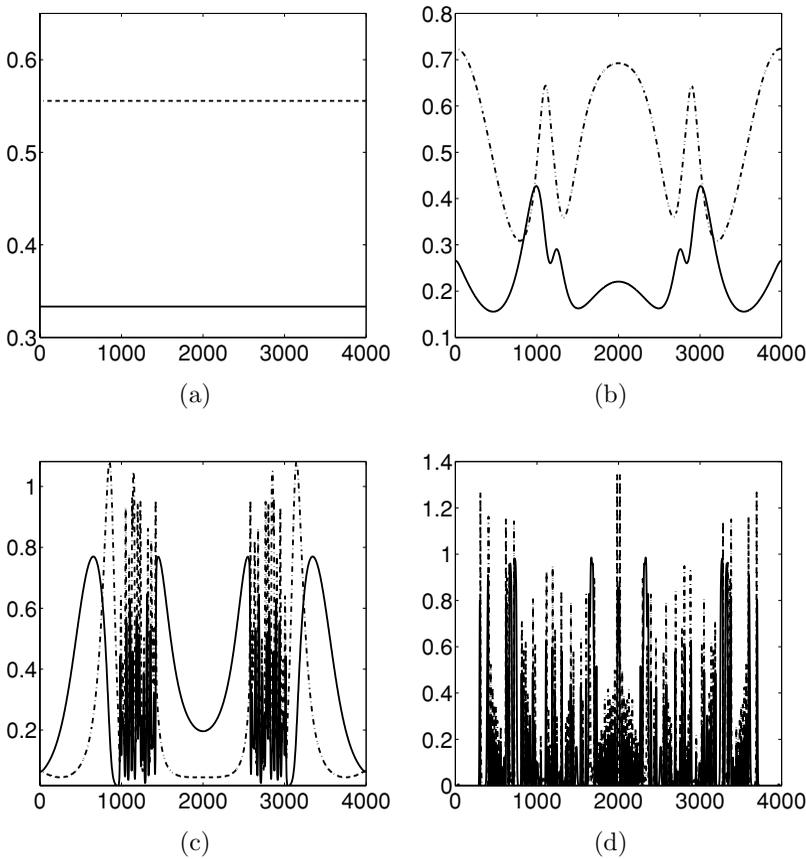


Fig. 3 One-dimensional numerical solutions of the predator–prey system with Kinetics (i), using Scheme 1 at $T = 600$: *solid lines* for prey U_i^n and *dashed lines* for predators V_i^n . Parameter values and initial data: $\beta = 2, \gamma = 4/5, \delta = 1, h = 1, \Delta t = 10^{-4}, U_i^0 = u^* + 10^{-8}(x_i - 1200)(x_i - 2800), V_i^0 = v^*$. (a) $\alpha = 1/2$ ($u^* = 1/3, v^* = 5/9$), (b) $\alpha = 33/80$ ($u^* = 11/40, v^* = 319/640$), (c) $\alpha = 3/10$ ($u^* = 1/5, v^* = 2/5$), (d) $\alpha = 1/20$ ($u^* = 1/30, v^* = 29/360$).

lutions u^* and v^* of the spatially homogeneous system. Varying α from 0.05 to 0.5, led to the four basic one-dimensional dynamics, namely stationary, smooth oscillatory, intermittent ‘chaos,’ and ‘chaos’ covering (almost all) of the domain. In Fig. 3(a), the approximations have evolved into the spatially homogeneous stationary states u^* and v^* . In Fig. 3(c), we used the same initial data and parameter values as in Medvinsky et al. (2002), resulting in the same intermittent spatial structures at roughly the same positions.

3.3. Experiments in two dimension

In Figs. 4–7, the numerical solutions $U_{i,j}^n$ of the predator–prey system are plotted using Scheme 1 (figures in first columns), and Scheme 2 (figures in second

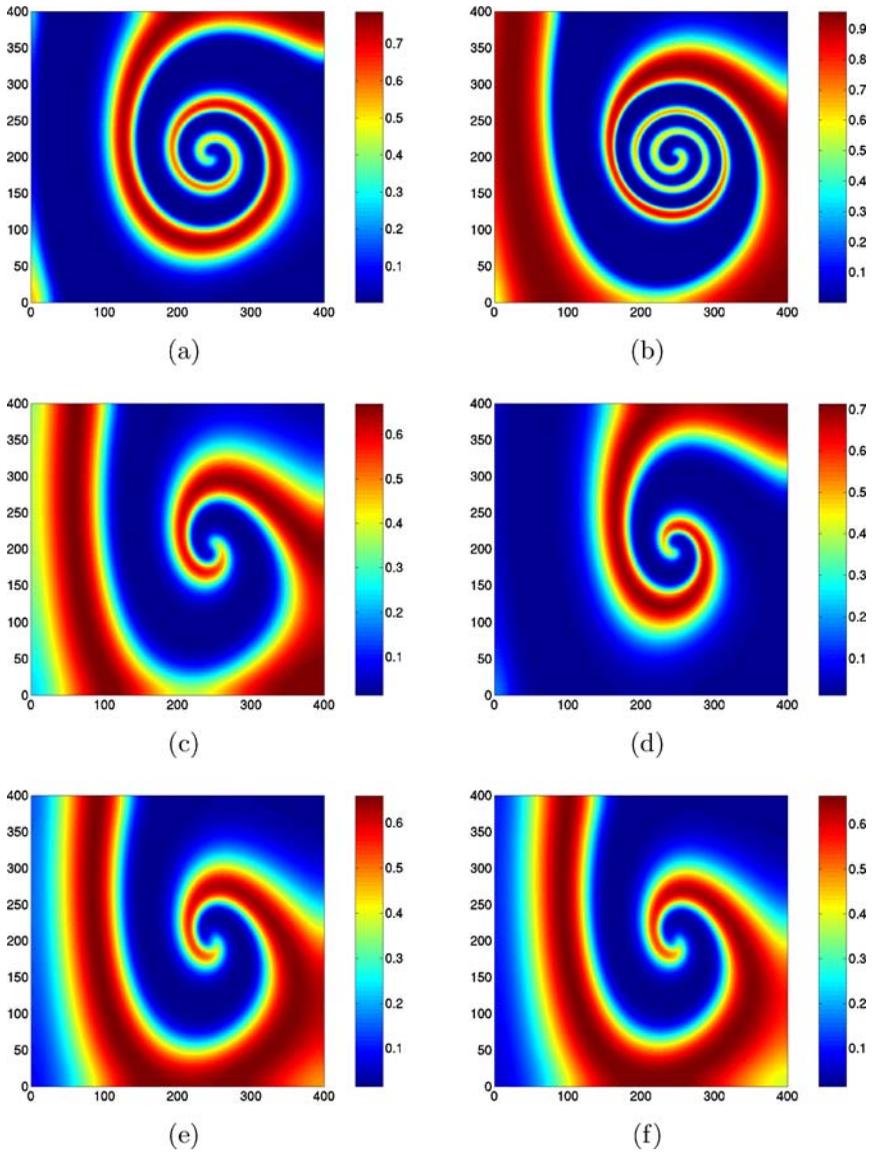


Fig. 4 Two-dimensional approximate prey densities $U_{i,j}^n$ for Kinetics (i), using Scheme 1 (first column figures) and Scheme 2 (second column figures) at $T = 150$. Parameter values and initial data: $\alpha = 0.4$, $\beta = 2.0$, $\gamma = 0.6$, $\delta = 1$, $U_{i,j}^0 = u^* - 2 \times 10^{-7}(x_i - 0.1y_j - 225)(x_i - 0.1y_j - 675)$, $V_{i,j}^0 = v^* - 3 \times 10^{-5}(x_i - 450) - 1.2 \times 10^{-4}(y_j - 150)$. Plots show successive refinements of Δt with h fixed at 1: (a)–(b) $\Delta t = 1/3$, (c)–(d) $\Delta t = 1/24$, (e)–(f) $\Delta t = 1/384$ ($u^* = 6/35$, $v^* = 116/245$).

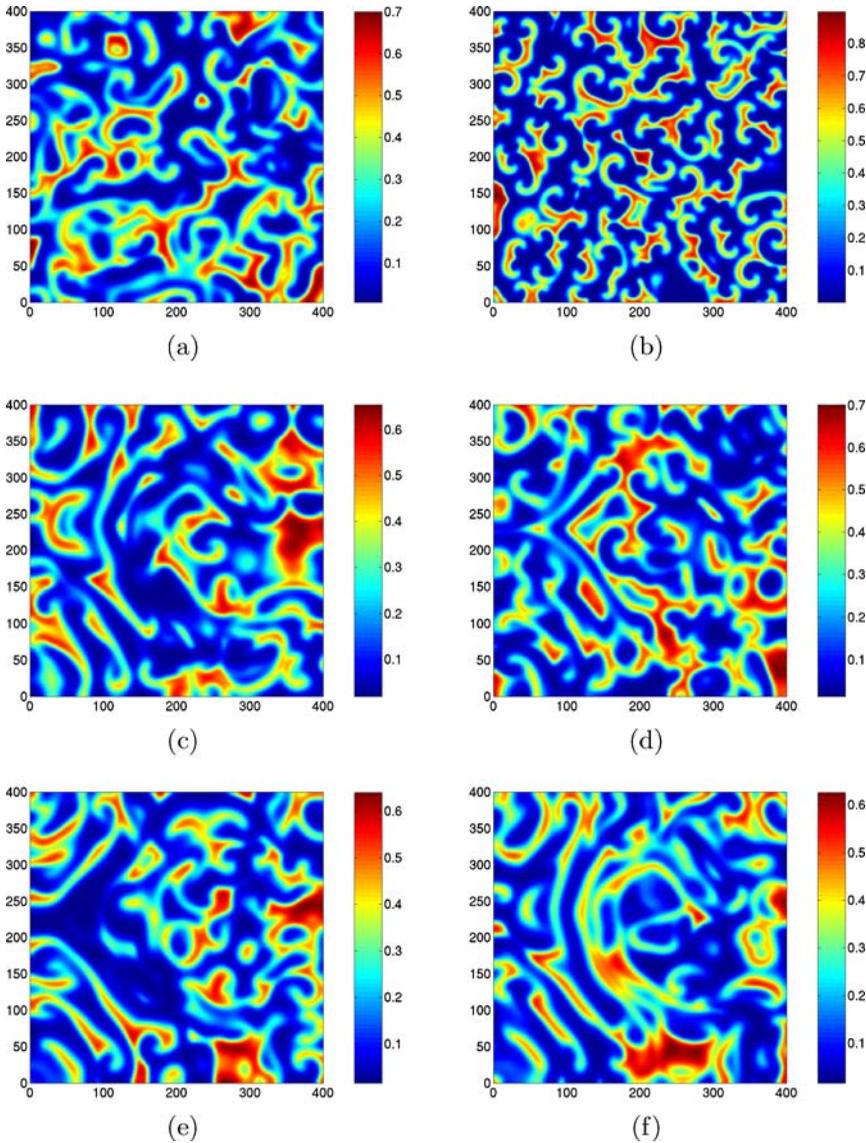


Fig. 5 Two-dimensional approximate prey densities $U_{i,j}^n$ for Kinetics (i), using Scheme 1 (first column figures) and Scheme 2 (second column figures) at $T = 1000$. Parameter values and initial data as in Fig. 4. This is the same experiment as in Fig. 4 but at a later time T .

columns). Kinetics (i) was used to generate Figs. 4–6, while Kinetics (ii) was employed in Fig. 7. The initial data and parameter values of Figs. 4–6 were chosen to correspond to Figs. 10(b), (f), and 11(b), respectively, in Medvinsky et al. (2002). As Δt is reduced to $1/384$, we observe the convergence of the numerical solutions from Schemes 1 and 2 in each case. In all cases, the space step h was

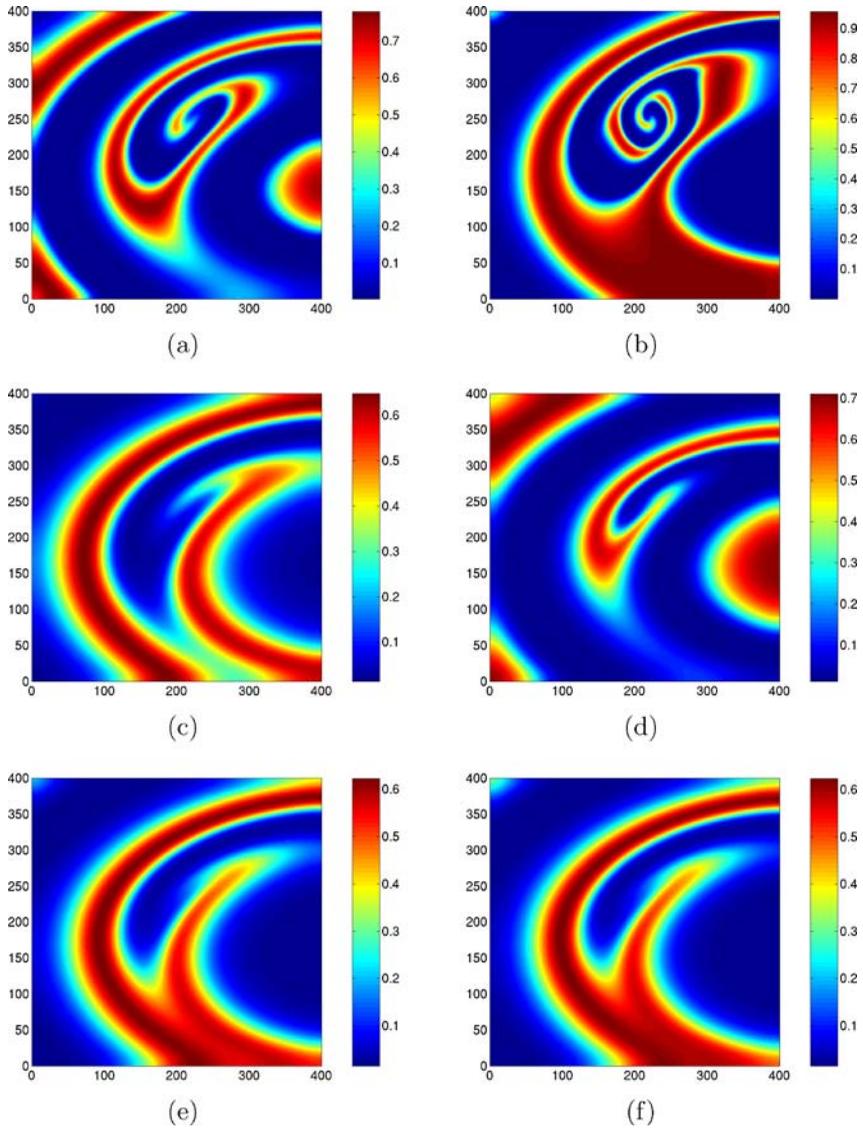


Fig. 6 Two-dimensional approximate prey densities $U_{i,j}^n$ for Kinetics (i), using Scheme 1 (first column figures) and Scheme 2 (second column figures) at $T = 120$. Parameter values and initial data: $\alpha = 0.4$, $\beta = 2.0$, $\gamma = 0.6$, $\delta = 1$, $U_{i,j}^0 = u^* - 2 \times 10^{-7}(x_i - 180)(x_i - 720) - 6 \times 10^{-7}(y_j - 90)(y_j - 210)$, $V_{i,j}^0 = v^* - 3 \times 10^{-5}(x_i - 450) - 6 \times 10^{-5}(y_j - 135)$. Plots show successive refinements of Δt with h fixed at 1: (a)–(b) $\Delta t = 1/3$, (c)–(d) $\Delta t = 1/24$, (e)–(f) $\Delta t = 1/384$ ($u^* = 6/35$, $v^* = 116/245$).

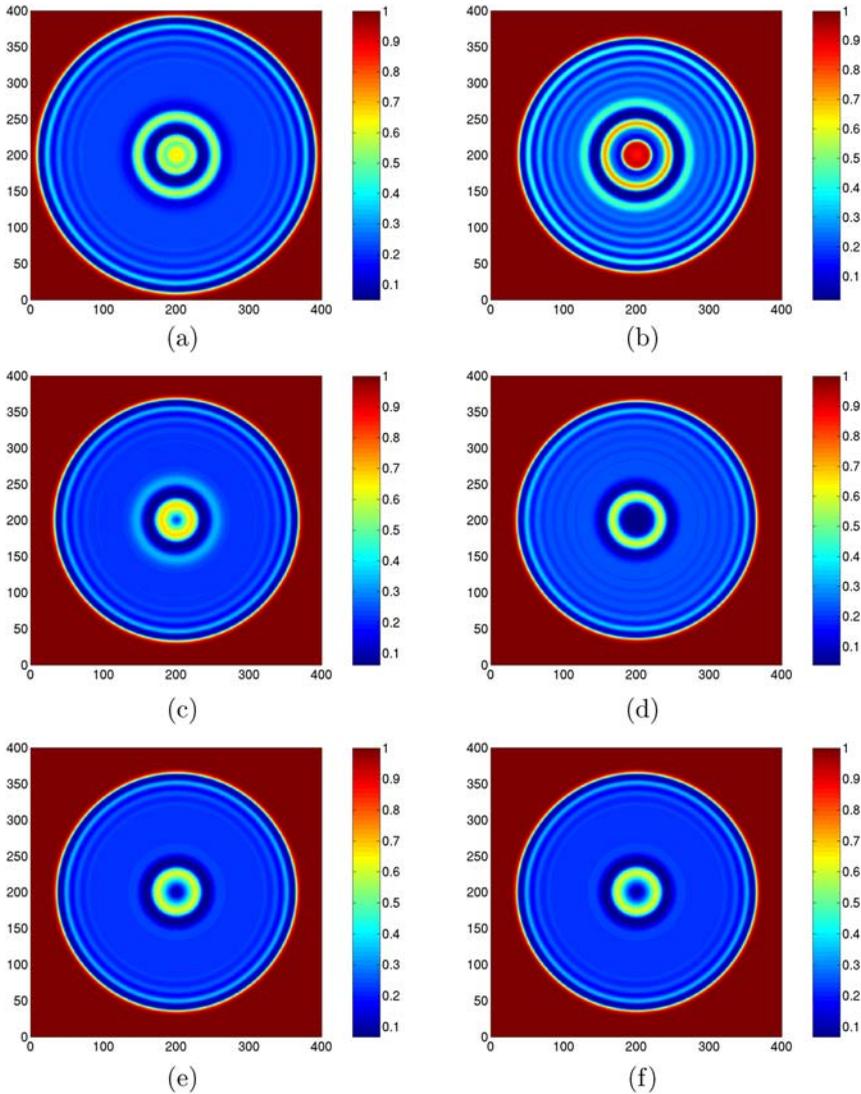


Fig. 7 Two-dimensional approximate prey densities $U_{i,j}^n$ for Kinetics (ii), using Scheme 1 (first column figures) and Scheme 2 (second column figures) at $T = 110$. Parameter values and initial data: $\alpha = 1.5$, $\beta = 1.0$, $\gamma = 5.0$, $\delta = 1$, $U_{i,j}^0 = 1.0$, $V_{i,j}^0 = 0.2$ if $(x_i - 200)^2 + (y_j - 200)^2 < 400$ and zero otherwise. Plots show successive refinements of Δt with h fixed at 1: (a)–(b) $\Delta t = 1/3$, (c)–(d) $\Delta t = 1/24$, (e)–(f) $\Delta t = 1/384$.

kept equal to unity, i.e., $\Delta x = \Delta y = 1$. We repeated some of the experiments with $h = 1/2$ to check that further reductions in the space step had no significant affect on the numerical results. Figure 5 demonstrates that we were unable to achieve a perfect match between the numerical solutions for Schemes 1 and 2. This may

be due to the fact that first-order time-stepping schemes may fail to accurately reproduce highly oscillatory solutions for reaction–diffusion systems (Ruuth, 1995). Furthermore, the final time T used to obtain the results in Fig. 5 is much greater than the final time used to obtain Figs. 4, 6, and 7, resulting in a greater global error.

In Figs. 4–6, the initial data are small spatial perturbations of the stationary solutions u^* and v^* of the spatially homogeneous system. In Fig. 4, the evolution of the system led to the formation of spiral patterns, followed by irregular patches covering the whole domain (see Fig. 5), confirming the qualitative behavior reported in (Medvinsky et al., 2002). The size of these patches has been related to the characteristic length of observed plankton patterns in the ocean (see Medvinsky et al., 2002, and the references therein). An animation of this spiral wave ‘break-up’ from $t = 0$ to 1000 can be seen at <http://www.uoguelph.ca/~mgarvie/> (click under ‘research’ and see the section entitled ‘Spatially-extended predator–prey models’).

In Fig. 6, we observe that as Δt is reduced, the initial spiral patterns disappear, indicating that the spiral pattern of Fig. 11(b) in Medvinsky et al. (2002) is probably a numerical artifact. The different numerical results may be due to the different domain shape used in Medvinsky et al. (2002), namely, a 900×300 rectangle. However, we think this is unlikely. Preliminary results obtained on the 900×300 rectangle (omitted for the sake of brevity) did not differ significantly from the results obtained on the square domain. We comment that we *would* expect significant differences in the numerical results obtained on the two domains with different boundary conditions, e.g., in the Homogeneous Dirichlet boundary condition case.

In Fig. 7, the initial localized introduction of predators into a uniform distribution of prey led to the spread of predators over the domain, with perfectly circular bands of regular spatiotemporal oscillations behind the wave front (‘target pattern’). This behavior mimics the invasion of prey by predators in ecological systems. For an in-depth discussion of the possible applicability of these results to real ecological communities see Sherratt et al. (1997).

Note that in Medvinsky et al. (2002), the numerical method and the space and time steps used to obtain the two-dimensional results are not stated, thus only approximate comparisons could be made with our results.

4. Description of the MATLAB code

The MATLAB code in Appendix B is mostly self explanatory, with the names of variables and parameters corresponding to the symbols used in the finite-difference methods described in Section 2. The codes for Schemes 1 and 2, with either Kinetics (i) or (ii), can be downloaded from the web site: <http://www.uoguelph.ca/~mgarvie/> (accessed December 2005). We give a description of the different parts of the one-dimensional and two-dimensional MATLAB codes for Scheme 2 applied to Kinetics (i), called FD1D and FD2D, respectively, and provide some user details. The codes for Scheme 2 applied to Kinetics (ii) and Scheme 1 are similar.

The code employs the sparse matrix facilities in MATLAB when assembling and solving the linear systems, which provides advantages in both matrix storage and computation time. The code is *vectorized* to minimize the number of ‘for-loops’ and conditional ‘if-then-else’ statements, which again helps speed up the computations. To solve the linear systems we used MATLAB’s built in functions LU and GMRES in FD1D and FD2D, respectively. The GMRES algorithm in MATLAB requires a number of input arguments. For the experiments, we found it acceptable to use the default settings, e.g., no ‘restarts’ of the iterative method or use of preconditioners, and a tolerance for the relative error of 1×10^{-6} . In practice, the user will need to experiment with the restart value, tolerance, and other input parameters in order to achieve satisfactory rates of convergence of the GMRES function. For definitions of these input arguments (and others), see the description in MATLAB’s Help pages. We remark that a pure C or FORTRAN code is likely to be faster than the MATLAB codes but with the disadvantage of much greater complexity and length.

The user is prompted for all necessary parameters, time and space steps, and initial data. Due to a limitation in MATLAB vector indices cannot be equal to zero, thus the nodal indices $0, \dots, J$ are shifted up one unit to give $1, \dots, J + 1$ so $x_i = (i - 1)h + A$. Note that we use ‘*a*’ and ‘*b*’ instead of ‘*A*’ and ‘*B*’ in the code, in keeping with the MATLAB convention of reserving capital letters for matrices.

4.1. One-dimensional code

Program FD1D, listed in Appendix D, is structured as follows:

- Lines 4–12: User prompted for model parameters.
- Lines 14–15: User prompted for initial data as a string (allowable formats discussed below).
- Lines 17–20: Calculate some constants.
- Lines 22–25: Initialize matrices.
- Lines 27–29: Assign initial data numerically.
- Lines 31–38: Assemble matrices L , B_1 , and B_2 .
- Lines 40–41: LU factorization of B_1 and B_2 .
- Lines 43–59: Solve the linear system repeatedly up-to time level $t_N = T$ using forward and backward substitution.
- Line 61: Plot numerical solutions for u and v at time T .

The initial data functions are entered by the user as a string, which can take several different formats. Functions are evaluated on an element-by-element basis, where $x = (x_1, \dots, x_{J+1})$ is a vector of grid points, and so a ‘.’ must precede each arithmetic operation between matrices. The exception to this rule is when applying MATLAB’s intrinsic functions where there is no ambiguity. Some arbitrary examples with an acceptable format include the following:

```
>> Enter initial prey function u0(x) 0.2*exp(-(x-100).^2)
>> Enter initial predator function v0(x) 0.4*x./(1+x)
```

or

```
>> Enter initial prey function u0(x) 0.3+(x-1200) .
      *(x-2800)
>> Enter initial predator function v0(x) 0.4
```

This last example shows that for a constant solution vector, we need only enter a single number. It is also possible to enter functions that are piecewise defined by utilizing MATLAB's logical operators & ('AND'), | ('OR'), ~ ('NOT'), applied to matrices. For example, on a domain $\Omega = [0, 200]$ to choose an initial prey density that is equal to 0.4 for $90 \leq x_i \leq 110$, and equal to 0.1 otherwise, the user inputs:

```
>> Enter initial prey function u0(x)
0.4*(x>=90)&(x<=110)+0.1*(x<90)|(x>110)
```

4.2. Two-dimensional code

Program FD2D, listed in Appendix D, is structured as follows:

- Lines 4–12: User prompted for model parameters.
- Lines 14–15: User prompted for initial data as a string (allowable formats discussed below).
- Lines 17–21: Calculate some constants.
- Lines 23–25: Initialize matrices.
- Lines 27–30: Assign initial data numerically.
- Line 32: Re-order initial grid values into solution vector consistent with natural ordering of the grid.
- Lines 34–58: Assemble matrices L , B_1 , and B_2 .
- Lines 60–73: Solve the linear system repeatedly up-to time level $t_N = T$ using GMRES.
- Lines 75, 77: Re-order solution vector back to a solution grid in an appropriate orientation for plotting.
- Lines 79–80: Plot numerical solutions for u and v at time T , where color represents density of predators and prey (a vertical 'colorbar' provides a scale).

The initial data is also entered by the user as a string, but now the functions depend on a grid of x and y values X and Y (the capitals indicate that these are matrices). Functions are entered using the same element-by-element rules for the arithmetic operation as in the one-dimensional code. An example with an acceptable format is the following:

```
>> Enter initial prey function U0(X,Y) 0.2*exp(-(X.^2+Y.^2))
>> Enter initial predator function V0(X,Y) 1.0
```

We can also define functions in a piecewise fashion. For example, with $\Omega = [-500, 500]^2$, in order to choose an initial prey density of 0.2 within the circle with radius 10 and center $(-50, 200)$, and a density of 0.01 elsewhere on Ω we input the following:

```
>> Enter initial predator function V0(X,Y)
```

$$0.2 * ((X+50) . ^2 + (Y-200) . ^2) < 100) + 0.01 * ((X+50) . ^2 + (Y-200) . ^2) >= 100)$$

5. Discussion and conclusions

In this final section, we make some concluding comments and discuss our results from both a numerical and an ecological perspectives.

5.1. Numerical comments

We presented two high-quality finite-difference schemes that allowed us to confirm a wide variety of spatiotemporal dynamics reported in the literature for spatially extended predator–prey interactions. Complete implementational details were given so that applied mathematicians and biologists can quickly apply and adapt the numerical methods to investigate the dynamics of predator–prey interactions. Although the finite-difference methods (Schemes 1 and 2) are subject to the same conditions that guarantee stability and convergence, they differ somewhat in their convergence properties. Thus, using both methods together provides a useful additional test of convergence. For example, in Fig. 4, we had to reduce the time step to 1/384 before the snapshots for Schemes 1 and 2 matched well, while Figs. 3 and 5 require a much smaller time step for convergence. Some simple MATLAB code was also presented and described, which can easily be adapted for different initial conditions, parameter values, and kinetics. Due to the simplicity of the schemes, we only need 80 lines of code to solve a two-dimensional problem with 3 million degrees of freedom many thousands of times. The main disadvantage of the code is that the run time can be prohibitive when using a combination of large domain size and final time T coupled with small space and time steps. We make no claims as to the efficiency of the MATLAB code but hope that users will find this a useful starting point for their own work.

5.2. Ecological comments

In one dimension, we showed how modest changes in a single parameter of the system with Kinetics (i), namely α , can lead to dramatic changes in the qualitative dynamics of solutions. The parameter α is the (nondimensional) half-saturation abundance of prey. Thus, the ecological implication is that, how fast the consumption rate of predators saturate with increasing prey density can have a profound effect on the fate of the system. In contrast, the experiments in [Medvinsky et al. \(2002\)](#) focused on how the solution dynamics depend on the current state of the system (initial conditions). The details governing the dynamics of the spatially extended system are complicated and will depend on the system parameters, the initial data, and also the specifics of habitat geometry. Nevertheless, there are situations where the local dynamics of solutions gives us important clues to the behavior in the spatially extended situation. The one- and two-dimensional numerical experiments presented in this paper employ parameter sets that guarantee stable limit cycles in the reaction kinetics. As the diffusion coefficients used are equal,

and the direction field of the reaction kinetics does not point out of the limit cycle (see, for example, Fig. 1), the limit cycle encloses an ‘invariant region’ in phase space (Smoller, 1983). Consequently, as the initial data, used to generate Figs. 4–6, lies entirely in the limit cycle at every point of the domain, the solution is trapped in this region for all time. Numerical experiments for large T confirm this behavior. The ecological implication of these results is that in the absence of external influences, certain initial conditions can lead to spatial and temporal variations in densities of predators and prey that persist indefinitely.

The results of this paper are an important step toward providing the theoretical biology community with simple practical numerical methods, for investigating the key dynamics of realistic predator–prey models.

Appendix A. Boundary conditions for the schemes

In order to implement the one-dimensional finite-difference schemes, with general form (10), we define ‘reflection’ boundary conditions at the endpoints of the domain given by

$$U_{-1}^n := U_1^n, \quad U_{J+1}^n := U_{J-1}^n, \quad V_{-1}^n := V_1^n, \quad V_{J+1}^n := V_{J-1}^n. \quad (\text{A.1})$$

The ‘reflection’ boundary conditions arise from the use of fictitious nodes x_{-1} and x_{J+1} to approximate the zero-flux boundary conditions, via

$$\frac{(U_1^n - U_{-1}^n)}{2h} = 0 = \frac{(U_{J+1}^n - U_{J-1}^n)}{2h},$$

(and similarly for the approximations to v). The two-dimensional case is more complicated. In order to implement the schemes with general form (10), we still require ‘reflection’ boundary conditions along the four edges of the square domain, i.e., for $1 \leq i, j \leq J - 1$ define

$$U_{i,-1}^n := U_{i,1}^n, \quad U_{i,J+1}^n := U_{i,J-1}^n, \quad (\text{A.2})$$

$$U_{J+1,j}^n := U_{J-1,j}^n, \quad U_{-1,j}^n := U_{1,j}^n, \quad (\text{A.3})$$

(and similarly for the approximations to v). However, in addition to these conditions, we require the ‘corner’ boundary conditions

$$U_{0,-1}^n = (U_{0,0}^n + U_{0,1}^n)/2, \quad U_{-1,0}^n = (U_{0,0}^n + U_{1,0}^n)/2, \quad (\text{SW corner})$$

$$U_{J,J+1}^n = (U_{J,J}^n + U_{J,J-1}^n)/2, \quad U_{J+1,J}^n = (U_{J,J}^n + U_{J-1,J}^n)/2, \quad (\text{NE corner})$$

$$U_{J,-1}^n = 2U_{J,1}^n - U_{J,0}^n, \quad U_{J+1,0}^n = 2U_{J-1,0}^n - U_{J,0}^n, \quad (\text{SE corner})$$

$$U_{0,J+1}^n = 2U_{0,J-1}^n - U_{0,J}^n, \quad U_{-1,J}^n = 2U_{1,J}^n - U_{0,J}^n, \quad (\text{NW corner})$$

(and similarly for the approximations to v ; see Fig. 2).

The coefficient matrix of the linear system (16) is defined via

$$A_{n-1} := (1 - \Delta t)I + \Delta t \operatorname{diag}\{|U_0^{n-1}|, \dots, |U_J^{n-1}|\} + \Delta t L,$$

$$B_{n-1} := \Delta t \operatorname{diag}\{\widehat{h}(aU_0^{n-1}), \dots, \widehat{h}(aU_J^{n-1})\},$$

$$C_{n-1} := (1 + \Delta t c)I - \Delta t b \operatorname{diag}\{\widehat{h}(aU_0^{n-1}), \dots, \widehat{h}(aU_J^{n-1})\} + \delta \Delta t L,$$

and the coefficient matrix of the linear system 19 is defined via

$$B_1 := I + \Delta t L,$$

$$B_2 := I + \delta \Delta t L.$$

Acknowledgments

I would like to thank Jonathan A. Sherratt (Heriot-Watt University, UK) and an anonymous referee for his helpful comments.

References

- Alonso, D., Bartumeus, F., Catalan, J., 2002. Mutual interference between predators can give rise to Turing spatial patterns. *Ecology* 83(1), 28–34.
- Ascher, U., Ruuth, S., Wetton, B., 1995. Implicit–explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.* 32(3), 797–823.
- Beckett, G., Mackenzie, J., 2001. On a uniformly accurate finite difference approximation of a singularly perturbed reaction–diffusion problem using grid equidistribution. *J. Comput. Appl. Math.* 131, 381–405.
- Brenner, S., Scott, L., 1994. *The Mathematical Theory of Finite Element Methods*. Vol. 15: Texts in Applied Mathematics. Springer, New York.
- Ciarlet, P., 1979. *The Finite Element Method for Elliptic Problems*. Vol. 4: Studies in Mathematics and its Applications. North-Holland, Amsterdam.
- Elliott, C., Stuart, A., 1993. The global dynamics of discrete semilinear parabolic equations. *SIAM J. Numer. Anal.* 30(6), 1622–1663.
- Freedman, H., 1980. *Deterministic Mathematical Models in Population Ecology*. Vol. 57: Monographs and Textbooks in Pure and Applied Mathematics. Marcel Dekker, New York.
- Garvie, M., Trenchea, C., 2005a. Analysis of two generic spatially extended predator–prey models. *Nonlinear Anal. Real World Appl.*, submitted for publication.
- Garvie, M., Trenchea, C., 2005b. Finite element approximation of spatially extended predator–prey interactions with the Holling type II functional response. *Numer. Math.*, submitted for publication.
- Gentleman, W., Leising, A., Frost, B., Strom, S., Murray, J., 2003. Functional responses for zooplankton feeding on multiple resources: A review of assumptions and biological dynamics. *Deep Sea Res. II* 50, 2847–2875.
- Gurney, W., Veitch, A., Cruickshank, I., McGeachin, G., 1998. Circles and spirals: Population persistence in a spatially explicit predator–prey model. *Ecology* 79(7), 2516–2530.
- Hildebrand, F., 1968. *Finite-Difference Equations and Simulations*. Prentice-Hall, Englewood Cliffs, NJ.
- Hoff, D., 1978. Stability and convergence of finite difference methods for systems of nonlinear reaction–diffusion equations. *SIAM J. Numer. Anal.* 15(6), 1161–1177.
- Holling, C., 1959. Some characteristics of simple types of predation and parasitism. *Can. Entomol.* 91, 385–398.
- Holling, C., 1965. The functional response of predators to prey density and its role in mimicry and population regulation. *Mem. Entomol. Soc. Can.* 45, 1–60.

- Holmes, E., Lewis, M., Banks, J., Veit, R., 1994. Partial differential equations in ecology: Spatial interactions and population dynamics. *Ecology* 75(1), 17–29.
- Isaacson, E., Keller, H., 1966. *Analysis of Numerical Methods*. Wiley, New York.
- Ivlev, V., 1961. *Experimental Ecology of the Feeding Fishes*. Yale University Press, New Haven.
- Jerome, J., 1984. Fully discrete stability and invariant rectangular regions for reaction–diffusion systems. *SIAM J. Numer. Anal.* 21(6), 1054–1065.
- Jeschke, J., Kopp, M., Tollrian, R., 2002. Predator functional responses: Discriminating between handling and digesting prey. *Ecol. Monogr.* 72(1), 95–112.
- Li, N., Steiner, J., Tang, S.-M., 1994. Convergence and stability analysis of an explicit finite difference method for 2-dimensional reaction–diffusion equations. *J. Aust. Math. Soc. Ser. B* 36(2), 234–241.
- Malchow, H., Petrovskii, S., 2002. Dynamical stabilization of an unstable equilibrium in chemical and biological systems. *Math. Comput. Model.* 36, 307–319.
- May, R., 1974. *Stability and Complexity in Model Ecosystems*. Princeton University Press, New Jersey.
- Medvinsky, A., Petrovskii, S., Tikhonova, I., Malchow, H., Li, B.-L., 2002. Spatiotemporal complexity of plankton and fish dynamics. *SIAM Rev.* 44(3), 311–370.
- Mickens, R., 2003. A nonstandard finite difference scheme for a Fisher PDE having nonlinear diffusion. *Comput. Math. Appl.* 45, 429–436.
- Morton, K., Mayers, D., 1996. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, Cambridge.
- Murray, J., 1993. *Mathematical Biology*. Vol. 19: Biomathematics Texts. Springer, Berlin.
- Neubert, M., Caswell, H., Murray, J., 2002. Transient dynamics and pattern formation: Reactivity is necessary for Turing instabilities. *Math. Biosci.* 175, 1–11.
- Pao, C., 1998. Accelerated monotone iterative methods for finite difference equations of reaction–diffusion. *Numer. Math.* 79, 261–281.
- Pao, C., 1999. Numerical analysis of coupled systems of nonlinear parabolic equations. *SIAM J. Numer. Anal.* 36(2), 393–416.
- Pao, C., 2002. Finite difference reaction–diffusion systems with coupled boundary conditions and time delays. *J. Math. Anal.* 272, 407–434.
- Pascual, M., 1993. Diffusion-induced chaos in a spatial predator–prey system. *Proc. R. Soc. Lond. Ser. B* 251, 1–7.
- Petrovskii, S., Malchow, H., 1999. A minimal model of pattern formation in a prey–predator system. *Math. Comput. Model.* 29, 49–63.
- Petrovskii, S., Malchow, H., 2001. Wave of chaos: New mechanism of pattern formation in spatio-temporal population dynamics. *Theor. Populat. Biol.* 59, 157–174.
- Petrovskii, S., Malchow, H., 2002. Critical phenomena in plankton communities: KISS model revisited. *Nonlinear Anal. Real* 1, 37–51.
- Pujol, M., Grimalt, P., 2002. A non-linear model for cerebral diffusion: Stability of finite differences method and resolution using the Adomian method. *Int. J. Numer. Method H* 13(4), 473–485.
- Rai, V., Jayaraman, G., 2003. Is diffusion-induced chaos robust? *Curr. Sci. India* 84(7), 925–929.
- Richtmyer, R., Morton, K., 1967. *Difference Methods for Initial Value Problems*. Vol. 4: Interscience Tracts in Pure and Applied Mathematics. Wiley-Interscience, New York.
- Rosenzweig, M., MacArthur, R., 1963. Graphical representation and stability conditions for predator–prey interaction. *Am. Nat.* 97, 209–223.
- Ruuth, J., 1995. Implicit–explicit methods for reaction–diffusion problems in pattern formation. *J. Math. Biol.* 34, 148–176.
- Saad, Y., 2003. Iterative methods for sparse linear systems. *SIAM*.
- Saad, Y., Schultz, M., 1986. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.* 7(3), 856–869.
- Savill, N., Hogeweg, P., 1999. Competition and dispersal in predator–prey waves. *Theor. Populat. Biol.* 56, 243–263.
- Segel, L., Jackson, J., 1972. Dissipative structure: An explanation and an ecological example. *J. Theor. Biol.* 37, 545–559.
- Sherratt, J., 2001. Periodic travelling waves in cyclic predator–prey systems. *Ecol. Lett.* 4, 30–37.
- Sherratt, J., Eagan, B., Lewis, M., 1997. Oscillations and chaos behind predator–prey invasion: Mathematical artifact or ecological reality? *Phil. Trans. R. Soc. Lond. B* 352, 21–38.
- Sherratt, J., Lambin, X., Thomas, C., Sherratt, T., 2002. Generation of periodic waves by landscape features in cyclic predator–prey systems. *Proc. R. Soc. Lond. Ser. B* 269, 327–334.

- Sherratt, J., Lewis, M., Fowler, A., 1995. Ecological chaos in the wake of invasion. *Proc. Natl. Acad. Sci. U.S.A.* 92, 2524–2528.
- Skalski, G., Gilliam, J.F., 2001. Functional responses with predator interference: Viable alternatives to the Holling type II model. *Ecology* 82(11), 3083–3092.
- Smoller, J., 1983. *Shock Waves and Reaction–Diffusion Equations*. Vol. 258: Grundlehren der mathematischen Wissenschaften. Springer-Verlag, New York.
- Stuart, A., 1989. Nonlinear instability in dissipative finite difference schemes. *SIAM Rev.* 31(2), 191–220.
- Stuart, A., Humphries, A., 1998. *Dynamical Systems and Numerical Analysis*. Vol. 2: Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge.
- Turing, A., 1952. The chemical basis of morphogenesis. *Phil. Trans. R. Soc. Lond. B* 237, 37–72.
- Yee, H., Sweby, P., 1994. Global asymptotic behavior of iterative implicit schemes. *Int. J. Bifurcat. Chaos* 4(6), 1579–1611.
- Yee, H., Sweby, P., 1995. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations II. Global asymptotic behaviour of time discretizations. *Comp. Fluid Dyn.* 4, 219–283.