Efficient simulation of complex patterns in reaction–diffusion systems

Anand L. Pardhanani, Graham F. Carey

CFD Laboratory, WRW 305, The University of Texas at Austin, Austin, TX 78712, USA

Received 9 August 1995; revised 22 January 1996

Abstract

Efficient numerical simulation techniques based on iterative and multigrid concepts are developed for solving coupled, nonlinear, reaction–diffusion systems. The solution approach is developed in a general setting and then applied to specific reaction–diffusion systems that give rise to complex dynamical patterns. The numerical strategy is based on semi-discretizing the coupled equations using a finite-difference formulation, with time integration of the resulting system of ordinary differential equations. Iterative and multigrid strategies are used to improve integration efficiency and to accelerate convergence. Numerical experiments are carried out to demonstrate the performance of the methods.

Keywords: Chemical patterns; Simulation; Multigrid; Explicit/implicit

1. Introduction

Numerical modeling and simulation is playing an increasingly important role in the study of nonlinear dynamical systems. However, the complexity of nonlinear interactions and the need for fine-scale resolution frequently prohibit computation for the time periods of practical interest. This also limits the scope of such studies, especially in problems that require simulation over a range of bifurcation parameters. This situation is improving as computer capabilities continue to grow, and as a result of advances in computational methods and algorithms. In particular, multigrid methods offer the possibility of fast solution with fine grid resolution. Recent advances in generalized gradient iterative schemes also appear promising for the reaction–diffusion problems of interest here. Hence, our focus in the present work is to develop iterative and multigrid-type solution techniques for simulating the nonlinear dynamical behavior and investigate pattern formation for a class of nonlinear reaction–diffusion systems. The solution approach and algorithms are constructed to accommodate multiple reacting species and a variety of reaction kinetics of current research interest. We demonstrate the performance of these schemes on two different applications: spiral wave formation in a model

* Corresponding author.

0377-0427/96/$15.00 © 1996 Elsevier Science B.V. All rights reserved
PHI S0377-0427(96)00028-3
Belousov–Zhabotinskii system, and the formation of self-replicating spots in the reaction–diffusion model studied in [9, 14].

The presentation is organized as follows: First the reaction–diffusion systems and mathematical models are introduced in Section 2; then in Section 3 the present discretization approach is described; our multigrid schemes are developed in Section 4, and numerical results for the specific applications using multigrid solution techniques are given in Section 5.

2. Reaction–diffusion systems

The present focus is on nonlinearly coupled reaction–diffusion systems that exhibit spatio-temporal patterns with complex dynamical behavior. Mathematical models for such systems can be written in the following general form:

\[
\frac{\partial u}{\partial t} = D \nabla^2 u + G(u) \quad \text{in } \Omega \times [t_0, t_f],
\]

\[
\frac{\partial u}{\partial n} = B(u) \quad \text{on } \partial\Omega \times [t_0, t_f],
\]

where \(u(x, t)\) is the vector of \(m\) species concentrations, \(\Omega\) represents the spatial domain with boundary \(\partial\Omega \ (x \in \Omega)\), \(D\) is the \(m \times m\) diffusion matrix, \(G\) is the nonlinear vector function of reaction kinetics and \(B\) is the vector function of boundary conditions.

In principle, this generic model accommodates any number of reacting species and holds in any spatial dimension. In the present work we focus on two spatial dimensions, because the existing experimental studies and prior numerical work has primarily dealt with this case. Even in two dimensions the numerical solution of these systems is computationally intensive. The numerical solution schemes and algorithms formulated in Sections 3 and 4 assume this generic form for the model. Validation studies to demonstrate the performance of our solution schemes are carried out on two specific applications of current interest in studies of pattern formation and their nonlinear dynamics.

The first application of interest here is the Belousov–Zhabotinskii (BZ) reaction, which models the catalytic oxidation of an organic substrate in an acidic bromate solution. It includes more than 25 reactions involving as many chemical species [17]. Its study has attracted considerable interest in the field of nonlinear dynamics, since it embodies certain essential dynamical features seen in more complex systems such as cardiac tissue, nerve tissue and amoeba colonies [15, 16]. The BZ system exhibits an intricate and complex dynamical behavior in laboratory experiments, including the formation of spiral patterns (or waves) that oscillate or rotate in complex trajectories [3, 16–19]. Fig. 1 shows a typical rotating spiral pattern seen in the experimental studies of Skinner and Swinney [16]. These spirals are, in fact, wave fronts representing steep concentration gradients in the reacting species. As the spiral moves, its tip traces a trajectory whose complexity depends on the reaction parameters. Thus, the tip trajectory serves as an indicator of the dynamical behavior of the spirals.

Modeling the complete reaction kinetics of the BZ system would be extremely complex, and is presently beyond the scope of practical computations. This fact, together with the need to elucidate the essential dynamical features, has led to simplified models to qualitatively study the spatiotemporal
patterns. A standard model that reproduces many of the experimental observations is the two-variable Oregonator system, and is described in [7, 3, 15]. It can be obtained from (1) by setting $u = [u, v]^T$, $x = [x, y]^T$ and

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{u - u^2 - f v (u - q)}{(u + q)} \end{bmatrix}, \quad B = 0.$$ (2)

Here $u$ and $v$ represent the concentration of two chemical species with diffusion coefficients $d_1$ and $d_2$, and $f$, $q$ and $\varepsilon$ are constant reaction parameters. Typical (nondimensional) values of these parameters are $d_1 = 1.0$, $d_2 = 0.6$, $\varepsilon = 0.01$, $q = 0.002$, and $0 \leq f \leq 4$. Precise values of the reaction parameters — which are also referred to in the literature as “recipe” parameters — are difficult to estimate and remain a subject of debate. It is known, however, that the values strongly influence the nature of the patterns and their dynamical behavior.

For certain ranges of the recipe parameters, the solutions to the Oregonator model exhibit features very similar to those seen in BZ experiments. Contour plots of the solution component $u$ have spiral shapes that move in similar trajectories to those observed. This suggests that the Oregonator model provides a convenient mechanism for studying the spiral patterns and their dynamical behavior, and is one focus of the present work.

Another reaction–diffusion application that will be considered here is the model used in [10, 14] which leads to the formation of self-replicating spots and stripes. The variety of these patterns, with their ability to indefinitely replicate and annihilate themselves, has drawn speculation that such systems embody the basic dynamics seen in more complex natural systems which exhibit life-like phenomena such as birth and death (e.g., cell division). The mathematical model is similar to (2),
except that the reaction kinetics are defined by

$$G = \begin{bmatrix} -uv^2 + a(1 - u) \\ uv^2 - bv \end{bmatrix},$$

(3)

and the constant reaction parameters in this model typically have the following non-dimensional values: \( d_1 = 2.0 \times 10^{-5} = 2 d_2 \), \( 0.01 \leq a \leq 0.06 \), \( 0.04 + a \leq b \leq 0.06 + a \). Depending on the choice of the control parameters \( a \) and \( b \), this model exhibits a variety of dynamical patterns as evident in the contour plots of \( u \) and \( v \) [14]. An initial perturbation is typically required to trigger these patterns, which then sustain themselves indefinitely. Further details concerning their behavior and evolution will be discussed in the section on numerical results.

3. Numerical formulation

A major difficulty in computing solutions to these PDE systems arises from “stiffness” due to both the reaction kinetics and the need to use fine grids to obtain meaningful solutions. Accordingly, we focus on methods that use meshes of fine resolution, but preserve good efficiency. To this end, our first objective is to explore the applicability of multilevel iterative strategies for efficiently solving the class of reaction–diffusion problems described by (1). To ensure generality, we assume the system involves two or more reacting species that are coupled via strongly nonlinear reaction kinetics. We also avoid any simplifications or approximations in the model or computational procedure that are based on a priori knowledge of the solution physics (which has been used for economy in previous studies [1, 3]).

In our numerical formulation, the PDEs are discretized and solved in a fully coupled form. Spatial derivatives are discretized using second-order symmetric differencing. This results in the following system of ODEs at a representative grid point \((i, j)\):

$$\frac{du_{i,j}}{dt} = D(A_{i,j}u_{i,j}) + G_{i,j}$$

(4)

with

$$A_{i,j}u_{i,j} = \frac{1}{\Delta x^2}[u_{i+1,j} - 2u_{i,j} + u_{i-1,j}] + \frac{1}{\Delta y^2}[u_{i,j+1} - 2u_{i,j} + u_{i,j-1}],$$

(5)

where \((\Delta x, \Delta y)\) is the mesh spacing, \(u_{i,j} = [u_{i,j}, v_{i,j}, \ldots]^T\), \(G_{i,j} = G(u_{i,j})\), and \(D\) is the local diffusion matrix in (1).

Equations of the form (4) are accumulated over the entire grid, along with boundary conditions, to yield the globally coupled system of ODEs,

$$\frac{dU}{dt} = [A]U + G(U),$$

(6)

where \(U\) is the vector of unknowns, \([A]\) is the global coefficient matrix of the diffusion terms, and \(G\) is the global vector arising from the nonlinearly coupled reaction terms. This global system can be integrated in time using explicit or implicit methods in conjunction with the multigrid techniques described in the following section.
4. ODE solution with multigrid

Nonlinearly coupled ODE systems of the form (6) can be numerically integrated using implicit or explicit techniques. However, for the fine grid resolution and general reaction kinetics of interest here, the ODE systems are often stiff, so the step size for explicit methods may be limited by stability requirements rather than accuracy. The stiffness increases as the mesh scale is further refined. Implicit schemes have better stability properties, but they require algebraic system solution at each integration step and, therefore, entail higher computational cost per step. This is precisely where multigrid methods can be exploited to advantage. If the large algebraic systems that arise at each step during implicit integration can be solved much faster using multigrid techniques, then this will improve the overall efficiency of the solution scheme considerably. Furthermore, the fine grid time accuracy of the integration scheme will remain unaffected by the multigrid acceleration. This is not true of the explicit multilevel approach, however, since the actual time step traversed varies from one grid to the next. In fact, the present explicit multigrid scheme is a viable choice only for certain applications where the stiffness of the system on the fine grid dictates the time step.

4.1. Implicit integration

To illustrate our basic approach, we consider the backward Euler method and a second-order semi-implicit Runge–Kutta method for performing the numerical integration. For convenience, we rewrite the ODE system (6) as

\[
\frac{dU}{dt} = F(U),
\]

where \( F(U) = [A]U + G(U) \). Then, a backward Euler scheme applied to (7) yields

\[
[I - \Delta t J_n](U_{n+1} - U_n) = \Delta t F(U_n),
\]

where \( I \) is the identity matrix, \( J_n = \frac{\partial F}{\partial U_n} \) is the Jacobian matrix, and the subscripts denote time steps. A two-stage semi-implicit Runge–Kutta method applied to (7) can be written in the form (see, e.g., [8])

\[
U_{n+1} = U_n + \Delta t (\alpha_1 k_1 + \alpha_2 k_2),
\]

where

\[
k_1 = [F(U_n) + \Delta t a_1 J(U_n) k_1],
\]

\[
k_2 = [F(U_n + \Delta t b_1 k_1) + \Delta t a_2 J(U_n + \Delta t c_1 k_1) k_2].
\]

Here \( \alpha_1, \alpha_2, a_1, a_2, b_1 \) and \( c_1 \) are constants that depend upon the specific integration scheme under consideration. For example, the second-order Rosenbrock version of the semi-implicit scheme used
in the present work corresponds to the following choice of constants:

\[
\begin{align*}
 a_1 &= a_2 = 1 - \frac{\sqrt{2}}{2}, \\
 b_1 &= \frac{\sqrt{2} - 1}{2}, \\
 c_1 &= 0, \quad \alpha_1 = 0, \quad \alpha_2 = 1
\end{align*}
\] (11)

Observe that both integration schemes require solution of linear algebraic systems during each step. These linear systems are large, sparse and banded, for the natural grid-point ordering used here. Although sparse elimination-based solvers are very robust, for large sparse systems they are often far more computationally expensive than iterative solvers, particularly if multigrid acceleration of the iteration is possible. Consequently, we have developed multigrid algorithms that rely on several embedded meshes for solving the linear systems in (8)-(10). The increased computational efficiency attained using this approach is demonstrated in the numerical studies in Section 5.

A basic introduction to multigrid methods and related concepts can be found in standard references including, for example, [2, 4, 11, 20]. To illustrate the present approach, consider solving the algebraic systems in (8) or (9)-(10) on a given grid \(\Omega^h\) of spacing \(h\). For convenience, we rewrite the algebraic systems in the following standard form:

\[ A^h u^h = f^h. \] (12)

A basic two-grid multigrid cycle can be constructed by introducing a nested coarser grid \(\Omega^{2h} \subset \Omega^h\), and performing the following sequence of steps:

1. Perform a few smoothing iterations (e.g., Gauss–Seidel) on (12) starting from initial guess on grid \(\Omega^h\); obtain approximate solution \(u^h\).
2. Compute the residual \(r^h = f^h - A^h u^h\), restrict \(r^h\) and \(A^h\) to the coarse grid (\(\Omega^{2h}\)) and define the following coarse grid problem:

\[ A^{2h} e^{2h} = r^{2h}. \]

3. Solve for \(e^{2h}\).
4. Interpolate \(e^{2h}\) to the fine grid and improve fine grid approximation

\[ e^{2h} \rightarrow e_{\text{new}}^{2h}, \]

\[ u_{\text{new}}^h = u^h + e_{\text{new}}^h. \]

5. Perform a few more smoothing iterations on (12) using \(u_{\text{new}}^h\) as initial guess.

The above two-grid cycle can be recursively generalized to include several nested grid levels. A variety of strategies exist for cycling between these grids. In the present work we use standard \(V\)-cycles. The inter-grid transfer operations in steps 2 and 4 above are performed using standard techniques (see, e.g., [11]). The restriction from \(\Omega^h\) to \(\Omega^{2h}\), which is denoted by the operator \(I_h^{2h}\) (e.g., \(r_{2h} = I_h^{2h} r_h\)), is performed using the following full-weighting procedure:

\[
\begin{align*}
 r_{2h}^{(i,j)} = \frac{1}{16} &\left\{ 4r_{(2i,2j)}^h + 2(r_{(2i+1,2j)}^h + r_{(2i-1,2j)}^h + r_{(2i,2j+1)}^h + r_{(2i,2j-1)}^h) \\
 &+ r_{(2i+1,2j+1)}^h + r_{(2i-1,2j+1)}^h + r_{(2i+1,2j-1)}^h + r_{(2i-1,2j-1)}^h \right\}.
\end{align*}
\] (13)
Here $i$ and $j$ correspond to grid indices on $\Omega^{2h}$, with $r^{2h} = [..., r_{i,j}^{2h},...]^T$ and $r^h = [..., r_{i,j}^{h},...]^T$. The matrix $A^h$ is restricted to the coarser grid by repeating the spatial discretization of the original PDE on the new grid level. The transfer in the reverse direction (from $\Omega^{2h}$ to $\Omega^h$), which is denoted as $I_{2h}^h$ (e.g., $e_2^h = I_{2h}^h e^{2h}$), is performed using bilinear interpolation.

### 4.2. Explicit integration

In certain cases, multigrid techniques can also be used in conjunction with explicit integration schemes to increase their effective stability range (see, e.g., [5, 13]). Such schemes are relatively new, and their behavior is not as well understood as that of implicit multigrid schemes. In addition, the explicit approach is less straightforward, and in the multilevel context it is generally applied to extend the stability domain in problems where such restrictions are severe. For the present application, we have conducted some preliminary investigations using an explicit multilevel integration approach. The basic integration scheme used in this multilevel algorithm is the following second-order Runge-Kutta-Fehlberg form [8]

$$U_{n+1} = U_n + \frac{\Delta t}{891} (214k_0 + 27k_1 + 650k_2), \quad (14)$$

where

$$k_0 = F(U_n),$$

$$k_1 = F \left( U_n + \frac{\Delta t}{4} k_0 \right),$$

$$k_2 = F \left( U_n + \frac{\Delta t}{800} (-189k_0 + 729k_1) \right),$$

$$k_3 = F \left( U_n + \frac{\Delta t}{891} (214k_0 + 27k_1 + 650k_2) \right). \quad (15)$$

The truncation error is estimated using the third-order approximation

$$\hat{U}_{n+1} = U_n + \frac{\Delta t}{2106} (533k_0 + 1600k_2 - 27k_3) \quad (16)$$

as

$$\tau = |\hat{U}_{n+1} - U_{n+1}|. \quad (17)$$

A two-grid cycle using an explicit scheme of the form (14) applied to the fine grid ODE system

$$\frac{dU^h}{dt} = F^h(U^h) \quad (18)$$

can be constructed as follows [12]:

1. Perform a few integration steps of size $\Delta t^h$ on the fine grid; e.g., if only one step is taken we have

$$U_{n+1}^h = U_n^h + \Delta t^h (\alpha_0 k_0^h + \alpha_1 k_1^h + \alpha_2 k_2^h)$$
where \( \alpha_i \) and \( k_i \) are the weights and function values in the multistage scheme; this takes the solution from \( t_n \) to \( t_n + \Delta t^h \).

2. Proceed to the coarse grid and take another step of length \( \Delta t^{2h} > \Delta t^h \):

\[
U_{n+2}^{2h} = \hat{I}_{h}^{2h} U_{n+1}^{h} + \Delta t^{2h}(\alpha_0 k_0^{2h} + \alpha_1 k_1^{2h} + \alpha_2 k_2^{2h}),
\]

where \( k_i^{2h}, i = 0, 1, 2, \) are now defined with respect to

\[
F^{2h}(U^{2h}) + L^2h(I_h^{2h}U_{n+1}^h, t) - I_h^{2h}L^h(U_{n+1}^h, t)
\]

with the operator \( L \) defined as

\[
L^h(U^h, t) = \frac{dU^h}{dt} - F^h(U^h)
\]  

(19)

This takes the coarse grid solution from \( t_n + \Delta t^h \) to \( t_n + \Delta t^h + \Delta t^{2h} \).

3. Interpolate and compute fine grid solution at \( t_{n+2} \),

\[
U_{n+2}^{h} = U_{n+1}^{h} + I_{2h}^{h}[U_{n+2}^{2h} - I_{h}^{2h}U_{n+1}^{h}]
\]

At the end of the above two-grid cycle, the solution has traversed in time from \( t = t_n \) to \( t = t_n + \Delta t^h + \Delta t^{2h} \). Thus, the cumulative step size is \( \Delta t^h + \Delta t^{2h} \). This two-grid method can be generalized to multiple grids.

5. Results and discussion

The numerical simulations reported in this section focus on the two reaction–diffusion applications described in Section 2. First, the formation of spiral waves in the BZ system is simulated using the Oregonator model (2) with the indicated parameter values. The simulation domain is a square of side 15 units and, following Jahnke et al. [3], the initial conditions are taken as

\[
\begin{aligned}
0.8 & \quad \text{if } 0 < \theta < 0.5, \\
\frac{q(f+1)}{(f-1)} & \quad \text{elsewhere,}
\end{aligned}
\]

\[
v = \frac{q(f+1)}{(f-1)} + \frac{\theta}{8\pi f},
\]  

(20)

where \( \theta \) is the polar coordinate angle with origin at the center of the domain. The spiral patterns are generated for bifurcation parameter values \( f = 1.4 \) and \( f = 3.0 \).

The numerical results are computed primarily using the implicit integration schemes described in Section 4 in conjunction with multigrid solution of the algebraic systems. The multigrid algorithm for solving the algebraic systems in the implicit scheme is based on \( V \)-cycles involving 5 nested grid-levels with: collective Gauss–Seidel smoothing iterations (2 pre-smoothing and 1 post-smoothing...
iteration); full-weighted restriction and bilinear interpolation for prolongation; and band solution at the coarsest grid level. Collective Gauss–Seidel smoothing is a block-iterative strategy designed for fully coupled formulations. It involves updating all the unknowns at a given grid point simultaneously. This is very similar to the standard Gauss–Seidel method, the only difference being that the global matrix and vectors are partitioned into blocks of length \( m \) (since there are \( m \) unknowns at each grid point). Such a partition (with \( m = 2 \)) arises naturally in our fully coupled discretization approach. A highly efficient, vectorized version of this smoothing algorithm is developed for use with the multigrid solver.

When the numerical simulation is carried out starting from the initial conditions (20), the contours evolve through a transient stage into a spiral pattern which orbits in a periodic trajectory. Fig. 2 shows the evolution of the species \( u \) during a representative orbit, for the case \( f = 1.4 \). The figure shows "snapshots" of the concentration density, at intervals of 0.44 time units as the solution evolves over a time period of about 1.75 units. These results were computed with a step size of 0.001 units in the backward Euler scheme, using a uniform \( 65 \times 65 \) fine grid, with uniform embedded coarser grids for the multigrid solver. When the parameter \( f \) is increased to 3.0, the characteristics of the spiral pattern change and its tip traces a more complex trajectory. This is because the solution undergoes a bifurcation between \( f = 1.4 \) and 3.0. Fig. 3 shows a representative evolution sequence of \( u \) for this case, analogous to Fig. 2.

The motion of the spiral tip is often monitored in BZ experiments, since it is a useful indicator of the dynamical behavior of the spiral patterns. In our numerical experiments the tip trajectory is automatically computed during the simulation. We observe that when the bifurcation parameter \( f \) is set to 1.4 the spiral tip traces a circular trajectory, but when \( f \) is increased to 3.0 its tip traces a more complex trajectory. Fig. 4 shows the tip trajectories for \( f = 1.4 \) and \( f = 3.0 \).

To evaluate the performance gain with our multigrid-based numerical schemes, sample results were also computed using an optimized band solver in the place of the multigrid solver. In this problem, the speed of execution is nearly 40 times faster when the multigrid solver is used instead of the band solver. This is despite the fact that a highly efficient and vectorized band solver, running at a sustained rate of about 100 MFLOPS (million floating point operations per second) on the Cray Y-MP, was developed for comparison purposes. Performance statistics comparing the multigrid and band solver for a sample run are summarized in Table 1. This problem strikingly illustrates the disparity between band solvers and multigrid solvers when the number of unknowns gets large. In fact, if the number of points on the fine grid were increased further, the disparity would get even larger. (Incidentally, this example also emphasizes that a superior megaflop rate — or gigaflop rate on current supercomputers — can be misleading insofar as true computational cost is concerned if the underlying algorithm is not efficient.)

Some numerical studies were also conducted using the multilevel explicit scheme described in Section 4.2. In preliminary tests using this approach it was observed that the accuracy required on the fine grid limited the number of permissible levels in the multigrid algorithm significantly. Yet it does seem feasible to see some performance improvements even with two grid levels. For example, sample results with a two-level Runge–Kutta–Fehlberg scheme in which the combined step size was fixed at 0.00075 units showed about 20% performance improvement over the previous implicit multigrid scheme.

Numerical simulations have also been carried out using the reaction–diffusion model described by Eq. (3). In this problem the domain \( \Omega \) is a square of side 2.5 units, and initial conditions are
Fig. 2. Concentration density plots of $u$ over a time period $t = t_0$ to $t_0 + 3\Delta t$ with $\Delta t = 0.44$ and $f = 1.4$. Density values range from 0 (lighter shade) to 0.8 (darker shade).

taken as

$$u = \begin{cases} 0.3 & \text{if } (x, y) \in \hat{\Omega}, \\ 1.0 & \text{otherwise}, \end{cases}$$

$$v = \begin{cases} 0.25 & \text{if } (x, y) \in \hat{\Omega}, \\ 0 & \text{otherwise}, \end{cases}$$

(21)
where $\hat{\Omega}$ is a square subdomain of length 0.2 units centered within $\Omega$. These initial conditions, with parameter values $a = 0.02$ and $b = 0.079$ in Eq. (3), lead to an evolving pattern of spots, which replicate themselves through a series of divisions.

Numerical results for this problem were computed using multilevel implicit integration involving several nested grid levels. We first attempted to compute the solution using $65 \times 65$ nodes on the finest grid, but found this yields a fictitious solution: both $u$ and $v$ rapidly converge to the simple
steady-state pattern shown in Fig. 5. However, when the resolution on the finest grid is increased to $129 \times 129$, the solution yields a sequence of dynamically evolving patterns that is very similar to those observed in the experiments in [10]. This result also emphasizes the need for fine grid resolution and the sensitivity of the computation to the grid. Fig. 6 shows the evolution of the species $u$ at selected time instants starting from the initial condition (21). These results were computed using a time step of 10 units in the implicit integration procedure, with multigrid solution of the algebraic systems. The $V$-cycle details are similar to those used in the BZ application: 6 nested grid levels, collective Gauss–Seidel smoothing, 2 pre-smoothing and 1 post-smoothing iteration, full-weighted restriction and bilinear interpolation.

In this problem the initial perturbation decays to a simple steady state for a wide range of parameter values [14]. Another set of parameter values for which we found more interesting patterns was: $a = 0.036$ and $b = 0.093$. The concentration contours in this case also evolve into a series of spots but, unlike the previous case, they do not replicate themselves through division. They appear behind

---

**Table 1**

<table>
<thead>
<tr>
<th></th>
<th>Multigrid solver</th>
<th>Band solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall CPU time (s)</td>
<td>20.1</td>
<td>848.8</td>
</tr>
<tr>
<td>Overall MFLOPS</td>
<td>90</td>
<td>106</td>
</tr>
<tr>
<td>MFLOPS in solver routine</td>
<td>88.3</td>
<td>106.5</td>
</tr>
<tr>
<td>CPU % in solver routine</td>
<td>31</td>
<td>98.3</td>
</tr>
</tbody>
</table>

Fig. 4. Trajectories showing spiral tip motion for $f = 1.4$ (solid curve) and $f = 3.0$ (dashed curve).
the initial concentration front as it propagates out toward the domain boundaries, and they seem to retain temporal stability after their formation. Fig. 7 shows the evolution of $u$ at selected time instants for this case.

6. Concluding remarks

Detailed numerical study of the evolution and dynamical behavior of chemical patterns in reaction–diffusion systems is complicated by the need for fine grid resolution and the stiffness of the associated ODE systems. As seen in the replicating spot problem, insufficient grid resolution may even lead to the "wrong" solution, i.e., the true pattern may not be realized. In the present work we demonstrate how multigrid strategies can be used within both implicit and explicit frameworks to circumvent this problem and yet provide a more efficient algorithm. Our numerical approach is based on semi-discretizing the partial differential equation system using finite differences, with time integration of the resulting system of ordinary differential equations. Multigrid techniques are used to perform the integration more efficiently and accelerate the overall convergence. Numerical validation studies are carried out using the Oregonator model for the Belousov–Zhabotinskii reaction, as well as a new model that yields a pattern of self-replicating spots. We have simulated the characteristic BZ spiral patterns and the tip trajectories for two representative values of the bifurcation parameter. For implicit integration schemes we have shown that multigrid techniques work well, and they significantly enhance the computational efficiency. Furthermore, the implicit multigrid approach does not degrade the integration accuracy on the fine grid. Results for the second application also indicate
Fig. 6. Contour plots of $u$ on $129 \times 129$ fine grid at selected time instants. Contour values range from 0.25 (lightest shade) to 1.0 (darkest shade).
Fig. 7. Contour plots of $u$ for parameter values $a=0.036$, $b=0.093$, with $129 \times 129$ fine grid. Contour values range from 0.25 (lightest shade) to 1.0 (darkest shade).
that this multigrid approach works quite well. Exploratory studies have also been conducted using an explicit multilevel scheme.

The schemes developed here begin to bring this class of reaction–diffusion problems within the scope of practical computation, and are promising for larger systems involving more variables as well as future three-dimensional analysis. Improvements in the explicit approach, particularly in the implementation of boundary conditions, are currently under investigation. An alternative approach based on multigrid acceleration of waveform-relaxation-type integration schemes is also being explored. This approach would be attractive from the standpoint of parallel computation.

Acknowledgements

We thank Dr. Harry Swinney at the University of Texas Center for Nonlinear Dynamics for providing experimental results and analytical models that were used in this work.

References

