International Journal of Bifurcation and Chaos, Vol. 7, No. 11 (1997) 2529–2545

FAST SIMULATIONS OF WAVES IN THREE-DIMENSIONAL EXCITABLE MEDIA

MATTHEW DOWLE, ROLF MARTIN MANTEL and DWIGHT BARKLEY*
Mathematics Institute, University of Warwick, Coventry CV4 7AL UK

Received January 6, 1997; Revised February 21, 1997

A fast numerical scheme based on the model of Barkley [Physica **49D** (1991), 61] is extended to three space dimensions (3D). The original time-stepping scheme is improved to provide greater accuracy and a 19-point approximation for the Laplacian operator in 3D is shown to have significant advantages over the commonly used 7-point formula. Simulations are coupled to a state-of-the-art surface rendering algorithm such that the combined code allows real-time interactive simulations of 3D waves on a desktop workstation. Results are presented from simulations over a range of spatio-temporal resolutions, from coarse cellular-automaton type simulations to fully resolved simulations of the underlying partial differential equations.

1. Introduction

Experimental and numerical work suggests that scroll waves in three-dimensional cardiac tissue are, at least in some cases, at the origin of sudden-cardiac death, e.g. [Winfree, 1987, 1994; Davidenko *et al.*, 1992, Gray *et al.*, 1995, Panfilov & Hogeweg, 1995; Panfilov & Keener, 1995]. One basic problem in the numerical study of these waves is that cardiac tissue is an excitable medium and therefore waves in the medium involve very disparate time and space scales. The time scale of excitation (the action potential) is orders of magnitude faster than the time scale of recovery. Similarly, the space scale of an excitation front is orders of magnitude smaller than the size of the medium. This puts heavy computational demands, both in memory and execution time, on numerical simulations of waves in three space dimensions (3D) and effectively rules out, at the current time, full simulations of cardiac tissue using accepted physiological models. Even a model such as the Beeler and Reuter [1977] model which is quite simple physiologically, would be extremely time-consuming to simulate in three space dimensions.

Because of the need to obtain at least qualitatively accurate simulations of waves under such demanding conditions, several cellular automaton (CA) models of excitable media were proposed in the early 1990s [Markus & Hess, 1990; Gerhardt *et al.*, 1990a, 1990b, 1990c, 1991]. Weimar *et al.* [1992a, 1992b] further extended the CA approach and showed how CA rules could be matched to specific reaction-diffusion equations. At about the same time, a reaction-diffusion model of essentially FitzHugh–Nagumo type was proposed by Barkley [1991] which allows for time stepping over a range of spatio-temporal resolutions. In the coarse limit the time-stepping scheme becomes similar to the CA model of Weimar *et al.*, while in the fine limit the scheme gives accurate solutions to the underlying partial-differential equations. Henze and Tyson [1996] have recently extended the CA model of Weimar *et al.* to 3D and have addressed quite clearly many of the issues which arise when simulating waves in 3D.

*Corresponding author. E-mail: barkley@maths.warwick.ac.uk

Here we describe an extension of the Barkley model to 3D and show how the original time-stepping scheme can be improved to obtain more accurate solutions. We show that reasonable results are obtainable at speeds exceeding 10 times what is required for fully resolved simulations. This makes it feasible to study the dynamics of many of the simplier 3D stuctures in excitable media on desktop workstations with real-time interactive graphics.

2. Model and Methods

Our starting point is the general two-variable system of reaction-diffusion equations:

$$\frac{\partial u}{\partial t} = f(u, v) + \nabla^2 u, \quad \frac{\partial v}{\partial t} = g(u, v) + D\nabla^2 v\,, \tag{1}$$

where the functions $f(u, v)$ and $g(u, v)$ express the local reaction kinetics of the two variables u and v. When modeling excitable media, the variable u is fast in comparison to the variable v. By choice of length scales, we set the diffusion coefficient for the u-variable to unity. The parameter D is then the ratio of diffusion coefficients. In the case of cardiac tissue and other physiological media $D = 0$, and this is the only case we shall consider in detail here.

Our approach can be applied to models of the form (1) in which the function $f(u, v)$ can be represented by:

$$f(u, v) = \frac{1}{\varepsilon} u(1 - u)(u - u_{\rm th}(v))h(v)\,, \tag{2}$$

where $u_{\rm th}(v)$ and $h(v)$ are functions of v only and ε is a small parameter. (The function f lacks the $-v$ term of the classical FitzHugh–Nagumo equations.) The function $g(u, v)$ can be arbitrary so long as it is slow, i.e. order 1, compared to the function $f(u, v)$, which is order $1/\varepsilon$. In fact, because there are essentially no restrictions on the slow dynamics, there may be more than one slow variable. Then $u_{\rm th}$ and h could be functions of all the slow variables.

The simplest choice for the functions g, $u_{\rm th}$, and h is:

$$g(u, v) = u - v, \quad u_{\rm th}(v) = (v + b)/a, \quad h(v) = 1 \tag{3}$$

where a and b are parameters. This is the form we shall consider here, though we hope in the future to use other choices which more closely mimic cardiac electrophysiology.

2.1. *Reaction terms*

We begin by considering just reaction terms $f(u, v)$ and $g(u, v)$ in Eq. (1). This follows the treatment of the 2D case presented in [Barkley *et al.*, 1990, Barkley, 1991]. The two important aspects of the fast numerical scheme are: (1) the approximation of u by zero whenever u becomes sufficiently close to zero, and (2) the use of semi-implicit time stepping for u.

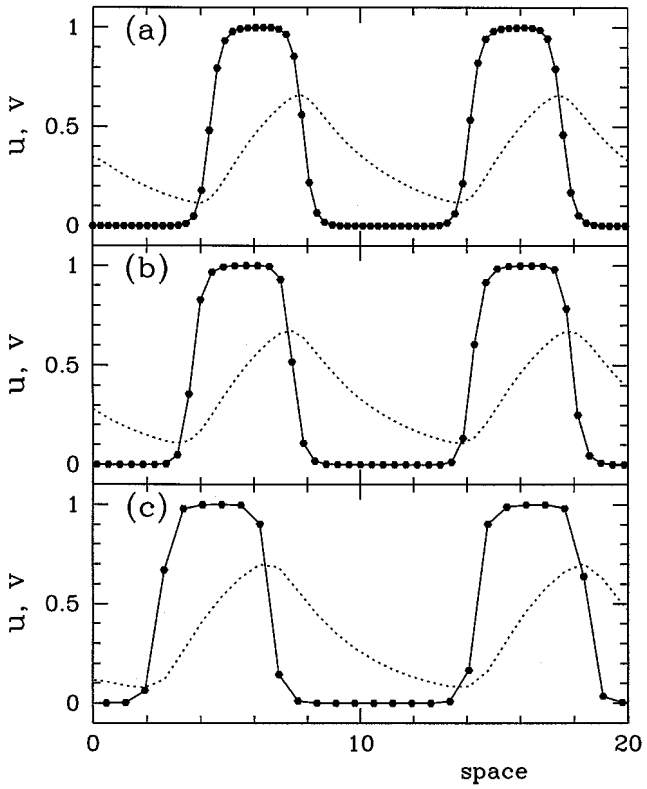


Fig. 1. One-dimensional cuts through the model excitable medium showing the profile of typical waves. Plotted are the u-field (points connected by solid curve) and the v-field (dashed curve). The three cases correspond to simulations at different computational grid spacings considered throughout the paper: (a) $h = 2/7$, (b) $h = 3/7$, (c) $h = 5/7$. The grid spacing determines the number of points representing the sharp "interface" between $u \simeq 0$ and $u \simeq 1$. The waves travel from right to left and so a time series at a fixed spatial location looks essentially the same except that the number of points on the "interface" is set by the time step Δt. Model parameters in all three cases are: $a = 0.8$, $b = 0.01$, $\varepsilon = 0.02$, and $D = 0$.

2.1.1. *Setting* $u = 0$

Figure 1 shows typical spatial profiles for the u- and v-fields from simulations of the full reaction-diffusion system (1). It can be seen that u is quite small (very near zero) over approximately 50% of space. This is parameter-dependent of course, but 50% is typical and in fact it is quite common for the flat regions to occupy 70% or more of the domain. The crucial point is that there is no need to carefully follow the dynamics of u once it becomes very close to zero. Hence we introduce a small numerical parameter, δ, where typically δ is in the range 10^{-5} to 10^{-3}, and then simply set $u = 0$ whenever $u < \delta$.

This leads to the following simple algorithm for time stepping the reaction terms:

$$\begin{aligned}
&\text{if } (u^n < \delta) \\
&\quad \{ u^{n+1} = 0 \\
&\qquad v^{n+1} = v^n + \Delta t\, g(0,\, v^n) \} \\
&\text{else} \\
&\quad \{ u_{\text{th}} = u_{\text{th}}(v^n) \\
&\qquad v^{n+1} = v^n + \Delta t\, g(u^n,\, v^n) \\
&\qquad u^{n+1} = F(u^n,\, u_{\text{th}}) \}
\end{aligned} \tag{4}$$

where u^n and v^n are the values of variables u and v at the nth time step and Δt is the time step. We shall use $u_{\text{th}}(v)$ to refer to the function of v and u_{th} to refer to a variable storing its value at a particular value of v. The function F specifies the stepping of u when $u > \delta$ and will be discussed shortly.

The variable v is stepped by the explicit-Euler method, with the condition that u is set to zero for $u < \delta$. Because the time-scale of v is slow in comparison with the fast time-scale of u (see Fig. 1), even with time steps that are large relative to the u-time scale, explicit-Euler stepping of v is both stable and accurate.

2.1.2. *Semi-implicit time stepping*

Here we consider how best to choose the function F for stepping the u-dynamics. This choice can have significant impact on the overall accuracy and speed of simulations. The simplest choice for F is explicit-Euler time stepping given by:

$$\begin{aligned}
u^{n+1} &= F^{(e)}(u^n,\, u_{\text{th}}) \\
&= u^n + \Delta t\, f(u^n,\, v^n) \\
&= u^n + (\Delta t/\varepsilon)\, u^n(1-u^n)(u^n - u_{\text{th}})\,,
\end{aligned} \tag{5}$$

where the one-step error is $O(\Delta t^2)$. [The global error is $O(\Delta t^1)$]. With explicit-Euler time stepping, $\Delta t/\varepsilon \simeq 1$ is the maximum time step for obtaining a reasonable simulation of the fast dynamics (because of the $1/\varepsilon$ time scale of the u-equation, it is natural to measure time steps in units of $1/\varepsilon$). A time step with $\Delta t/\varepsilon \simeq 1$ is also near the stability limit for an explicit-Euler step and so it is not possible to take time steps much larger than this.

In order to take larger time steps, a semi-implicit form for F can be used. The simplest form is obtained from:

$$u^{n+1} = \begin{cases} u^n + (\Delta t/\varepsilon) u^{n+1}(1-u^n)(u^n - u_{\text{th}}) & \text{if } u^n \le u_{\text{th}} \\ u^n + (\Delta t/\varepsilon) u^n(1-u^{n+1})(u^n - u_{\text{th}}) & \text{if } u^n > u_{\text{th}}\,. \end{cases}$$

where u at the future time is used in those factors on the right-hand side that undergo largest relative change over the time step. Solving the above expressions for u^{n+1}, we obtain:

$$u^{n+1} = F^{(i)}(u^n,\, u_{\text{th}}) = \begin{cases} \dfrac{u^n}{1 - (\Delta t/\varepsilon)(1-u^n)(u^n - u_{\text{th}})} & \text{if } u^n \le u_{\text{th}} \\ \dfrac{u^n + (\Delta t/\varepsilon) u^n (u^n - u_{\text{th}})}{1 + (\Delta t/\varepsilon) u^n (u^n - u_{\text{th}})} & \text{if } u^n > u_{\text{th}} \end{cases} \tag{6}$$

By expanding the denominators in the above expressions it can be seen that this scheme agrees with the explicit-Euler scheme to $O(\Delta t^2)$ and hence it also has a one-step error of $O(\Delta t^2)$. This is the implicit form proposed in [Barkley, 1991].

Unlike the explicit form, the semi-implicit form is numerically stable for arbitrarily large values of Δt and in the limit of large $\Delta t/\varepsilon$, it goes over to:

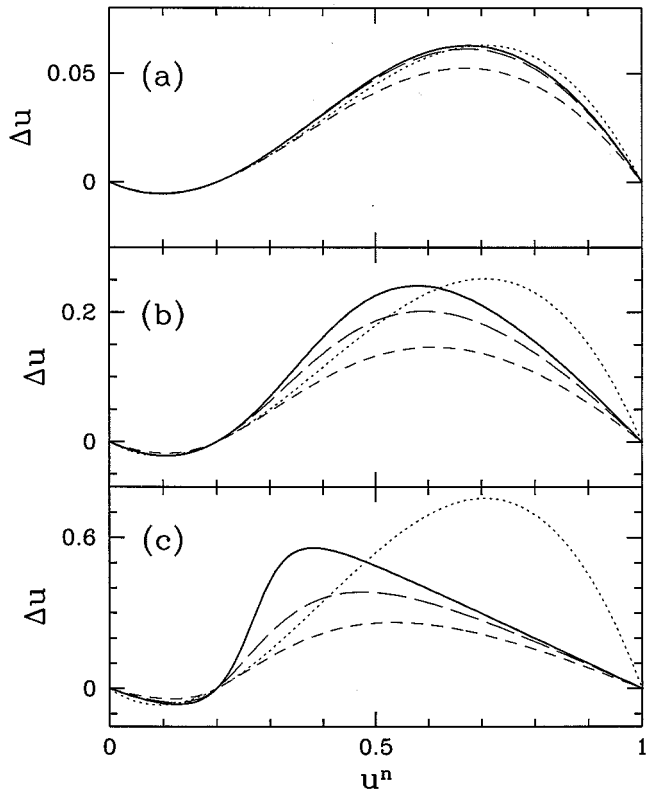


Fig. 2. Comparison of the various time-stepping schemes considered in the text. Shown is $\Delta u \equiv u^{n+1} - u^n$ as function of u^n for exact (solid), explicit-Euler (dotted), second-order implicit (dashed), and new third-order implicit (long dashed) methods. (a) Small time step, $\Delta t/\varepsilon = 0.6$. The second-order implicit method is not as good as the explicit-Euler; the third-order implicit scheme is barely distinguishable from the exact solution. (b) moderate time step, $\Delta t/\varepsilon = 2.4$. The explicit-Euler scheme leads to numerical instability, while both implicit schemes remain stable. (c) large time step, $\Delta t/\varepsilon = 7.2$. The implicit schemes begin to approach the large-time-step (CA) limit of a piecewise linear function.

$$u^{n+1} = \begin{cases} 0 & \text{if } u^n < u_{\rm th} \\ u_{\rm th} & \text{if } u^n = u_{\rm th} \\ 1 & \text{if } u^n > u_{\rm th}\,. \end{cases} \tag{7}$$

The large-Δt limit is thus very similar to the CA model Weimar *et al.* [1992a, 1992b] in which the u-variable takes on just two values: 0 and 1. In this limit u jumps between 0 and 1 in one time step, and when space is included, u jumps between 0 and 1 over a single grid spacing. The coarsest simulations that we consider are not quite at this limit: u generally takes a few space and time steps to go from $u \simeq 0$ to $u \simeq 1$ [see Fig. 1(c)]. The fact that u is not discontinuous in our fast simulations is a significant advantage of our approach over the CA approach.

Figure 2 illustrates various time-stepping schemes for different values of $\Delta t/\varepsilon$. Plotted is the change in u over one time-step, $\Delta u \equiv u^{n+1} - u^n$, as a function of u^n, with v^n and hence $u_{\rm th}$ is fixed. A moderately small value is chosen for $u_{\rm th}$ corresponding to a typical value at the excitation front. The "exact" Δu is obtained by accurately numerically solving $\dot{u} = f(u, v = \text{const})$ over the time interval Δt.

It can be seen that while the implicit form $F^{(i)}$ has the advantage of numerical stability for large values of $\Delta t/\varepsilon$, it is actually less accurate than the explicit form $F^{(e)}$ for small values of $\Delta t/\varepsilon$. This is accentuated by the fact that the implicit form systematically errs by making Δu too small, whereas in the explicit form Δu errs in both directions. The net effect is that the time for u to cross the fast region between $u = u_{\rm th}$ and $u = 1$ in the implicit case is always smaller than the true time. Hence while both the explicit and implicit scheme are of the same order of accuracy, the numerical pre-factor is such that the explicit scheme is superior in its range of stability.

We propose a rather simple improvement to implicit scheme (6) which leads to significantly higher accuracy while maintaining the advantage of stability at large time steps. First note that any scheme of the form:

$$\begin{aligned} u^{n+1} &= F^{(i)} \\ &= \begin{cases} \dfrac{u^n}{1 - (\Delta t/\varepsilon) F_<(u^n,\, u_{\rm th})} & \text{if } u^n \le u_{\rm th} \\ \dfrac{u^n + (\Delta t/\varepsilon) F_>(u^n,\, u_{\rm th})}{1 + (\Delta t/\varepsilon) F_>(u^n, u_{\rm th})} & \text{if } u^n > u_{\rm th} \end{cases} \end{aligned} \tag{8}$$

will have (7) as a limit. By requiring (8) to be order $O(\Delta t^3)$, considering v and hence $u_{\rm th}$ fixed, it is possible to find a simple choice for the functions $F_<$ and $F_>$:

$$\begin{aligned} F_<(u^n,\, u_{\rm th}) &= (1 - u^n)(u^n - u_{\rm th}) \\ &\quad \times \left(1 + \frac{\Delta t}{2\varepsilon}(u_{\rm th} - (u^n)^2)\right) \end{aligned} \tag{9}$$

$$\begin{aligned} F_>(u^n,\, u_{\rm th}) &= u^n(u^n - u_{\rm th}) \\ &\quad \times \left(1 + \frac{\Delta t}{2\varepsilon}(2u^n - u_{\rm th} - (u^n)^2)\right). \end{aligned} \tag{10}$$

It can be checked that the resulting scheme is $O(\Delta t^3)$ by substituting into (8) and expanding to $O(\Delta t^3)$. We have written the functions in such a

way that it is clear that for $\Delta t/\varepsilon$ small these are small modifications of the original implicit scheme. Figure 2 shows how this scheme compares with the other methods, not just for small $\Delta t/\varepsilon$ but over a range of time-steps. The additional computational expense of this higher-order method is a few additional multiplications, and when taking into account that F need only be evaluated for $u^n > \delta$, this is well worth the gain in precision.

In our time-stepping scheme for u we assume that v is fixed and this requires some explanation. Because the slow v-equation is time-stepped via the explicit-Euler method (4), (and because we also step the diffusion terms via the explicit-Euler method), the time-stepping scheme for the full reaction-diffusion equations can have a one-step error of at best $O(\Delta t^2)$, independently of how accurately the fast u-equation is treated. However, the change in v over a time-step is very small, and in the full reaction-diffusion system the largest contribution to Δu comes from the reaction terms. Hence by focusing on the fast dynamics and stepping the u reaction terms with a more accurate method we can obtain a significant reduction in the time-stepping error. Even if there is no formal improvement in the *order of accuracy* of the full scheme, the *size* of the dominant errors is reduced. This is accomplished with a trivial programming change and adds little time to the computations. In Sec. 3.1 we consider how the improved implicit scheme compares with the other schemes for full reaction-diffusion simulations.

2.2. *Including diffusion*

With the approximation that $u = 0$ for $u < \delta$, the u-field is constant (zero) over substantial regions of the computational domain, e.g. Fig. 1. Hence the Laplacian of the u-field is also zero in these regions and need not be evaluated. To avoid unnecessary computation, the Laplacian of u can be evaluated by scattering values rather than by gathering values. Consider the seven-point finite-difference formula for the Laplacian on a regular cubic lattice in 3D:

$$h^2\nabla^2 u_{ijk} \simeq u_{i+1,jk} + u_{i-1,jk} + u_{i,j+1,k} + u_{i,j-1,k} + u_{ij,k+1} + u_{ij,k-1} - 6u_{ijk}\,,$$

where u_{ijk} is the value of u at grid point $(i,\, j,\, k)$ and h is the grid spacing. Letting Σ denote the weighted sum of points given by the above formula, then a scatter evaluation of this sum is obtained by looping over the grid indices and scattering values of u to neighboring Σ:

for each $i,\, j,\, k$

$\Sigma_{ijk} \leftarrow \Sigma_{ijk} - 6u_{ijk}$

$\Sigma_{i+1,j,k} \leftarrow \Sigma_{i+1,j,k} + u_{ijk}$

$\Sigma_{i-1,j,k} \leftarrow \Sigma_{i-1,j,k} + u_{ijk}$

$\Sigma_{i,j+1,k} \leftarrow \Sigma_{i,j+1,k} + u_{ijk}$

$\Sigma_{i,j-1,k} \leftarrow \Sigma_{i,j-1,k} + u_{ijk}$

$\Sigma_{i,j,k,+1} \leftarrow \Sigma_{i,j,k+1} + u_{ijk}$

$\Sigma_{i,j,k,-1} \leftarrow \Sigma_{i,j,k-1} + u_{ijk}$

where all Σ_{ijk} are initially zero. The factor of h^2 is absorbed into a coefficient to be multiplied by Σ. The result is exactly the same as if one had gathered values according to:

for each $i,\, j,\, k$

$$\Sigma_{ijk} \leftarrow u_{i+1,jk} + u_{i-1,jk} + u_{i,j+1,k} + u_{i,j-1,k} + u_{ij,k+1} + u_{ij,k-1} - 6u_{ijk}\,.$$

What makes a scatter evaluation of the sum desirable is that it can be combined into the algorithm for the reaction dynamics in such a way that unnecessary computation is avoided at points that make zero contribution to the Laplacian of the u-field. Specifically, the following algorithm updates a single grid point in the spatial domain and computes its contribution to the sum of neighboring points for use at the next time step:

if $(u_{ijk} < \delta)$

$\{u_{ijk} \leftarrow r\Sigma_{sijk}$

$v_{ijk} \leftarrow v_{ijk} + \Delta t\, g(0,\, v_{ijk})\}$

else

$\{u_{\rm th} \leftarrow u_{\rm th}(v_{ijk})$

$v_{ijk} \leftarrow v_{ijk} + \Delta t\, g(u_{ijk},\, v_{ijk})$

$u_{ijk} \leftarrow F(u_{ijk},\, u_{\rm th}) + r\Sigma_{sijk}$

$\Sigma_{s'ijk} \leftarrow \Sigma_{s'ijk} - 6u_{ijk}$ (11)

$\Sigma_{s',i+1,j,k} \leftarrow \Sigma_{s',i+1,j,k} + u_{ijk}$

$\Sigma_{s',i-1,j,k} \leftarrow \Sigma_{s',i-1,j,k} + u_{ijk}$

$\Sigma_{s',i,j+1,k} \leftarrow \Sigma_{s',i,j+1,k} + u_{ijk}$

$\Sigma_{s',i,j-1,k} \leftarrow \Sigma_{s',i,j-1,k} + u_{ijk}$

$\Sigma_{s',i,j,k+1} \leftarrow \Sigma_{s',i,j,k+1} + u_{ijk}$

$\Sigma_{s',i,j,k-1} \leftarrow \Sigma_{s',i,j,k-1} + u_{ijk}\}$

$\Sigma_{sijk} \leftarrow 0$

where $r \equiv \Delta t/h^2$. The sum $\Sigma_{s,i,j,k}$ has four subscripts, the first of which takes on just two values

(zero and one, say). The values of s and s' are interchanged at every time step. In effect, there are two Laplacian fields which are used in alternation: The first (unprimed) is used to update the u-field at the current step and is then set to zero for use at the next time step. The second (primed) is computed for use at the next time step.

It can be seen how the approximation $u = 0$ whenever $u < \delta$ leads to a reduction in the amount of computation: Whenever $u = 0$ the reaction terms for u do not need evaluation and no work is necessary for updating the spatial sum Σ for the next time step.

There are two additional points to consider with regard to diffusion. The first is that we have only treated the fast u-field. If the slow variable v is diffusing as well, i.e. $D \neq 0$, then one must compute the Laplacian of this field. In this case there are no tricks to apply and one must compute the weighted sum Σ at all points in the domain. Fortunately, we will be primarily concerned with the case where $D = 0$.

The second point is more general and concerns the finite-difference approximation to the Laplacian operator. We write the approximation in the form:

$$\nabla^2 u_{ijk} = \frac{1}{wh^2}\Sigma_{ijk} + E\,,$$

where Σ is a sum with integer weights, w is a constant, and E is the truncation error of the approximation. Then the 7-point finite difference formula considered thus far is given by

$$\Sigma_{ijk} = u_{i+1,jk} + u_{i-1,jk} + \cdots - 6u_{ijk}$$

with $w = 1$; see Fig 3(a). Only the center and nearest neighbor points are used in this approximation. Assuming one is approximating the Laplacian of a differentiable function u at a location corresponding to grid point (i, j, k), then the error incurred in using the 7-point approximation is given by:

$$E = \frac{h^2}{12}\left(\frac{\partial^4}{\partial x^4} + \frac{\partial^4}{\partial y^4} + \frac{\partial^4}{\partial z^4}\right) u|_{ijk} + O(h^4)\,,$$

and the stability constraint for explicit-Euler time stepping leads to a maximum time step of:

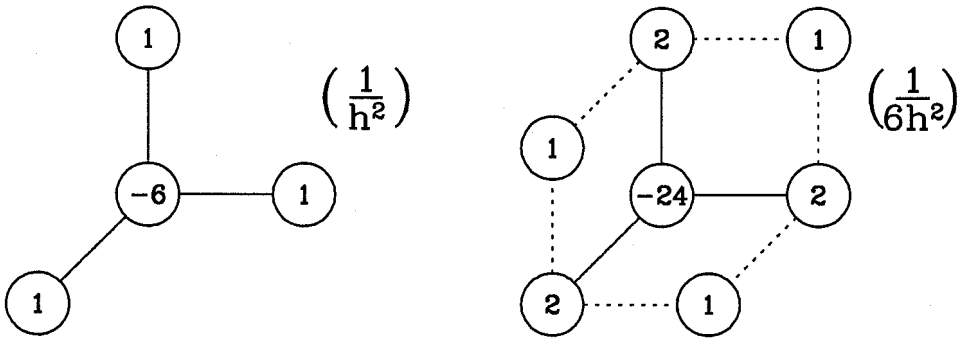
$$\Delta t_{\max} = \frac{h^2}{6}\,.$$

Other approximations can be used for the Laplacian operator, however. In particular consider the approximation given by the 19-point sum:

$$\begin{aligned}\Sigma_{ijk} = {} & 2u_{i+1,jk} + \cdots + 2u_{ij,k-1} + u_{i+1,j+1,k} \\ & + u_{i-1,j+1,k} + \cdots + u_{ij-1,k-1} - 24u_{ijk}\end{aligned}$$

with $w = 6$, [See Fig 3(b)]. The truncation error is given by:

$$E = \frac{h^2}{12}\nabla^4 u|_{ijk} + O(h^4)\,,$$



(a) (b)

Fig. 3. Stencils showing the integer weights for (a) the 7-point and (b) the 19-point finite-difference representation of the Laplacian operator in 3D. For clarity only one quadrant is shown in each case. For the 7-point case there are six nearest-neighbors having weight 1. The factor multiplying all weights is $1/h^2$. For the 19-point case there are six nearest neighbors with weight 2 and 12 next-nearest neighbors with weight 1. The overall factor is $1/6h^2$.

and the maximum stable time step for explicit-Euler time stepping is given by (see Appendix):

$$\Delta t_{\max} = \frac{3h^2}{8}\,.$$

The 19-point formula is superior to the more common 7-point formula in the following ways. Most importantly, simulations with the 19-point approximation are some 80% faster than those using the 7-point approximation. The reason is that the limiting time step ($\Delta t_{\max}$) for the 19-point approximation is 2.25 times larger than that for the 7-point approximation. Of course the 19-point approximation requires more computation, but our tests show that with only the u-field diffusing, the 19-point approximation takes 1.3 times the CPU time, per time step, of the 7-point approximation (this will depend to some extent on computer architecture). The larger time step possible with the 19-point method more than makes up for the additional computation required and thus leads to overall faster simulations with no additional memory requirements.

A second advantage of the 19-point formula is that the leading contribution to the error in this case is invariant under rotations. This means that grid anisotropies do not show up at leading order when approximating $\nabla^2 u$ for smooth functions u. Such anisotropies do appear at leading order when using the 7-point approximation.

A final small advantage of using the 19-point formula is that the stability limit $\Delta t_{\max}$ is identical to that for the 9-point formula for approximating the Laplacian in 2D (see Appendix). It is convenient when considering simulations in both 2D and 3D to be able to use the same time-step Δt for a given grid spacing h independent of the space dimension considered.

The 19-point formula can be implemented in just the same way as the 7-point method in algorithm (11). The only significant shortcoming we see in the 19-point formula is the difficulty in coding the boundary conditions when using this more complicated stencil.

2.3. *Visualizations and filament detection*

We combine our fast simulation technique with real-time visualizations so as to view and interact with simulations as they progress. It is highly desirable to have such interactive capabilities and our method is sufficiently fast, at intermediate and coarse resolutions, that this is feasible while running on a desktop workstation. We have employed a state-of-the-art method known as *marching cubes* for rendering iso-surfaces in 3D. In our case these are surfaces of constant u or v. The marching-cubes method was first introduced by Lorensen and Cline [Lorensen & Cline, 1987; Cline *et al.*, 1988]. Problems were noted soon after it was introduced and these were solved in a clever way by Montani *et al.* [1994]. We use the Montani *et al.* extension of the marching cubes algorithm.

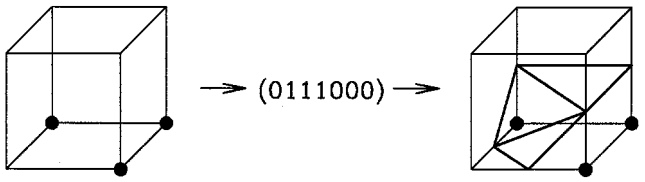


Fig. 4. Illustration of the marching-cubes method. Left: Cube formed from neighboring points of the computational mesh. Solid circles indicate cube vertices which have field values above the contour level (e.g. $u > 0.5$). An eight-bit index is formed based on which vertices are above or below the contour value. An index other than (00000000) or (11111111) indicates that the iso-surface passes through the cube and a lookup table determines a unique triangularization (right) representing the portion of the iso-surface lying within the cube.

The essence of the method is very simple: One considers, in turn, each elementary cube in the computational mesh and determines which vertices of the cube have values above and below the prescribed contour value. From this one obtains an 8-bit index (0–255 decimal). If the index is 0 or 255 then the iso-surface does not cut the cube and one continues (marches) to the next cube. If the iso-surface cuts the cube, then the index is used in a lookup table to determine a particular triangularization of the iso-surface in the cube (see Fig. 4). The intersections of the triangles and the cube edges are determined by linear interpolation from values of the variable at the cube vertices. The triangularization is such that the triangles from neighboring cubes combine to give a (locally) connected surface.

Marching cubes is an ideal scheme to use in conjunction with our simulations because of its simplicity and speed, and because it provides good spatial resolution while using only very local data within the volume. The method is fast because a decision is made quickly as to whether or not the contour intersects an elementary cube. It provides good surface detail because there are frequently several

triangles per elementary cube, particularly in regions of high surface curvature. The method was in fact designed for medical imagining where surface detail is of prime concern.

We have extended the marching cubes method so as to determine scroll filaments with minimal computational expense over what is necessary for rendering a single iso-surface. By filament we shall mean the curve of intersection between appropriate u and v iso-surfaces. The reason that it is computationally advantageous to combine the filament detection with surface rendering is that once the u-iso-surface, say, has been found then one has greatly reduced the set of cubes on which to test for intersections with the v-iso-surface. Given a cube which is intersected by both the u and v iso-surfaces, we find all intersections of triangles from the two triangularizations. This gives a set of line segments lying on (to within our numerical approximation) the filament. As with the surface rendering, this method gives good detail of the filament, particularly in regions of high filament curvature.

In 3D there is a question of how best to assay wave dynamics and in particular how to define the filament of scroll waves. We do not follow the common method of defining the filament as the region of space which does not become excited over a certain time interval (on the order of the rotation period of the wave), e.g. [Henze, 1993; Henze & Tyson, 1996]. Instead, we simply use crossed contours as the instantaneous location of the scroll ring. These intersections can be computed very precisely and do not depend on any averaging. Moreover, we plan to approach the dynamics of 3D excitable media from a dynamical-systems view point similar to that used successfully in the study of spiral waves [Barkley & Kevrekidis, 1994; Barkley, 1995; Mantel & Barkley, 1996], and for this it is desirable to have a simple, well-defined projection from the full phase space of the reaction diffusion equations onto some low-dimensional phase space.

3. Results

3.1. *Two space dimensions*

We first consider the dynamics of spiral waves in 2D. This case allows us to assess the quality of simulations over a wide range of resolutions with greater ease than is possible in the truly three-dimensional case. We have simulated spiral waves for a variety of grid spacings h and time-steps Δt with model parameters fixed at values such that the spiral executes rigid (periodic) rotations. The parameters are: $a = 0.8$, $b = 0.01$, $\varepsilon = 0.02$, and $D = 0$. We consider this fixed set of kinetics parameters throughout the paper.

Figure 5 shows a qualitative comparison of spirals at three spatio-temporal resolutions. The fine resolution case has been chosen such that the rotation period and spiral wavelength are within 1% of the "exact" values obtained at still higher resolutions. The medium resolution simulation is about 10 times faster and provides a period and

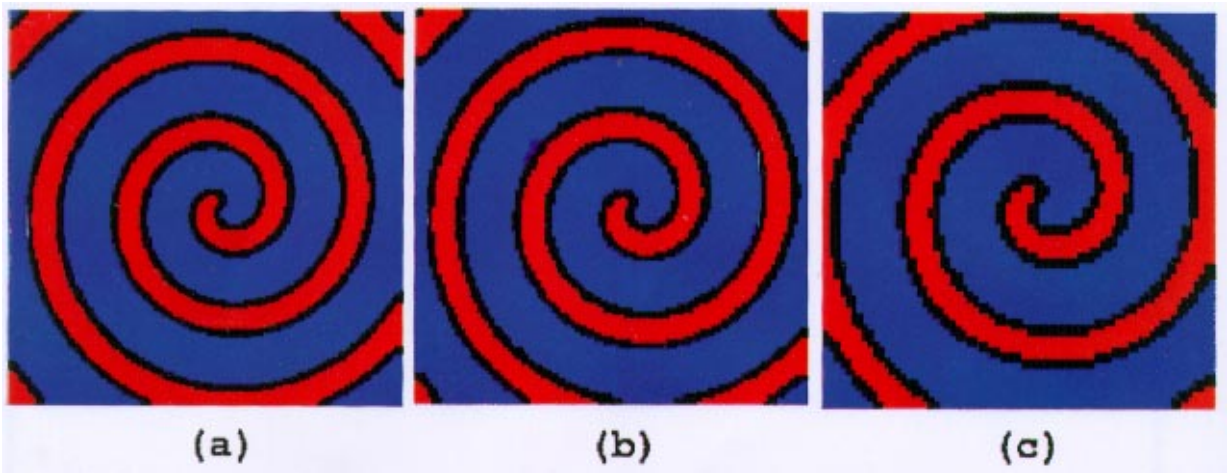


Fig. 5. Spiral waves at three spatio-temporal resolutions. Model parameters are as given in the text. (a) $h = 2/7$, $\Delta t/\varepsilon = 0.6$, $\delta = 10^{-4}$ (b) $h = 3/7$, $\Delta t/\varepsilon = 2.4$, $\delta = 2 \times 10^{-3}$ (c) $h = 5/7$, $\Delta t/\varepsilon = 7.2$, $\delta = 5 \times 10^{-2}$. The domain is of size $L = 50$ in each case. Simulation (b) is about 10 times faster than (a), and the coarse, CA-type, simulation (c) is about 100 times faster than (a). Plotted is the u-field with $u < 0.1$ (blue), $0.1 < u < 0.9$ (black), and $u > 0.9$ (red).

Table 1. Spiral rotation period for various grid spacings and time steps. Results in square brackets are for explicit-Euler time-stepping and results in parentheses are for second-order implicit time-stepping. All other results are for third-order implicit time-stepping.

	$h = 1/7$	$h = 2/7$	$h = 3/7$	$h = 4/7$	$h = 5/7$
$\Delta t/\varepsilon = 7.2$					3.8641
$\Delta t/\varepsilon = 4.8$				3.5771	3.7523
$\Delta t/\varepsilon = 2.4$			3.2896	3.3991	3.6163
$\Delta t/\varepsilon = 1.2$		3.1403 [3.1624] (3.5156)	3.1908	3.3257	3.5488
$\Delta t/\varepsilon = 0.6$		3.1055 [3.1372] (3.3269)	3.1622	3.3073	3.5425
$\Delta t/\varepsilon = 0.3$	3.0774	3.0999 [3.1229] (3.2212)	3.1593	3.3077	
$\Delta t/\varepsilon = 0.1$		3.1116 [3.1127] (3.1464)			

wavelength which are within 7% of the exact values. The coarse simulation is 100 times faster than the fine case and gives a period and wavelength which differ by roughly 25% from the true values. The values of h correspond to those in Fig. 1 and the values of $\Delta t/\varepsilon$ correspond to those in Fig. 2.

For a quantitative comparison we have computed the rotation period for the spiral as a function of grid spacing and time step. The results are shown in Table 1. Similar trends can be found in wavelength data, but it is possible to obtain much higher precision in period measurements by averaging over many periods, so we focus on these data.

There are two points to make based on the data shown and our general impressions from numerous simulations. The first can be seen from the runs at $h = 2/7$ in Table 1. The spiral period converges much faster as a function of time step with third-order implicit time-stepping than for either of the other two methods. The second-order implicit method is particularly poor. This is what one would expect based on the plots shown in Fig. 2. The simulation data show that the improved kinetics scheme does indeed improve the quality of full reaction-diffusion simulations. Thus we conclude that in almost all circumstances it is better to use the new third-order method.

The second point concerns the usefulness of coarse, CA-type, simulations such as shown in Figs. 1(c) and 5(c). While one obtains spiral waves which look quite reasonable in many respects, e.g. the form of the spiral wave in Fig. 5(c), our feeling is that these simulations are too far from the correct PDE results to be of use for serious study. On the other hand, we feel that the our approach is very advantageous at intermediate resolutions such as shown in Figs. 1(b) and 5(b). Here one obtains quite good results with simulations that are significantly faster than those necessary for full resolution. We shall see this more clearly in the 3D case.

3.2. *Three space dimensions*

Our detailed studies in 3D have been for scroll rings such as those illustrated in Fig. 6, though we have also explored more complicated structures. The scroll rings we simulate are axisymmetric and while we do not impose full axisymmetry on solutions, to reduce computational expense we impose a 4-fold rotational symmetry by simulating only 1/4 of the ring as shown. Because the scrolls are axisymmetric, their dynamics are limited to changes in radius, R, and position, Z, along the axis of symmetry, i.e. a scroll ring may shrink or expand in radius, and it may propagate (drift) along the symmetry axis, but that is all. We use periodic boundary conditions in the direction of drift so that rings may drift freely. Neumann boundary conditions are used in the other two directions. Except where stated otherwise, all results reported in this section are for the new implicit time stepping scheme and the 19-point Laplacian formula.

Taking the same approach as in the 2D case, we assess the reliability and speed of 3D simulations at different spatio-temporal resolutions by making quantitative measurements of the scroll dynamics for a set of grid spacings h and time-steps Δt. For each case we follow the dynamics of a fixed initial scroll ring and extract the ring radius and center as a function of time. Specifically, at each time step we find the best fit circle to the points defining the filament. For all but the coarsest resolutions the

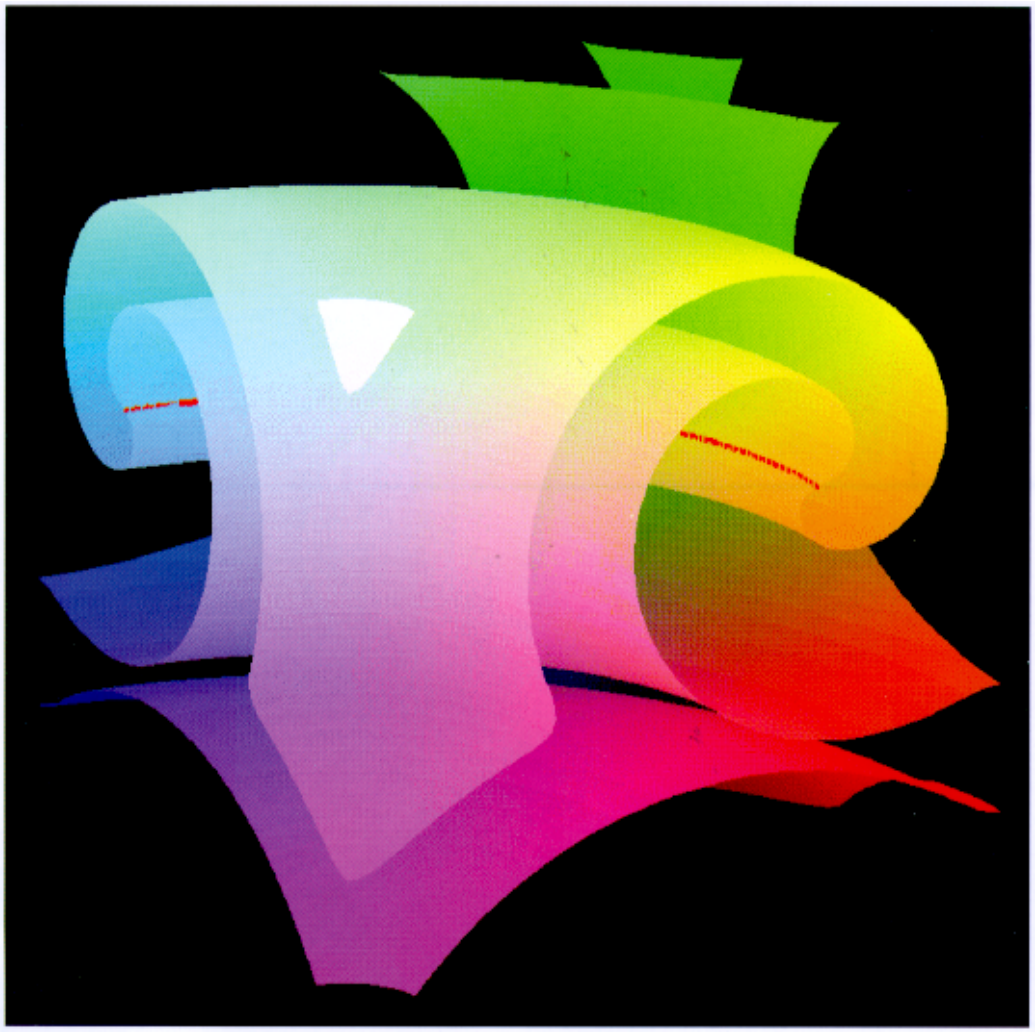


Fig. 6. Scroll ring from 3D simulations. One quarter of the ring is simulated. The boundary conditions on the cubical volume are periodic top to bottom and no-flux (Neumann) on the other four sides. The $u = 0.5$ iso-surface is shown with colors indicating position in three-space: the three axes correspond to each of red, green, and blue, with intensity varying from zero to full along each coordinate, e.g. green intensity varies from zero to full in the vertical direction. The bold red curve shows the location of the instantaneous filament. The kinetics parameters are given in Sec. 3.1; the volume has length $L = 24$ on a side; the numerical parameters for the simulation are: $h = 3/7$, $\Delta t/\varepsilon = 2.4$, $\delta = 10^{-3}$.

filament data lie very close to this circle. We then define R to be the radius of this circle and Z to be the location of the circle center along the symmetry axis.

Figure 7 shows results for scroll dynamics at three representative spatio-temporal resolutions — the same three that were considered in the 2D case in Fig. 5 (though here $\delta = 10^{-3}$ for all cases). One can see the approximately periodic oscillations of the filament, due to scroll rotation, superimposed upon a slower shrink and drift of the scroll ring. We have defined the filament (Sec. 2.3) such that these oscillations appear in the filament dynamics. The medium-resolution case is roughly 12 times faster than the fine-resolution case, and the coarse, CA-type, simulation is more than 200 times faster than the fine simulation. (These timings are for the simulations only; they do not include contributions due to graphics or filament finding.)

The slow dynamics seen in Fig. 7 follows from the theory for the dynamics of untwisted scroll rings, e.g. [Panfilov *et al.*, 1986]. In this theory, the slow dynamics of axisymmetric ring filaments whose curvatures are not too great is given by:

$$\dot{R} = -c_s/R \tag{12}$$

$$\dot{Z} = c_d/R \tag{13}$$

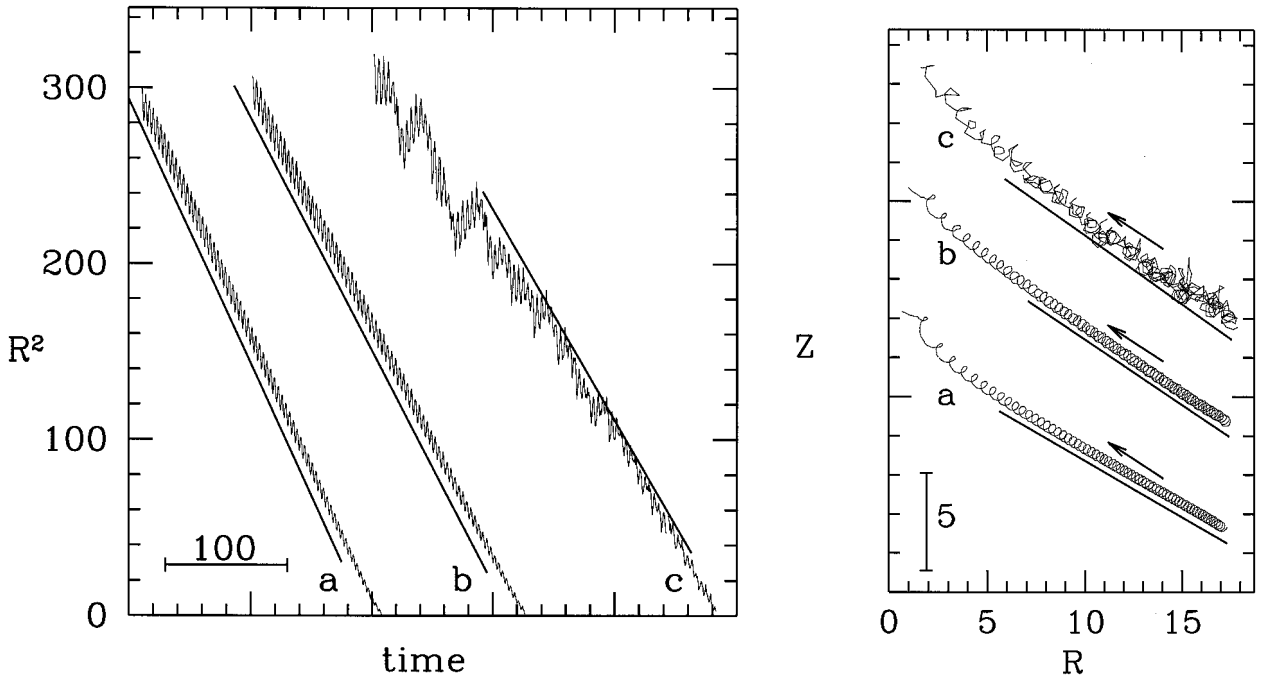


Fig. 7. Collapse dynamics of axisymmetric scroll rings. Plotted is R^2 versus time (left) and Z versus R (right) at three numerical resolutions (a) $h = 2/7$, $dt/\varepsilon = 0.6$, (b) $h = 3/7$, $dt/\varepsilon = 2.4$, and (c) $h = 5/7$, $dt/\varepsilon = 7.2$. Model parameters are as elsewhere. Arrows on the right show the evolution direction. For R greater than about 5 the collapse is "linear". The coarse-resolution case shows only qualitatively the expected shrink and drift. The medium-resolution case gives a good approximation to fine-resolution simulations while requiring less than 10% of the CPU time.

where the constants c_s and c_d are called the shrink and drift coefficients respectively. These equations can be written in the more useful form:

$$\frac{dR^2}{dt} = -2c_s \tag{14}$$

$$\frac{dZ}{dR} = -\frac{c_d}{c_s}\,. \tag{15}$$

Figure 7 shows that the slow dynamics are qualitatively in accord with theory. That is, so long as R is not too small, the slow dynamics are such that R^2 decreases linearly with time and the path in (R-Z) coordinates is linear. The coarse, CA-type, simulation shows considerable "noise", but even in this case the dynamics are qualitatively in accord with Eqs. (14) and (15).

For a more quantitative comparison, we use relations (14) and (15) to extract shrink and drift coefficients from our simulations. Tables 2 and 3 summarize our results. The error bounds for the coefficients in Tables 2 and 3 come from uncertainty inherent in estimating slopes from finite, and in the case of coarse simulation, noisy data sets. We include results from a single simulation using explicit Euler method to show that, as in the 2D case, it does not perform as well as the new implicit scheme.

The most striking aspect of the shrink and drift data is that the coefficients are much more sensitive to spatio-temporal resolution than is the rotation period in 2D simulations (Table 1). This is especially true for the drift coefficient. Even at a resolution where the spiral period is converged to within about 1% of the "exact" value (i.e. $h = 2/7$, $\Delta t/\varepsilon = 0.6$), the drift coefficient is off by about 10%, which is far more than the uncertainty in its measurement. The explicit-Euler result is off by a considerably larger amount. (We did not run with a timestep smaller than $\Delta t/\varepsilon = 0.3$; nevertheless, we use these values as a basis to estimate the precision of coefficients.) The shrink coefficient is somewhat better being within about 3% of the converged value. Moreover, the medium-resolution simulation in Fig. 7, which would appear to be reasonably good, gives a drift coefficient which is off by as much as 28% from the converged result. We will return to this point in the discussion, but for now we note there is not, as far as we are aware, a fundamental difference in the resolution

Table 2. Scroll shrink coefficient c_s for various grid spacings and time steps. The coefficient in square brackets is for explicit-Euler method; all other results are for third-order implicit time-stepping. Except for $h = 5/7$, the uncertainty in the coefficients due to the fitting procedure is about $\pm 2\%$. For $h = 5/7$ the uncertainty is about $\pm 5\%$.

	$h = 1/7$	$h = 2/7$	$h = 3/7$	$h = 4/7$	$h = 5/7$
$\Delta t/\varepsilon = 7.2$					0.604
$\Delta t/\varepsilon = 4.8$				0.593	
$\Delta t/\varepsilon = 2.4$			0.674		0.544
$\Delta t/\varepsilon = 1.2$		0.711			
$\Delta t/\varepsilon = 0.6$		0.757 [0.749]	0.752	0.665	0.446
$\Delta t/\varepsilon = 0.3$	0.793	0.779			

Table 3. Scroll drift coefficient c_d for various grid spacings and time steps. The coefficient in square brackets is for explicit-Euler method; all other results are for third-order implicit time-stepping. For $h \leq 3/7$, the uncertainty in the coefficients due to the fitting procedure is about $\pm 5\%$. For the others the uncertainty is $\pm 15\%$ or less, except for $h = 5/7$, $\Delta t/\varepsilon = 0.6$, where the uncertainty is about $\pm 25\%$.

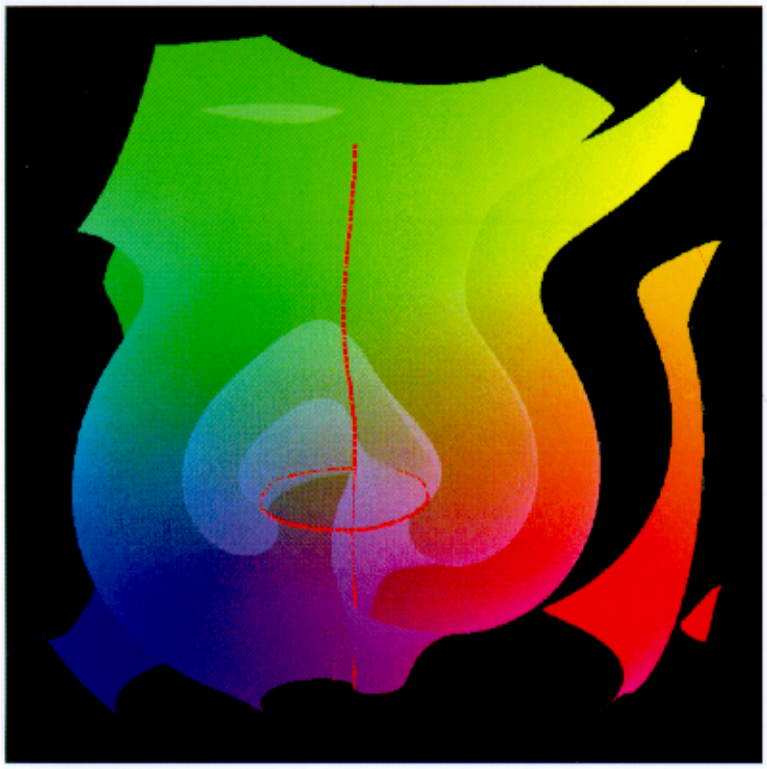
	$h = 1/7$	$h = 2/7$	$h = 3/7$	$h = 4/7$	$h = 5/7$
$\Delta t/\varepsilon = 7.2$					0.859
$\Delta t/\varepsilon = 4.8$				0.753	
$\Delta t/\varepsilon = 2.4$			0.973		0.771
$\Delta t/\varepsilon = 1.2$		1.15			
$\Delta t/\varepsilon = 0.6$		1.36 [1.140]	1.44	1.40	0.493
$\Delta t/\varepsilon = 0.3$	1.390	1.48			

requirements of 2D and 3D simulations of the type we are considering here, rather it is a question of what is being measured. For example, one can extract the local period from the measurements in the Fig. 7 and one finds a dependence on resolution very close to what was found in the 2D case.

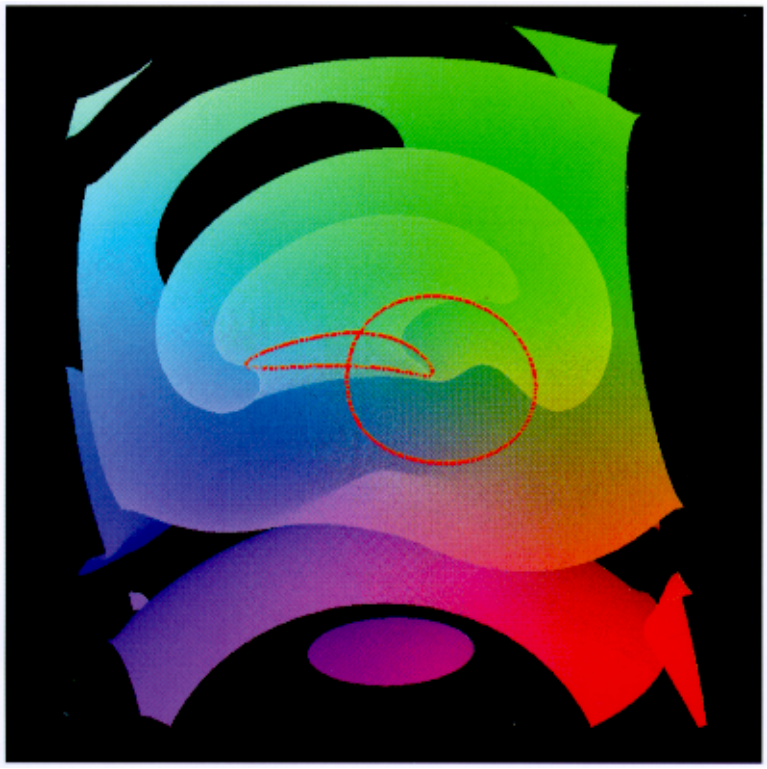
We further note that it is not the drift in Z that is so sensitive to resolution, rather it is the drift coefficient. By this we mean that $Z(t)$ is quite insensitive in resolution. In fact the plots of $Z(t)$ for different resolutions lie quite close to one another for a considerable time. This is less true for plots of $R(t)$ and it is in relating $\dot{Z}$ to R though the drift coefficient that the sensitivity to resolution appears.

There are two further points which we mention here and return to in the discussion. The first is that, except for the largest grid spacing considered, the drift coefficient is much more sensitive to time step than to grid spacing and it appears that for an accurate measure of drift coefficient the best approach is to simulate with a small time step. The related though somewhat contradictory point is that for larger grid spacings, $h \geq 4/7$, the quality of filament paths is better at larger rather than smaller time-steps. We do not show any filament data for small time-steps, but one can see, for example, the very poor shrink and drift coefficients obtained at $h = 5/7$, $\Delta t/\varepsilon = 0.6$. We have no explanation for this behavior.

We end this section by showing in Fig. 8 results from simulations of two other 3D structures: A twisted scroll ring and a pair of linked rings. The numerical and model parameters are the same as for the medium resolution simulations considered previously, e.g. Fig. 7(b). The initial conditions for these are obtained using the complex polynomial method described by Winfree [1987, 1995]. Our method is sufficiently fast that one may simulate these structures interactively, in real time on a moderately powerful workstation.



(a)



(b)

Fig. 8. (a) Twisted scroll ring and (b) pair of linked rings. In each case a clipping plane cuts surfaces halfway through the volume; the filaments are not clipped. The color scheme and parameter values are the same as in Fig. 6; only the initial conditions are different and here the volume has length $L = 26$ on a side.

Importantly, these simulations are qualitatively reliable. Hence simulations at these resolutions can be used to explore parameter space or examine a variety of initial conditions at these fixed model parameters.

4. Summary and Discussion

We have presented a computationally efficient method for simulating waves in 3D excitable media. Our principal contributions are (1) a more accurate method for implicitly time stepping the reaction terms in the equations and (2) a superior finite-difference representation for the Laplacian operator in 3D. This second contribution is new in that, to our knowledge, no one currently simulating 3D excitable media is using this representation, and its advantages are not known or appreciated within this field.

We have focused our considerations on three spatio-temporal resolutions (fine, medium, and coarse) with a fixed set of model parameters. In 3D the relative speeds (reciprocal of execution times) of these are $fine : medium : coarse = 1 : 12 : 240$. We have shown that simulations at medium resolution are qualitatively quite good and in some measures these simulations do not differ significantly from fully resolved ones. Using our approach the medium-resolution case offers more than a factor of 10 increase in speed when compared to fine simulations. (There is approximately another factor of 2 increase in speed when we compare with standard methods of simulating the FitzHugh–Nagumo equations.) Presumably one could further reduce the resolution and obtain speed increases of factors of 20 or more while maintaining useful results. We have, however, found that as one gets near the coarsest simulations possible in our approach, results become too inaccurate to be useful in most circumstances, and in particular we feel that a speedup by a factor of 100 is not realistic. Thus we limit our claim to having a method which allows approximately a factor of 10 increase in speed in simulations that can be reliably used to study waves in 3D excitable media.

We have noted that in the limit of large time steps our method becomes similar to the CA method considered by Henze and Tyson [1996]. It is thus natural to compare approaches and results. The important differences between the CA approach and ours are (1) our excitation variable u takes on a continuous range of values, so that even when simulating in the coarse spatio-temporal limit, there are points on the interface and we continue to use standard representations for the Laplacian operator. (2) Our grid spacings and time steps can be adjusted independently (within stability constraints). (3) Our spatio-temporal resolution can be varied continuously from fine to coarse. Many of the advantages that CAs originally enjoyed over PDE simulations, such as smaller memory requirements and faster integer arithmetic, have been lost in the improvements made to correct the shortcomings of CAs due to their discrete nature. We refer the reader to Henze and Tyson for a discussion of these points.

It is difficult to compare results from the two approaches and to answer the basic question: "Which method is faster?" The reason is that one must first have a precise measure of the quality of a simulation and this is probably impossible to define. A comparison of timings between the specific coarse simulations that we have considered and timings of the Henze–Tyson CA would make no sense without knowing that they are of comparable quality. Which of our continuous range of resolutions is to be compared with Henze–Tyson? There are other reasons comparison is not simple, for example, the result will depend not just on the algorithms but also on the actual implementations which may or may not be optimal for a given numerical approach.

One may take the point of view that our fast simulations are simply under-resolved simulations. There is certainly some truth in this. Even so, our numerical method allows one to time-step the reaction terms with larger steps than are stable in other numerical schemes. Without this stability, one could only reduce spatial resolution of the simulations while keeping a fixed time step set by the stability of the reaction terms.

A related point which Henze and Tyson raise and which we have also considered is what time step is optimal for fast, qualitative, simulations. In the CA approach where the space and time steps are linked, Henze and Tyson postulate that the effective large time-step of the CA may in some cases be too large to provide a good simulation. Because we are not restricted in our choice of time-steps, we have examined this and find that the answer depends on which situations one considers. Consider for example Table 2 containing scroll shrink data. In the column $h = 5/7$ one sees that the shrink coefficient becomes particularly bad at small time-

steps. As stated in Sec. 3.2 the poor quality of these simulations is evident in the filament data itself. We find this to be a general trend at least for $h \geq 4/7$. One can also compare the case $h = 4/7$, $\Delta t/\varepsilon = 0.6$ with the case $h = 3/7$, $\Delta t/\varepsilon = 2.4$. The latter simulation is faster than the former by about a factor of 2 yet it gives nearly the same shrink coefficient. This would indicate the one should take the largest time-step possible for a given spatial resolution.

If, on the other hand, one considers the drift coefficient, then except at the largest grid spacing, one sees that it depends primarily on the time-step and very weakly on the grid spacing. In fact, Henze and Tyson found that the drift coefficient was difficult to obtain accurately in their CA simulations and this is what led them to make speculations on the time-step. We offer no explanation for this effect, but we stress the observations reported in Sec. 3.2. Namely, it is not the drifting of the scroll wave in Z that is sensitive to resolution: $Z(t)$ is surprisingly insensitive to resolution. It is $R(t)$ that depends significantly on resolution. It is when one considers the relationship between $\dot{R}$ and R, and between $\dot{Z}$ and R that the sensitivity to time step shows up in the drift coefficient more than in the shrink coefficient.

We conclude with some points about general applicability of our approach and areas for future work.

Our two new contributions to numerical simulations stated above should be broadly applicable. The 19-point representation of the Laplacian operator clearly can be implemented in any finite-difference simulation based on a cubical lattice. We strongly advocate its general use in preference to the 7-point formula. Our approach to the improvement of the reaction terms is generally applicable in the following sense. In effect we have identified the weak link in the numerical simulations, i.e. the time stepping of the fast reaction terms during excitation, and we have strengthened that weak link in a simple way. While our particular improvement depends on the form of the reaction terms, it is plausible that with other models such as the classical FitzHugh–Nagumo equations (for which implicit time stepping is more difficult), employing a higher-order method on just the fast kinetic equation could produce a significant benefit.

This suggests a more general approach to improving numerical simulations of excitable media. Due to the disparate space and time scales associated with excitation it would be advantageous to use adaptive methods which locally evolve the fast regions within the medium using smaller space and time steps. Programming a fully adaptive method for waves in 3D excitable media seems a daunting task. However, one might be able to use a poor-man's approach by extending our method of treating the fast kinetics terms. The next most significant source of error comes from the diffusion of u in the fast regions, i.e. from the approximation of the Laplacian for a rapidly changing function of space. It might be possible to use a method which is higher-order in space in the fast regions only and thereby increase the accuracy where it is important and yet have a scheme which uses a uniform computational mesh. We leave this for the future.

Acknowledgments

This work has been partially supported by grants from the Royal Society and from the EPSRC.

References

Barkley, D. [1991] "A model for fast computer-simulation of waves in excitable media," *Physica* **D49** (1, 2), 61–70.

Barkley, D. [1995] "Spiral meandering," in *Chemical Waves and Patterns*, eds. Kapral, R. & Showalter, K. (Kluwer), pp. 163–190.

Barkley, D. & Kevrekidis, I. [1994] "A dynamical systems approach to spiral-wave dynamics," *Chaos* **4**, 453–460.

Barkley, D., Kness, M. & Tuckerman, L. [1990] "Spiral-wave dynamics in a simple-model of excitable media — The transition from simple to compound rotation," *Phys. Rev.* **A42**(4), 2489–2492.

Beeler, G. & Reuter, H. [1977] "Reconstruction of the action potential of ventricular myocardinal fibres," *J. Physiol.* **268**, 177–210.

Cline, H., Lorensen, W., Ludke, S., Crawford, C. & Teeter, B. [1988] "2 algorithms for the 3-dimensional reconstruction of tomograms," *Med. Phys.* **15**(3), 320–327.

Davidenko, J., Pertsov, A., Salomonsz, R., Baxter, W. & Jalife, J. [1992] "Stationary and drifting spiral waves of excitation in isolated cardiac-muscle," *Nature* **355**(6358), 349–351.

Gerhardt, M., Schuster, H. & Tyson, J. [1990a] "A cellular automaton model of excitable media including curvature and dispersion," *Science* **247**(4950), 1563–1566.

Gerhardt, M., Schuster, H. & Tyson, J. [1990b] "A cellular automaton model of excitable media. 2. Curvature, dispersion, rotating waves and meandering waves," *Physica* **D46**(3), 392–415.

Gerhardt, M., Schuster, H. & Tyson, J. [1990c] "A cellular automaton model of excitable media. 3. Fitting the Belousov–Zhabotinskii reaction," *Physica* **D46**(3), 416–426.

Gerhardt, M., Schuster, H. & Tyson, J. [1991] "A cellular automaton model of excitable media. 4. Untwisted scroll rings," *Physica* **D50**(2), 189–206.

Gray, R., Jalife, J., Panfilov, A., Baxter, W., Cabo, C. & J. D. [1995] "Mechanisms of cardiac fibrillation," *Science* **270**(5239), 1222–1223.

Henze, C. [1993] "Vortex filaments in three dimensional excitable media," PhD thesis, University Arizona.

Henze, C. & Tyson, J. [1996] "Cellular-automaton model of three-dimensional excitable media," *J. Chem. Soc. — Faraday Trans.* **92**(16), 2883–2895.

Lorensen, W. & Cline, H. [1987] "Marching Cubes: A high resolution 3D surface construction algorithm," *Comput. Graph.* **21**(4), 163–169.

Mantel, R. & Barkley, D. [1996] "Periodic forcing of spiral waves in excitable media," *Phys. Rev.* **E54**, 4791–4802.

Markus, M. & Hess, B. [1990] "Isotropic cellular automaton for modeling excitable media," *Nature* **347**(6288), 56–58.

Montani, C., Scateni, R. & Scopigno, R. [1994] "A modified look-up table for implicit disambiguation of Marching Cubes," *Visual Comput.* **10**, 353–355.

Panfilov, A. & Hogeweg, P. [1995] "Mechanisms of cardiac fibrillation," *Science* **270**(5239), 1223 and 1224.

Panfilov, A. & Keener, J. [1995] "Reentry in three-dimensional Fitzhugh–Nagumo medium with rotational Anisotropy," *Physica* **D84**(3, 4), 545–552.

Panfilov, A., Rudenko, A. & Krinsky, V. [1986] "Turbulent rings in three-dimensional active media with diffusion by 2 components," *Biofizika* **31**(5), 850–854.

Weimar, J., Tyson, J. & Watson, L. [1992a] "3rd generation cellular automaton for modeling excitable media," *Physica* **D55**(3, 4), 328–339.

Weimar, J., Tyson, J. & Watson, L. [1992b] "Diffusion and wave-propagation in cellular automaton models of excitable media," *Physica* **D55**(3, 4), 309–327.

Winfree, A. [1987] *When Time Breaks Down* (Princeton Univ. Press, NJ).

Winfree, A. [1994] "Electrical turbulence in three-dimensional heart-muscle," *Science* **266**(5187), 1003–1006.

Winfree, A. [1995] "Persistent tangles of vortex rings in excitable media," *Physica* **D84**(1, 2), 126–147.

Appendix

In this Appendix we sketch a derivation of the time-stepping stability constraint for the 19-point representation of the Laplacian operator in 3D, and we discuss the relationship of this to the 9-point representation in 2D. While the derivation is straightforward, we are unaware of any standard reference which contains the result.

The issue is the numerical stability limit for the linear diffusion equation:

$$\frac{\partial u}{\partial t} = \nabla^2 u$$

when it is time-stepped using explicit-Euler method in time and the 19-point representation of $\nabla^2 u$:

$$u_{ijk}^{n+1} = u_{ijk}^{n} + \Delta t \frac{1}{6h^2} \Sigma_{ijk} \,, \tag{16}$$

where

$$\begin{aligned} \Sigma_{ijk} = {} & 2u_{i+1,jk}^{n} + \cdots + 2u_{ij,k-1}^{n} \\ & + u_{i+1,j+1,k}^{n} + u_{i-1,j+1,k}^{n} + \cdots \\ & + u_{ij-1,k-1}^{n} - 24u_{ijk}^{n} \,. \end{aligned}$$

Writing numerical scheme (16) as:

$$u_{ijk}^{n+1} = \mathbf{A} u_{ijk}^{n} \,, \tag{17}$$

then the scheme is stable so long as all eigenvalues of the operator $\mathbf{A}$ satisfy $|\lambda| < 1$. In practice the stability limit is reached when Δt becomes just large enough that $\mathbf{A}$ first has an eigenvalue $\lambda = -1$.

The eigenfunctions of $\mathbf{A}$ are simply discrete forms of the eigenfunctions of the Laplacian:

$$\psi_{ijk} = \cos(l_x i)\cos(l_y j)\cos(l_z k) \tag{18}$$

(and similarly with sine replacing any of the cosine functions), where l_x, l_y, l_z are wavenumbers. In principle these must be such that ψ_{ijk} satisfies some boundary conditions, but this is not very important. What is important is that the maximum value for each of the ls is π corresponding to the highest spatial frequency supported by the grid.

We substitute (18) into (17) and repeatedly use trigonometric identities which, for example give:

$$\psi_{i+1,jk} + \psi_{i-1,jk} = 2\,\cos(l_x)\psi_{ijk} \,.$$

The eigenvalue equation $\mathbf{A}\psi_{ijk} = \lambda\psi_{ijk}$ is exactly satisfied with:

$$\begin{aligned} \lambda = 1 + \frac{4\Delta t}{6h^2} \{ & \cos(l_x) + \cos(l_y) + \cos(l_z) \\ & + \cos(l_x)\cos(l_y) + \cos(l_x)\cos(l_z) \\ & + \cos(l_y)\cos(l_z) - 6 \} \,. \end{aligned}$$

Minimizing the expression in brackets over allowed values of the wavenumbers gives that -8 is the most negative value possible for the expression in brackets and this occurs with $l_x = \pi$, $l_y = \pi$, $l_z = 0$, or any permutation of l_x, l_y, l_z. Thus the most negative eigenvalue of $\mathbf{A}$ is $\lambda = 1 - 16\Delta t/3h^2$. Setting this to the limiting value, $\lambda = -1$, and solving for Δt gives $\Delta t = (3/8)h^2$. For any smaller Δt all eigenvalues satisfy $|\lambda| < 1$ and so this sets the numerical stability limit.

Note that the most dangerous numerical modes for the 19-point formula are in fact 2D modes, e.g. $\psi_{ijk} = \cos(\pi i)\cos(\pi j)$. This is not the case for the 7-point stencil where the most dangerous numerical modes are $\psi_{ijk} = \cos(\pi i)\cos(\pi j)\cos(\pi k)$.

Finally we note the 19-point Laplacian formula reduces to, in 2D, the the 9-point Laplacian formula considered in [Barkley *et al.*, 1990; Barkley, 1991]. This can be seen by applying the 19-point formula to a field that is independent of z, and hence index k. The result is to sum weights in the 3D stencil over k and this gives the 9-point 2D stencil. The stability limit for the 9-point 2D representation is the same as that just obtained: $\Delta t = (3/8)h^2$. This can be derived using the same method, but it follows simply because the most dangerous modes are the same in each case.