Asynchronous algorithm for integration of reaction–diffusion equations for inhomogeneous excitable media

Guillaume Rousseau and Raymond Kapral[a)]
Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada

(Received 23 March 2000; accepted for publication 18 July 2000)

An asynchronous algorithm for the integration of reaction–diffusion equations for inhomogeneous excitable media is described. Since many physical systems are inhomogeneous where either the local kinetics or the diffusion or conduction properties vary significantly in space, integration schemes must be able to account for wide variations in the temporal and spatial scales of the solutions. The asynchronous algorithm utilizes a fixed spatial grid and automatically adjusts the time step locally to achieve an efficient simulation where the errors in the solution are controlled. The scheme does not depend on the specific form of the local kinetics and is easily applied to systems with complex geometries. © *2000 American Institute of Physics.* [S1054-1500(00)00304-9]

Spatially distributed systems encountered in applications to physical problems are often inhomogeneous, with either the local dynamics or the diffusion or conduction properties varying widely with spatial location. If one considers the propagation of excitable chemical or electrochemical waves through such media one must in addition account for the rapid variations of the fields in the vicinity of the wave fronts which lead to stiff systems of differential equations. Wave propagation in cardiac tissue typifies such behavior. The heart is a very inhomogeneous excitable medium. It consists of a variety of different types of tissue, muscle and fiber with anisotropic conduction properties. Both the excitability and the diffusion depend strongly on the spatial location. The geometry of the medium is also complex with specialized conduction pathways and ''obstacles'' arising from anatomical features. The asynchronous algorithm developed here to simulate partial differential equation models for such media exploits the wide variations in system properties. It automatically adjusts the time step in spatial regions where the dynamics varies sharply and incorporates error control so that the solution satisfies predetermined accuracy criteria. The algorithm utilizes a fixed spatial grid so can be adapted easily to complex geometries and is applicable to any type of local dynamics.

I. INTRODUCTION

The reaction diffusion equation

$$\frac{\partial \boldsymbol{w}(\boldsymbol{r},t)}{\partial t} = \boldsymbol{R} + \nabla.(\boldsymbol{D}(\boldsymbol{r})\nabla \boldsymbol{w})), \tag{1}$$

describes the dynamical behavior of chemical reactions in heterogeneous media; for example, reactions occurring in porous materials, surface catalysis and a variety of biological systems including electrochemical wave propagation in cardiac tissue. In this equation $\boldsymbol{w}(\boldsymbol{r},t)$ is a vector local state variables (chemical concentrations, electrical potential, ion concentrations, etc.) at space point $\boldsymbol{r}$ at time t. The vector function of reaction rates is denoted by $\boldsymbol{R}(\boldsymbol{w}(\boldsymbol{r},t);\boldsymbol{p})$ and depends on a set of parameters $\boldsymbol{p}$. In the most general case, $\boldsymbol{D}(\boldsymbol{r})$ is a tensor whose elements specify the diffusion or conduction properties of the medium for each variable at each spatial point.

Two characteristics of such reaction diffusion equations make their simulation difficult: the local reaction kinetics may have contributions with very different time scales leading to stiff differential equations and sharp front solutions, and the structure of the inhomogeneities may be complex with widely varying diffusion coefficients and local dynamics at different spatial locations. For example, for excitable cardiac tissue, the fast time scale corresponds to the rapid depolarization of the cellular membrane. The use of a small time step which is needed to resolve this behavior makes it more difficult to study the long time behavior of the physical processes of interest. Cardiac tissue is also highly inhomogeneous with specialized high conduction channels with markedly different properties. Various schemes have been devised for the simulation of excitable media because of their importance for applications.[1–6]

Wave propagation in the excitable FitzHugh–Nagumo (FHN) system provides a convenient model to illustrate some of the essential issues. For the FHN system the elements of $\boldsymbol{R}$ are $R_u = (u - u^3/3 - v)/\epsilon$ and $R_v = \epsilon(u + \beta - \alpha v)$. Figure 1 shows a propagating wave in one spatial dimension for $\alpha = 0.5$, $\beta = 0.75$, and $\epsilon = 0.15$ with space-independent diffusion coefficients $D_u = 1$ and $D_v = 0$.

The fast variation of the u field at the front compared to the slow variation of the v field is a characteristic of the stiffness of this system determined by the parameter ϵ. This solution was obtained using an Euler scheme with the lowest order symmetric finite difference form for the Laplacian

a)Electronic mail: rkapral@gatto.chem.utoronto.ca

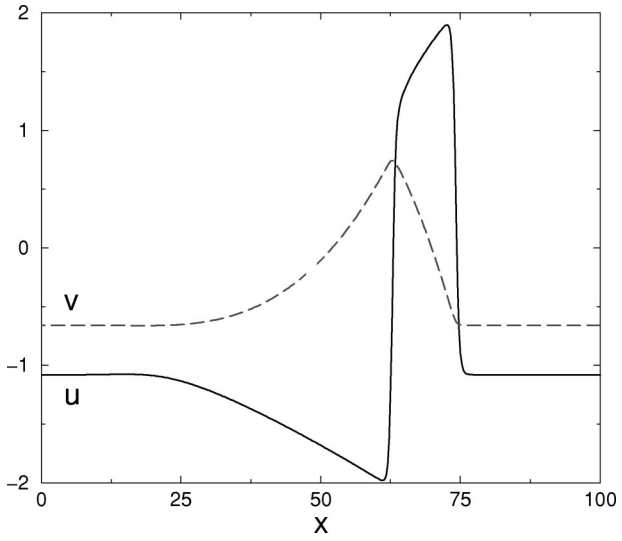

FIG. 1. Propagating wave solution of the FHN model for $\alpha=0.5$, $\beta=0.75$, $\epsilon=0.15$, $D_u=1$, and $D_v=0$ obtained using an Euler scheme and a finite difference approximation for the Laplacian term with $\delta x=0.25$ and $\delta t=0.25\,\delta x^2$, and periodic boundary conditions. Variable u (solid line); v (dashed line).

$$w_k(x,t+\delta t)=w_k(x,t)+\delta t R_k(\boldsymbol{w};\boldsymbol{p}) + \frac{\delta t D_k}{\delta x^2}[w_k(x+\delta x,t) + w_k(x-\delta x,t)-2w_k(x,t)], \quad (2)$$

with periodic boundary conditions. Here k labels the different chemical species or variables. This widely used scheme is easy to implement in any dimension, can accommodate many different boundary conditions, does not depend on a specific form for the reaction term, and can be modified to simulate other spatial derivative terms easily. However, it has poor stability properties and may require very small space and time steps to obtain accurate solutions.

The time step in the explicit Euler scheme cannot be increased indefinitely and is constrained by the requirement that the numerical solution be stable to small perturbations. The temporal evolution of an infinitesimal perturbation is given by the linearization of Eq. (2) which involves both the Jacobian matrix associated with the reaction term, $J_{k\ell}=\partial R_k/\partial w_\ell$, and the diffusion terms. Since the solution of interest may be time dependent, a full nonlinear analysis must be carried out, defining the Lyapunov exponents and associated modes.

In the continuous space limit, $\delta x\to 0$, the weight of the Jacobian terms tends to zero relative to the diffusion terms. If we omit the reaction term and focus on the Laplacian term, the resulting linear equation yields the stability constraint $\delta t<\delta x^2/(2\max_k D_k)$. This approximation is not generally applicable when a finite grid is employed. For example, in the very large grid size limit, $\delta x\to\infty$, the stability condition arises from the diagonalization of the matrix $(I-\delta t J)$ where I is the unit matrix. For the rest state of excitable media, the magnitudes of all eigenvalues λ_l must be smaller than unity and the stability condition is $|\lambda_l|<1, \forall l$. Thus, the possible

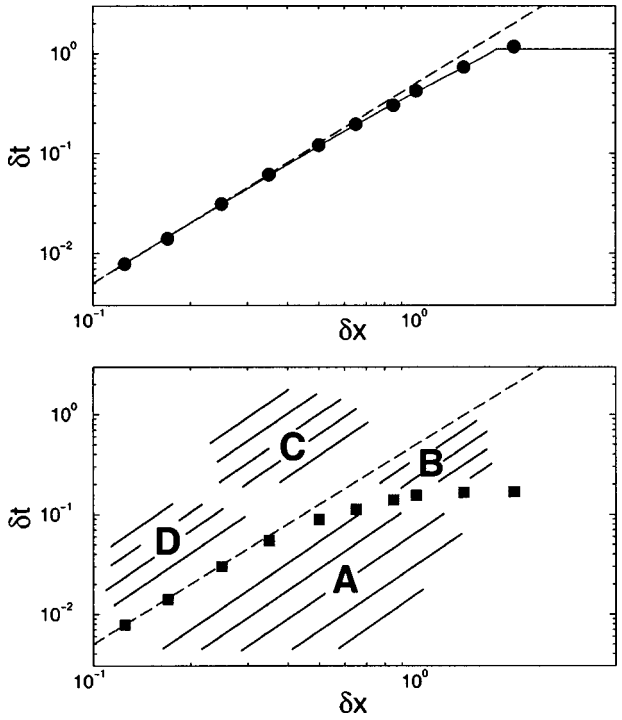

FIG. 2. Upper panel: results of the numerical stability analysis of the uniform solution of the FHN model obtained using the Euler scheme combined with a first order finite difference approximation to the Laplacian (black dots and solid black line). The parameters are the same as in Fig. 1. Lower panel: results of the stability analysis of the pulse solution (solid squares). The dashed line is $\delta t=0.16\,\delta x^2$. The labeled regions in this figure refer to the applicability of schemes based on various combinations of implicit (I) and explicit (E) evaluations of the reaction and diffusion terms, respectively ***A***: (E,E) (Ref. 7), ***B***: (I,E) (Ref. 1), ***C***: (I,I) (Ref. 3), and ***D***: (E,I).

time step values will tend to a constant value for large grid sizes. Figure 2 shows the stability limit of an explicit Euler scheme using first-order finite differences for the FHN model rest state (black dots) and pulse solution (filled squares). Saturation of the allowed time step is seen in both cases but occurs at a somewhat smaller value of δx for the pulse solution. This is typical of excitable media where the fastest dynamics is associated with the transition from the rest state to the excited state. In the continuous space and time limit, the linear stability analysis for the uniform rest state and the pulse solution coincide.

The differences in the stability thresholds in the pure diffusive limit and in intermediate resolution simulations are not negligible. The ''effective'' stability threshold may decrease more slowly than δx^2, leading to the use of a linear variation of the time step with the grid size in some applications (see, for instance, Ref. 2). Such behavior can be expected only far from the continuous space limit and, indeed, suggests the use of a smaller grid size to obtain accurate results. However, simulations performed at coarse or medium resolution provide useful qualitative insight into the different dynamical regimes that may exist. An algorithm that is able to incorporate a wide range of spatial resolutions is of considerable interest since it will allow a rapid, complete study of the dynamics using a single code. This is one of the appealing properties of Barkley's algorithm[1] for excitable media.

Algorithms may be classified according to whether they involve implicit or explicit evaluation of either one or both of the reaction and diffusion terms.[8] Although implicit numerical schemes are more difficult to implement, they are used frequently because of their good stability properties. One may delimit regions of applicability in $(\delta x, \delta t)$ space for each type of algorithm (see Fig. 2). Some algorithms use synchronous updating of all nodes of a regular lattice, while others, including our scheme, use an irregular distribution of time steps and/or grid sizes. Other examples of such schemes include the ''Adaptive Mesh Size Refinement Algorithm'' which uses a variable grid size and time step and has been successfully applied to excitable media[2] and the ''Domain Decomposition and Priority Queue Integration Algorithm'' where an implicit integration method is interlaced with an explicit scheme using an asynchronous control procedure.[6]

The paper is organized as follows. Section II presents a detailed description of an asynchronous time step algorithm defined on a regular lattice. Section III shows why this algorithm is an efficient scheme for the simulation of uniform media in regions ***A*** and ***B*** in Fig. 2, and in region ***D*** for inhomogeneous media where localized high diffusion domains exist. A discussion of our results and other schemes is presented in Sec. IV while the Appendices give some technical details to supplement the material in the text.

II. ASYNCHRONOUS SCHEME FOR INHOMOGENEOUS MEDIA

The asynchronous algorithm is motivated by two features of inhomogeneous excitable media: (1) accurate integration requires different time steps in different spatial regions and (2) the selection of the most appropriate time step must be accompanied by error control of the solution. To integrate Eq. (1) from $t=0$ to $t=T$, we partition the macroscopic time T (for instance the system size divided by the front speed) into small time steps ΔT determined by numerical instability constraints. Space is discretized into regions of length δx defining a set of spatial grid points.

The first step in the algorithm is the classification of spatial points according to the time step needed to obtain an accurate solution. We define a parameter e_{tol} which determines the acceptable error (or an estimate of it) over the time step ΔT. The error $e_{\max}$ is estimated using the difference between two numerical solutions. The first solution $\boldsymbol{w}_A$ is obtained using a numerical scheme of order n_t, with a fixed time step δt. For simplicity, δt is taken to be ΔT divided by an integer m. The second solution $\boldsymbol{w}_C$ is obtained with m twice as large and a time step of $\delta t/2$. (An implementation of such error estimates for an Euler scheme is given in Appendix A.) The solution at each spatial point is examined and is accepted if the absolute value of the difference between the two solutions is smaller than the control parameter e_{tol}. If the solution at a spatial point has an acceptable error then it need not be integrated with a smaller time step.

At this stage in the algorithm there are two categories of spatial point: points accepted or rejected by the error criterion. We must now recompute the solution at the rejected spatial points using a larger value of m, say m'. The value of m' is determined by extrapolation using a temporal scheme of order n_t and the maximum error $e_{\max}$. It cannot be smaller than $m+1$ and larger than a value that insures that errors do not accumulate because of a large number N_s of asynchronous steps. More specifically, m' may be determined from (see Appendix A)

$$m' = \min\left(1 + m\left(\frac{e_{\max}}{e_{\mathrm{tol}}} \right)^{1/n_t}, 1 + \mathcal{R}m \right), \tag{3}$$

with

$$\mathcal{R} = \left(\frac{m_l}{m_f} \right)^{1/(N_s - 1)}, \tag{4}$$

where m_f and m_l are the first and last values of m in the previous iteration.

We now perform two integrations using the new value m' and check the list of points previously rejected to determine those that may now be accepted. Before this can be done we must interpolate the solution values at the points which have been accepted previously to obtain estimates of $\boldsymbol{w}(\boldsymbol{r},t)$ at intermediate times corresponding to the smaller time increments in order to evaluate the Laplacian term. We use a second order polynomial interpolation in time and determine the expansion coefficients for each of the accepted points. The error estimation procedure yields the values of $\boldsymbol{w}(\boldsymbol{r},t)$ for at least three times t, $t+\Delta T/2$ and $t+\Delta T$. The expansion coefficients are defined by

$$w_k(\boldsymbol{r},t) = \sum_{i=0}^{2} a_{ik}(\boldsymbol{r}) t^i. \tag{5}$$

In this version of the algorithm, we fix the order of the interpolation scheme, taking advantage of the error estimation procedure. We may introduce a variable order interpolation scheme corresponding to the order of the integration scheme; however, its implementation is more difficult it makes added demands on the time and memory requirements of the algorithm.

The evaluation of the Laplacian term would be prohibitively time consuming if it were necessary to check that each of the points in the neighborhood of the given point had been accepted or rejected. To overcome this difficulty and avoid *all* loops over accepted points, we have used the following first order finite difference expression for the Laplacian term:

$$\frac{\partial^2 w_k(x,t)}{\partial x^2} = \frac{1}{\delta x^2}\left(-2w_k(x,t) + \sum_{i=0}^{2} \left(a_{ik}(x+\delta x) + a_{ik}(x-\delta x) \right) t^i \right),$$

where the interpolation coefficients are defined for rejected points by

$$\begin{aligned} a_{0k}(\boldsymbol{r}) &= w_k(\boldsymbol{r},t), \\ a_{1k}(\boldsymbol{r}) &= 0, \\ a_{2k}(\boldsymbol{r}) &= 0. \end{aligned} \tag{6}$$

We then simply need to update the arrays a_{0k} each time the w_k fields change at a point which has not yet been accepted.

We then set $m=m'$ and estimate a new value of m' and repeat the procedure until there are no rejected points left. Knowing the number of points which have been accepted after the first iteration we can estimate the next initial value of m, m_i, to avoid iterations where too many points will be rejected. We have obtained good results by varying m_i so that, on average, 85% of the points were accepted after the first step, and fixing $N_s=3$.

A. Algorithmic description of integration scheme

It is useful to present the method in algorithmic form. We introduce the following notation to simplify the presentation:

$\boldsymbol{w}^0$: Initial state at time t_i

$\boldsymbol{w}^1$: Final state at time t_e using m steps

$\boldsymbol{w}^{2/2}$: Final state at time t_e using $2m$ steps

$\boldsymbol{w}^{1/2}$: Intermediate state at time $(t_i+t_e)/2$ using m steps

ν_1: Constant$=1+1/(2^{n_t}-1)$

$\nu_{2/2}$: Constant$=-1/(2^{n_t}-1)$

μ_1: Constant$=1/n_t$

μ_2: Constant$=1/(N_s-1)$

The integration scheme requires that one control the errors during the asynchronous evolution. The integration of the system from time t_i to time t_e, with $\Delta T=t_e-t_i$, involves the following steps:

(i) First, we perform the initial synchronous step over all spatial points to determine the solutions at the three time steps needed to estimate the errors and compute the interpolation coefficients. Symbolically we may write

1 Calculation of $\boldsymbol{w}^1$
$m=m_i$
$\delta t=(t_e-t_i)/m$
$t_m=t_i$
$\boldsymbol{w}^1=\boldsymbol{w}^0$
LOOP m times
$\boldsymbol{w}^1\leftarrow$Evolution all $(\boldsymbol{w}^1,t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
2 Calculation of $\boldsymbol{w}^{1/2}$ and $\boldsymbol{w}^{2/2}$
$\boldsymbol{w}^{2/2}=\boldsymbol{w}^0$
$t_m=t_i$
$\delta t=\delta t/2$
LOOP m times
$\boldsymbol{w}^{2/2}\leftarrow$Evolution all $(\boldsymbol{w}^{2/2},t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
$\boldsymbol{w}^{1/2}=\boldsymbol{w}^{2/2}$
LOOP m times
$\boldsymbol{w}^{2/2}\leftarrow$Evolution all $(\boldsymbol{w}^{2/2},t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
GO TO III

Here ''Evolution all'' refers to the synchronous evolution of all points in the system.

The quantity m_i must be chosen so that numerical instability leads to an inaccurate, but not out-of-bound, value of the solution. An out-of-bound value will degrade the performance of the algorithm since it involves the calculation of arithmetic exceptions which are time consuming. Note, however, that this does not lead to a termination of the algorithm since the points where this occurs will be rejected.

(ii) The second element is the asynchronous evolution of the system involving points that were rejected in the error control procedure. Below ''Evolution rejected'' refers to such evolution.

1 Asynchronous calculation of $\boldsymbol{w}^1$
$\delta t=(t_e-t_i)/m$
$t_m=t_i$
$\boldsymbol{w}^1=\boldsymbol{w}^0$ for rejected points
LOOP m times
$\boldsymbol{w}^1\leftarrow$Evolution rejected $(\boldsymbol{w}^1,\boldsymbol{a}_i,t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
$\boldsymbol{a}_0=\boldsymbol{w}^1$ for rejected points
2 Calculation of $\boldsymbol{w}^{1/2}$ and $\boldsymbol{w}^{2/2}$
$\boldsymbol{w}^{2/2}=\boldsymbol{w}^0$ for rejected points
$\boldsymbol{a}_0=\boldsymbol{w}^1$ for rejected points
$t_m=t_i$
$\delta t=\delta t/2$
LOOP m times
$\boldsymbol{w}^{2/2}\leftarrow$Evolution rejected $(\boldsymbol{w}^{2/2},\boldsymbol{a}_i,t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
$\boldsymbol{a}_0=\boldsymbol{w}^{2/2}$ for rejected points
$\boldsymbol{w}^{1/2}=\boldsymbol{w}^{2/2}$ for rejected points
LOOP m times
$\boldsymbol{w}^{2/2}\leftarrow$Evolution rejected $(\boldsymbol{w}^{2/2},\boldsymbol{a}_i,t_m,\delta t,n_t)$
$t_m=t_m+\delta t$
$\boldsymbol{a}_0=\boldsymbol{w}^{2/2}$ for rejected points

(iii) This component of the algorithm specifies how the error calculation is to be carried out and how the list of rejected points is updated.

LOOP over active points $\boldsymbol{r}$
error$=|\boldsymbol{w}^{2/2}-\boldsymbol{w}^1|$
IF error $<e_{\text{tol}}$
$\boldsymbol{w}^1=\nu_{2/2}\boldsymbol{w}^{2/2}+\nu_1\boldsymbol{w}^1$
$\boldsymbol{a}_i\leftarrow$Interpolation $(\boldsymbol{w}^0,\boldsymbol{w}^{1/2},\boldsymbol{w}^1)$
ELSE
$\boldsymbol{a}_0=\boldsymbol{w}^0$
$\boldsymbol{r}$ added to the list of rejected point
IF error $<e_{\max}$
$e_{\max}=$error

The error introduced by the interpolation procedure is of order ΔT^3. The error introduced by the approximation of $\boldsymbol{w}$ at different times is approximately e_{tol} by construction. Since the points which have been first accepted will act like a boundary condition and can propagate the error to the rejected points (leading to convergence to the wrong solution defined by the approximate boundary condition), one may specify a smaller error, at least for the first step where most of the points will be accepted. In practice this feature does not change the performance of the algorithm significantly (see Sec. III).

(iv) The error control algorithm is very sensitive to the fast variations of the solution when m is odd for a multitime step algorithm, and always for a single time step algorithm (which reduces to the case $m=1$), so for the best detection and control of the numerical instability we should choose m odd. This can be achieved easily using integer division by $m=2(m/2)+1$. We may take $m_i=m_i\pm 2$ to preserve the odd character of m.

(v) This component of the algorithm describes how the parameters are updated.

IF number of rejected points $\neq 0$

$m_1=1+m(e_{\max}/e_{\mathrm{tol}})^{\mu_1}$

$m_2=1+\mathcal{R}m$

$m=\max(m_1,m_2)$

$m=m+1$, if m is even

GO TO II

ELSE

$\mathcal{R}=1+(m_i/m)^{\mu_2}$

depending on the number of rejected points after the first synchronous loop

$m_i=m_i\pm 2$ or is unchanged

Finally, we note that some optimization of the procedure can be achieved by retaining in memory the first time derivative at t_i since it will be called more than once for rejected points.

III. RESULTS

In this section we compare the efficiency of the asynchronous algorithm with a synchronous scheme for both homogeneous and inhomogeneous media in one and two dimensions. In making such comparisons one must insure for the given discretization that not only are the solutions stable but that they are free from unphysical high-frequency spatial and temporal oscillations.

A measure of the deviation from the physical branch of solutions is the integrated difference between the numerical pulse solution for a given spatial grid and the result for a high-order temporal integration for the same spatial grid, with any translation error eliminated by suitable displacement of the solutions. Figure 3 shows the integrated error function for three different grid sizes varying the time step.

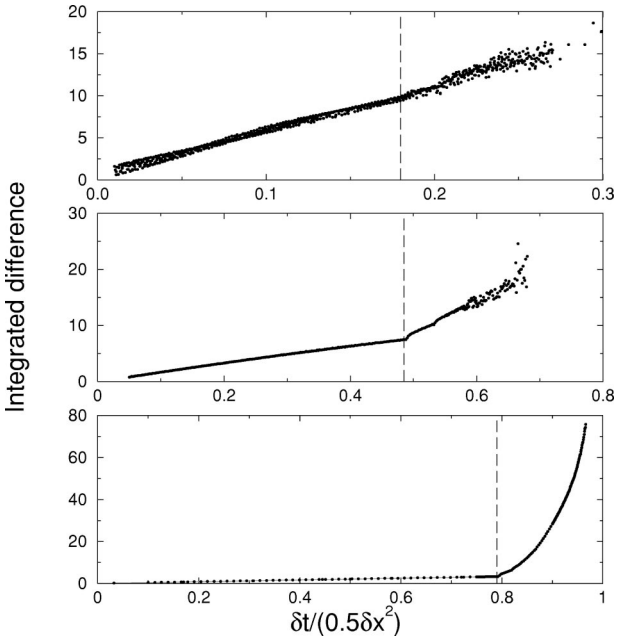

FIG. 3. Integrated difference between the numerical and exact shape of the front in the continuous time limit (obtained from very accurate solutions at fixed grid size) versus normalized time step ($\delta t/0.5\,\delta x^2=1$ is the instability limit in the continuous limit). From top to bottom $\delta x=1$, 0.5, and 0.25. The other parameters are the same as Fig. 1.

The maximum value of δt for which data is shown roughly corresponds to the stability limit δt_{sta} of the pulse solution for each grid size. Physically acceptable solutions which do not exhibit significant morphological differences from the continuous time limit solutions exist for δt values to the left of the dashed lines. The appearance of unphysical pulse solutions with high-frequency spatial oscillations, signaled by rapid increases in the error functions, are found for δt values to the right of the dashed lines. These unphysical solutions are stable from a dynamical systems point of view since all of the Lyapunov exponents are less than or equal to zero.[9,10] Thus, it may be dangerous to carry out simulations close to the numerical stability limit unless errors are controlled.

We must also insure that our finite grid numerical solutions can be taken to the continuous time limit in a consistent way. A natural and coherent way to converge to the continuous limit is to decrease the spatial grid size while adjusting the time step to satisfy the stability criteria. For an order n_t temporal scheme and an order n_s spatial scheme in d dimensions we show in Appendix B that the time step must satisfy,

$$\delta t^{n_t}=\frac{B}{A}\frac{n_s+1}{n_t d}\,\delta x^{n_s+1}\equiv\rho\,\delta x^{n_s+1}, \tag{7}$$

where A and B characterize the spatial and temporal errors.

Unless otherwise specified, the one-step temporal evolution will be carried out using Eq. (2).

A. One-dimensional homogeneous media

The full stability diagram of the pulse solution of the FHN model for the parameters in Fig. 1 is given in Fig. 4.

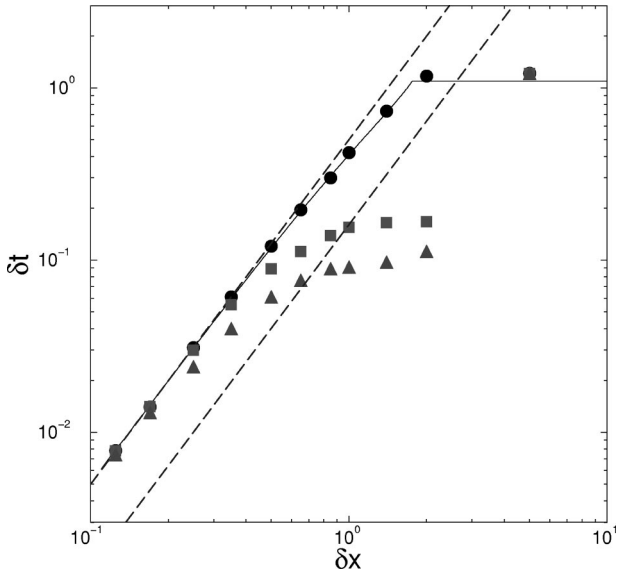

FIG. 4. Numerical stability analysis of the Euler scheme combined with first-order finite differences. The parameters are the same as Fig. 1. The solid line is the result of the analysis of the linearized equation for the uniform system in the rest state. This curve exhibits the expected behavior in the continuous limit $\delta t = 0.5\ \delta x^2$ (upper dashed line) and in the large grid size limit $\delta t < 2/|\lambda_{\max}|$. The intermediate region shows the discrepancy between the pure diffusive limit and the full problem including the Jacobian term. The circles correspond to the time step which leads to out-of-bound values in the simulation starting from steady state. This time step value is usually somewhat larger than that predicted by the linear stability analysis since the perturbation can saturate in the nonlinear stage of the evolution leading to unphysical solutions. The squares correspond to the stability limit defined as the first time step leading to out-of-bound values starting from the pulse solution. There is a significant difference between this curve and the result of the analysis of the rest state. There are no points for $\delta x > 2$ because the pulse solution cannot be sustained for such large values of δx. The triangles define the physical branch of the pulse solution. The lower dashed line is obtained from $\delta t = 2B/A\ \delta x^2 = 0.16\ \delta x^2$.

The circles correspond to the stability limit for the steady-state solution; i.e., the time step for a given grid size which leads to out-of-bound values of the variables starting from the steady state. The squares denote the stability limit for the pulse solution. These pulse values lie below the asymptotic stability limit of the steady state and the two results differ widely in region ***B*** in Fig. 2. For such pulse solutions one may expect that a domain decomposition algorithm will utilize a larger time step for regions that lie in the rest state and will decrease the time step in the excited part of the pulse solution. The triangles in Fig. 4 define the physical branch of the pulse solution. Provided the curve $\delta t = \rho\ \delta x^2$ lies below the physical branch, an explicit Euler scheme will not exhibit stability problems.

For the parameters in Fig. 1, namely $\alpha = 0.5$, $\beta = 0.75$, $\epsilon = 0.15$, $D_u = 1$ and $D_v = 0$, we may compare the Euler scheme with a fixed time step, the synchronous Euler scheme with a variable time step using the error control procedure, and the asynchronous algorithm. The system size is 200. Since this case probes the transition between regions ***A*** and ***B*** in Fig. 2 we fixed $\delta x = 1$ and $\delta t = 0.05$ which lies below the destabilization limit of the physical branch (~ 0.09) and below $\delta t = 0.16$ obtained using $\rho = 0.16$ which will yield the most efficient simulation (cf. Appendix B). We have performed 10^5 iterations providing a reference time unit for further comparisons. For the synchronous variable time step Euler scheme, we chose the parameter e_{tol} such that the average time step is the same as the fixed time step algorithm, i.e., $\Delta t = 0.05$ and $e_{\text{tol}} = 0.05$. The CPU time needed is 3.15 times longer than for the Euler scheme with a fixed time step, which mainly arises from the need to perform three iterations instead of one and the additional computations needed to carry out the error estimation.

TABLE I. Relative CPU times for different algorithms for the pulse solution in the homogeneous case. From top to bottom: synchronous Euler scheme with fixed time step, synchronous Euler scheme with variable time step controlled by e_{tol}, and the asynchronous variable time step Euler scheme controlled by e_{tol} and $e_{\text{tol}}^{\text{first}}$ for the first synchronous step. The velocity in the continuous *time* limit is ~ 2.540 and the relative error is determined with respect to this value. The local parameters are the same as in Fig. 1; $\delta x = 1$ and the system size $L = 200$.

Update	Time step	Δt	e_{tol}	$e_{\text{tol}}^{\text{first}}$	CPU/E	Vel	Error
sync	fixed	0.05	⋯	⋯	1.00	2.318	8.8%
sync	var	0.05	0.05	-	3.15	2.529	0.5%
async	var	0.50	0.05	0.01	1.05	2.495	1.5%

1. Simulation results: Coarse resolution

The asynchronous algorithm was implemented with a precision of $e_{\text{tol}} = 0.05$ over the time step fixed at the stability limit $\Delta t = 0.5$. Even though the time step is ten times larger, we do not demand a larger error since 0.05 is the maximum error allowed to constrain the numerical instability. (As noted earlier, the constraint on the allowed error after the first synchronous step can be larger, and we choose $e_{\text{tol}}^{\text{first}} = 0.01$.) The CPU time, normalized by its value for the fixed time step synchronous Euler scheme, (CPU/E), is 1.05 for this simulation. In this regime the algorithm is about three times faster than the synchronous version of the variable time step algorithm and takes about the same time as the simple fixed time step Euler scheme. We have checked that the solution does not have high-frequency unphysical oscillations. The velocities of the pulse solution are summarized in Table I. These results show that a better solution can be obtained for the same computational cost.

The efficiency of the algorithm will depend on factors such as the fractions of the system in the rest state and the excited states, the stiffness of the solution and the grid size. For instance, increasing the system size from 200 to 800, the ratio CPU/E is 0.64 instead of 1.05. The maximum speed up one may obtain is controlled by the ratio between the stability limit of the rest state and the pulse solution, divided by a factor of ~ 3 arising from the error control algorithm. For this particular case the asymptotic ratio is about 1/3. This ratio does not vary significantly with ϵ since both stability limits are roughly proportional to ϵ.

A similar analysis must be carried out for each system of interest and an implicit scheme may have to be used if the asymptotic ratio is about one.

TABLE II. CPU time and front speed using different algorithms for the 1D pulse solution for the same parameters as in Fig. 1 and $\delta x = 0.5$. The system size is 100 space units and less than 40% of the system is in the rest state. The ratio $\delta t/\delta_{\text{sta}} = 1/25$ for the upper part of this table. The velocity in the continuous time limit is 2.7368.

Scheme	Δt	e_{tol}	$e_{\text{tol}}^{\text{first}}$	CPU/E	CPU/S	Vel
Euler	0.005	⋯	⋯	1	0.32	2.7095
Sync	0.125	$6\ 10^{-3}$	⋯	3.10	1	2.73664
Async	0.125	$6\ 10^{-3}$	$6\ 10^{-4}$	0.38	0.12	2.7334

2. Simulation results: Continuous limit

Cases where $\rho \ll 1$ can be simulated efficiently using the asynchronous scheme since the error function associated with the pulse solution is sharply peaked in the vicinity of the pulse. Table II shows a comparison of the CPU times of the asynchronous and fixed time step Euler schemes for the FHN model with $\delta t/\delta t_{\text{sta}} = 1/25$, mimicking cases where $\rho = 1/50$. The asynchronous scheme is eight times faster than the synchronous version and three times faster than the Euler scheme. For this excitable medium model we could not find an example with very small ρ, but small ρ values are common in simulations of oscillatory media with a high-temporal frequency and a small wave length.

B. One-dimensional inhomogeneous media

Next consider an inhomogeneous medium where the diffusion coefficient varies in space. We assume the diffusion coefficient $D_v = 0$ while the spatial variation in D_u reflects a localized inhomogeneity centered at x_0, $D_u = D(x) = 1 + 9\exp(-(x-x_0)^2/200)$ (cf. top panel of Fig. 5).

In an inhomogeneous system with spatially dependent diffusion coefficient the evolution of a localized perturbation depends on the spatial location. As usual, to investigate the stability one must diagonalize the linear stability matrix, excluding the Jacobian term in the continuous limit.[11] For $\delta x = 0.5$, the instability threshold is such that $2\delta t/\delta x^2 \simeq 1/D_{\max}$. The first unstable eigenmode shown in Fig. 6 is localized, as are all the unstable eigenmodes for $1/D_{\max} < 2\delta t/\delta x^2 < 1/D_{\min}$. For $2\delta t/\delta x^2 > 1$, the first delocalized modes appear (see Fig. 7). There is one order of magnitude (roughly the ratio between the maximum and minimum values of the diffusion coefficient) between the time steps at which localized and delocalized modes appear.

Consequently, a time step which yields a stable solution will lead to inefficient simulation in those parts of the system which do not control the instability. The use of a variable

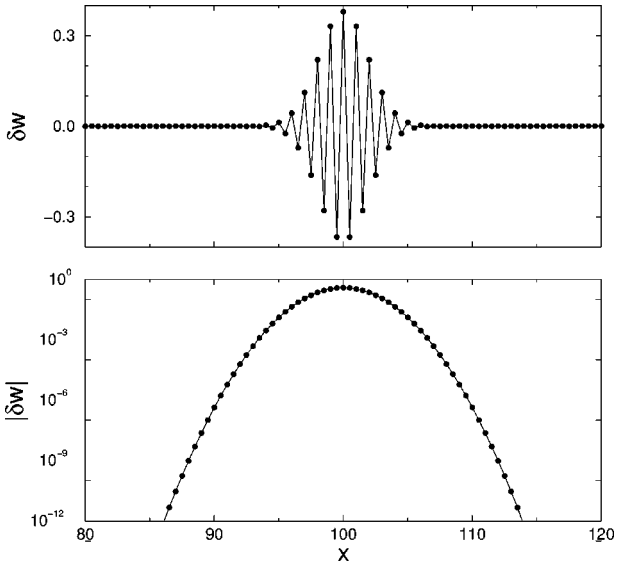

FIG. 6. First unstable eigenmode in the continuous limit for the system with a spatially dependent diffusion coefficient. Parameters are: $\delta x = 0.5$, $\delta t = 0.5\ \delta x^2/D_{\max}$.

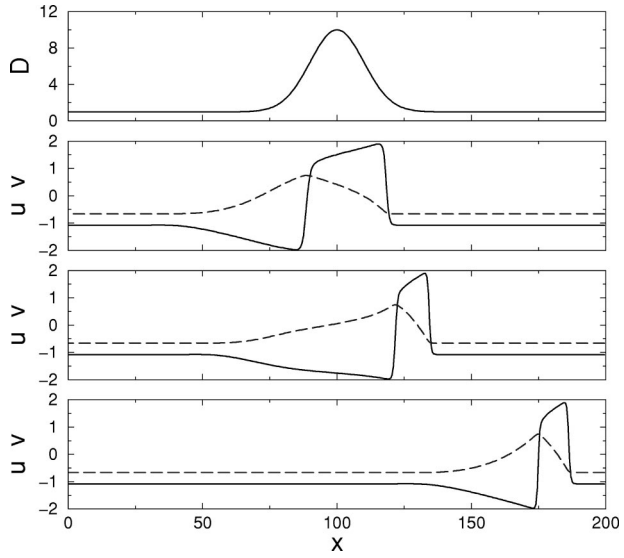

FIG. 5. Diffusion coefficient (upper curve) and pulse solution at three different times (lower curves) versus position for the FHN model. In the central spatial region the diffusion coefficient is ten times larger than elsewhere. One can clearly observe the variation of the action potential duration and refractory period as the pulse passes through the high-diffusion region.

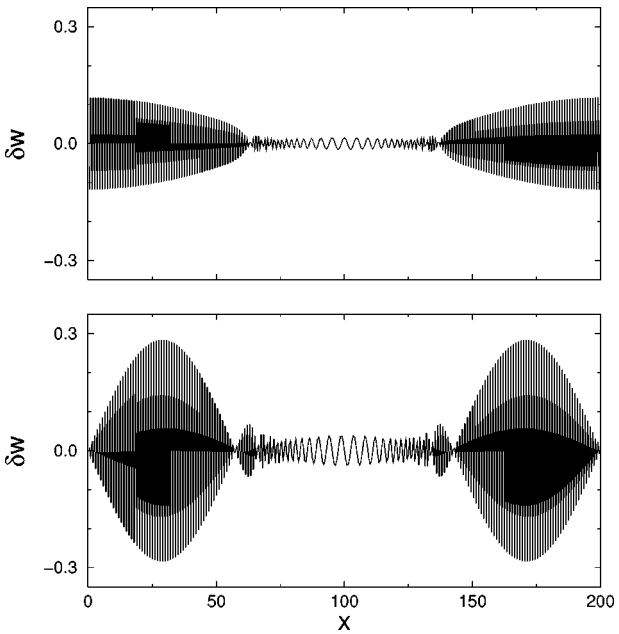

FIG. 7. Lowest unstable delocalized eigenmodes. Parameters are: $\delta x = 0.5$, $\delta t = 0.505\ \delta x^2$.

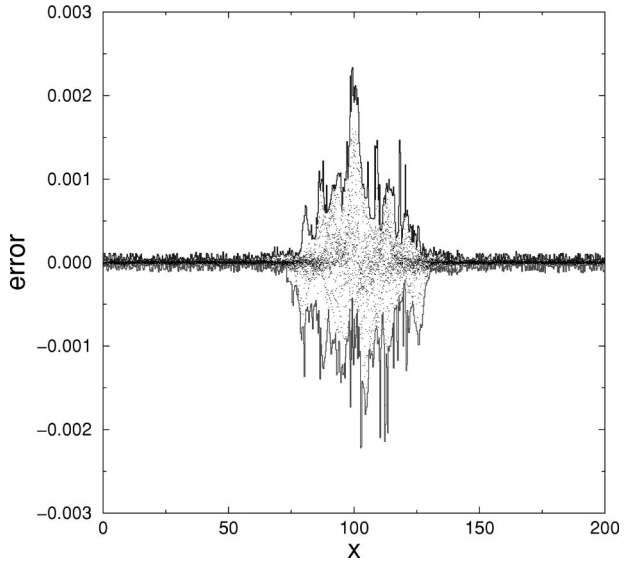

FIG. 8. Twenty different realizations of the error function (difference between uniform rest state and the numerical solution) using the asynchronous algorithm with $\Delta t=0.045$, $e_{\mathrm{tol}}=2\ 10^{-3}$ and $e_{\mathrm{tol}}^{\mathrm{first}}=2\ 10^{-4}$. The upper and lower solid lines mark the maximum and minimum values obtained for each lattice node.

grid size, i.e., varying the grid size δx such that $\delta x(x)/\sqrt{D(x)}\sim Cst$, is a possible solution to this problem. This is probably the most efficient solution in one-dimension (1D), and such a scheme was employed in a study of vortex filaments in order to deal with a numerical instability that arises in this problem.[12] Such a scheme requires a specific grid for each spatial distribution of the diffusion coefficient values which, in turn, requires extensive data manipulation making it difficult to implement the method efficiently in higher dimensions. However, the localization of the first unstable modes suggests that an efficient simulation may be carried out using domain decomposition.

1. Simulation results

We first analyze the efficiency of the algorithm for the integration of the inhomogeneous FHN system in the rest state. Figure 8 shows twenty instantaneous values of the error function obtained using the asynchronous algorithm with $\delta x=0.3$, $\Delta t=0.045$, $e_{\mathrm{tol}}=2\ 10^{-3}$ and $e_{\mathrm{tol}}^{\mathrm{first}}=2\ 10^{-4}$ for the same parameters as in Fig. 1.

The reference value of e_{tol} was fixed so that the average time step of the synchronous error control algorithm is just below the stability limit, i.e., $e_{\mathrm{tol}}\simeq 2\ 10^{-4}$ for $\Delta t=0.0045$. The time step for the appearance of the first delocalized mode is ten times larger, and one demands an error smaller than $2\ 10^{-3}$ over $\Delta t=0.045$. The error parameter for the first synchronous step was ten times smaller. This figure shows that the error control procedure constrains the error in the region of the system where the diffusion coefficient is large. One may also check that the time step is, on average, fifteen times smaller in this region than in the region where the diffusion coefficient is low. Using these parameters, about 75% of the sites are accepted after the first iteration.

Table III shows results similar to those obtained previously for the homogeneous system. For small system sizes (the high diffusion region occupies about 20% of the system length), the asynchronous scheme required as much CPU time as the fixed time step scheme but provides error control and, locally, is a second order scheme. Again the efficiency increases for larger systems (i.e., for systems where the high diffusion region is a smaller fraction of the total system size) and varies linearly with the ratio $D_{\max}/D_{\min}$. The efficiency can be increased further by varying these parameters.

TABLE III. Relative CPU times using the same algorithm as in Table I for the rest state in the inhomogeneous case. The system size $L=200$. The local parameters are the same as in Fig. 1; $\delta x=0.3$ and the nature of the inhomogeneous diffusion coefficient is described in the text.

Update	Time step	Δt	e_{tol}	$e_{\mathrm{tol}}^{\mathrm{first}}$	CPU/E
sync	fixed	0.0045	⋯	⋯	1.00
sync	var	0.0045	$2\ 10^{-4}$	⋯	3.40
async	var	0.0450	$2\ 10^{-3}$	$2\ 10^{-4}$	1.03

Next, we consider the performance of the asynchronous algorithm for the pulse solution. Results for various system sizes are given in Table IV. Comparisons are made between the Euler scheme with $\delta t=0.003$ and the synchronous error control algorithm with the same Δt and e_{tol} parameters. For the Euler scheme we did not use a time step of $0.16\ \delta x^2/D_{\max}$ since this will provide efficient simulation in the high diffusion region and very accurate temporal integration in the small-diffusion bulk of the system where the time step for efficient simulation, $0.16\ \delta x^2/D_{\min}$, is about ten time larger. We simply chose $\delta t=0.003$ close to the stability limit 0.0045 for $\delta x=0.3$. The parameter e_{tol} was fixed as discussed previously.

The results show that even for small system sizes ($L=512\times 0.3\simeq 160$ where the high-diffusion domain occupies 25% of the system length) the asynchronous algorithm easily compensates for the overhead incurred by the error control procedure and is faster than the Euler scheme. The ratio CPU/E varies roughly as $0.1+160/N$ and confirms that the maximum theoretical acceleration factor is given by $D_{\max}/D_{\min}$. We have also performed an analysis of the velocity. As in the homogeneous case the asynchronous scheme is more accurate than the Euler scheme.

We next consider more realistic and demanding simulations of two-dimensional homogeneous and inhomogeneous media.

TABLE IV. Relative CPU times varying the system size for the asynchronous scheme applied to the pulse solution propagating through the inhomogeneous medium (see text). The comparison Euler scheme used a fixed time step of $=3\ 10^{-3}$. The synchronous variable time step algorithm used the same time step Δt, the same error control parameter e_{tol}, and the same system size. The local parameters are the same as in Fig. 1 with $\delta x=0.3$.

N	Δt	e_{tol}	$e_{\mathrm{tol}}^{\mathrm{first}}$	CPU/E	CPU/S
512	0.03	$1\ 10^{-3}$	$1\ 10^{-4}$	0.93	0.41
666	0.03	$1\ 10^{-3}$	$1\ 10^{-4}$	0.88	0.34
1024	0.03	$1\ 10^{-3}$	$1\ 10^{-4}$	0.66	0.26
2048	0.03	$1\ 10^{-3}$	$1\ 10^{-4}$	0.52	0.17

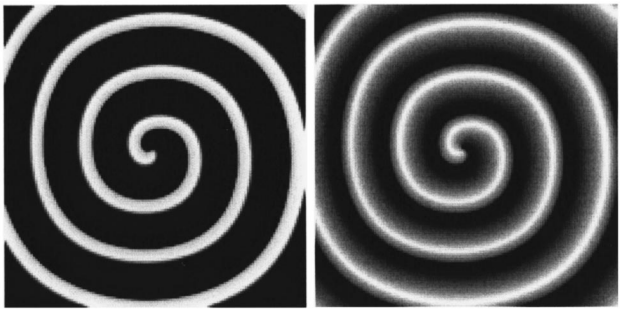

FIG. 9. The u (left) and v (right) fields of the spiral solution for the same parameters as in Fig. 1 with $\delta x=0.5$ and lattice size is 256×256. A linear gray scale is used from the minimum to the maximum of each field.

C. Two-dimensional homogeneous media

As an example of the simulation of wave propagation in two-dimensional media we examine spiral wave dynamics. Figure 9 shows both the u and v fields of a spiral wave using the same parameters as in Fig. 1.

There is almost no site in the rest state and one cannot expect faster simulation times because of the differences of the stability properties of the rest and the excited states, or at least not as large an acceleration factor as for the 1D case in the large system limit.

For $\delta x=0.5$ we fix $\delta t\simeq0.08$ $\delta x^2=0.02$ [extrapolating to two-dimensional (2D) the formulas for efficient simulation using the same set of local parameters] which is about 1/3 of the stability limit in the continuous limit $\delta t<0.25\delta x^2=0.0625$. In practice, for this value of the grid size, the maximum time step allowed is about 0.05. Setting $e_{\mathrm{tol}}\simeq0.01$, the synchronous error control algorithm leads to $\delta t\sim0.02$ on average and is about 3.3 times slower. The asynchronous version is 60% faster, but still slower than the Euler scheme. Data are summarized in Table V.

The asynchronous scheme is only about twice as fast as the synchronous version of the algorithm. These results demonstrate the influence of the solution properties on the efficiency of the algorithm. This is the situation when the explicit nature of the numerical scheme and its stability constraints prevent efficient simulation. Of course, for systems where the ratio B/A is much smaller than $1/D$, the stability limit is not a determining factor and the maximum acceleration is then dominated by the spatial variations in the error function, should such variations exist.

D. Two-dimensional inhomogeneous media

As an example of a more complex inhomogeneous medium we suppose that the diffusion coefficient varies randomly through most of the medium with values ranging between 0.75 and 1.25 and, in addition, there is a localized high diffusion domain (taken to have the form of the letter Y) where the maximum value of D_u is 10. Figure 10 (left panel) shows the D_u diffusion field using a nonlinear gray scale.

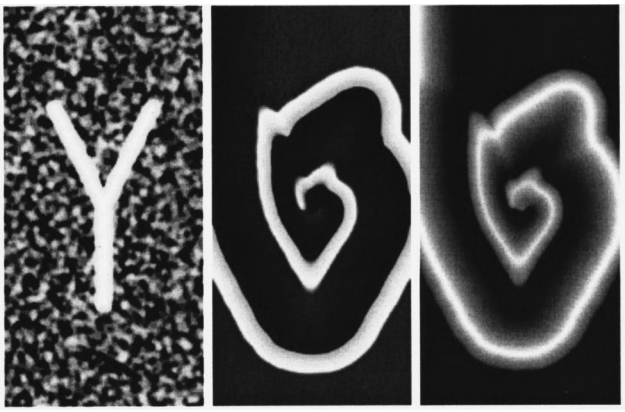

FIG. 10. Spiral solution in an inhomogeneous medium. The local reaction parameters are the same as in Fig. 1 with $D_v=0$. The D_u field (left panel) contains three longitudinal structures converging in the center of the lattice, forming a Y-shape where $D_u\sim10$. The background is such that $D_u\in[0.75,1.25]$. The u and v fields are shown in the center and right panels. $\delta x=0.5$ and the lattice size is 256×512 (the aspect ratio of the picture is preserved).

This example was chosen as a caricature of a piece of cardiac tissue where the inhomogeneity arises from the different conduction characteristics of cardiac cells in the tissue and the high-diffusion regions mimic specialized conduction channels like Bachman's bundle in the atrium or Purkinje fibers in the ventricles.[13]

Figure 10 also shows an instantaneous view of a spiral wave in such an inhomogeneous medium. One can see the deformation induced by the high conduction Y-shaped channel. Table VI summarizes the simulation time results and shows that the asynchronous algorithm is twice as fast as the Euler scheme and about six times faster than the synchronous version of the algorithm. As in the 1D case, the maximum value of the acceleration factor is controlled mainly by the ratio between the largest diffusion value and the average value.

IV. DISCUSSION

In the preceding sections we introduced criteria which allow one to distinguish regimes where use of the asynchronous scheme is appropriate for both homogeneous and inho-

TABLE V. Relative CPU times of different algorithms for the spiral solution in a homogeneous medium using the same parameters as in Fig. 1 with $\delta x=0.5$. The lattice size is 256×256 with no-flux boundary conditions.

Scheme	Δt	e_{tol}	$e_{\mathrm{tol}}^{\mathrm{first}}$	CPU/E	CPU/S
Euler	0.02	⋯	⋯	1	0.30
Sync	0.02	0.01	⋯	3.27	1
Async	0.08	0.04	0.01	1.42	0.43

TABLE VI. Relative CPU times of different algorithms for the spiral solution using the same parameters as in Fig. 1 with $\delta x=0.5$ for an inhomogeneous medium: maximum value of $D_u\sim10$; 80% of the sites are such that $D_u\in[0.75,1.25]$, and $D_v=0$ for all sites (see text for detail). The lattice size is 256×512 with no-flux boundary conditions.

Scheme	Δt	e_{tol}	$e_{\mathrm{tol}}^{\mathrm{first}}$	CPU/E	CPU/S
Euler	0.002	⋯	⋯	1	0.32
Sync	0.020	$5\ 10^{-4}$	⋯	3.15	1
Async	0.020	$5\ 10^{-4}$	$5\ 10^{-4}$	0.52	0.16

mogeneous media. For example, the ratio between the time step for efficient simulation and the stability threshold,

$$\mathcal{R}_1 = \frac{\delta t_{\text{eff}}}{\delta t_{\text{sta}}} = 2D\frac{B}{A},$$

determines when the conditionally stable character of the explicit Euler scheme is a handicap. We have shown in Appendix B how one may estimate A and B from computational data or approximately using the typical physical scales of the problem.

If $\mathcal{R}_1 \gg 1$ the time step is constrained by the stability limit and efficient simulations are not possible with explicit schemes. In this circumstance fully implicit or semi-implicit schemes can be used, but one must re-determine the A and B coefficients which are scheme dependent (usually A and B are larger for implicit schemes).

If $\mathcal{R}_1 \ll 1$ the simulation lies within the stability limit. The key factor to consider is the large fluctuations in the error function in different spatial domains. One may use larger time steps and/or grid sizes where the solution is smooth. For excitable media the magnitude of this effect depends on the fraction $\mathcal{R}_2$ of the medium in the rest state.

If $\mathcal{R}_1 \sim 1$ in homogeneous media, the problem is more complex because varying the time step without varying the grid size leads to a delocalized numerical instability. In the continuous limit it is not possible to constrain the numerical instability without varying the grid size ($\delta t \propto \delta x^2$). For a coarse resolution (large grid size) the stability limit is controlled by the reaction term and the fastest time scale. For excitable media this usually corresponds to the up-stroke of the wave (associated with the transition from rest to excited states). If the rest state occupies a large region of the system, for example, as is the case in normal cardiac wave propagation, one may perform faster simulations using the asynchronous scheme. The relevant criterion is the ratio between the stability threshold of the pulse solution and that for the system in the rest state

$$\mathcal{R}_3 = \delta t_{\text{sta}}^{\text{rest}} / \delta t_{\text{sta}}.$$

If this ratio is large the asynchronous scheme with fixed grid size will be efficient. The maximum theoretical acceleration factor varies like $1/(\mathcal{R}_2\mathcal{R}_3)$. If $\mathcal{R}_3 \approx 1$, one must use an implicit scheme for the reaction term and/or the first step of an asynchronous scheme. In this case an explicit variable grid size algorithm will be less useful since it does not lead to an increase of the time step.

For inhomogeneous systems we must introduce new parameters that characterize the localization of the unstable mode: $\mathcal{R}_4 = \delta t_{\text{sta}}^{\text{deloc}} / \delta t_{\text{sta}}^{\text{loc}}$, and the fraction of the system with high diffusion coefficient, $\mathcal{R}_5$. The corresponding efficiency criterion is

$$\mathcal{R}_1^{\text{deloc}} = \frac{\delta t_{\text{eff}}}{\delta t_{\text{sta}}^{\text{deloc}}} = 2D_{\min}\frac{B}{A}.$$

The ideal case corresponds to $\mathcal{R}_4 \gg 1$, $\mathcal{R}_5 \ll 1$ and $\mathcal{R}_1^{\text{deloc}} < 1$ and the asynchronous scheme will be efficient from fine to coarse resolutions.

A limitation of the algorithm we have presented is the utilization of the synchronous error control procedure which introduces an overhead factor of 3 in the simulation with respect to the fixed time step Euler scheme. Alternative versions of error control exist; for example, estimates based on the local spatial or temporal gradient or the fourth-order spatial derivative or the second order temporal derivative (since they appear in the error function). We have performed simulations using the first order temporal partial derivative for the error estimate. The codes execute roughly twice as fast as the asynchronous multistep algorithm.

A. Previous work

The literature on numerical schemes is large and it is difficult to make comparisons between algorithms because of the various criteria used to determine their efficiency. Below we comment on several schemes designed for the simulation of excitable media in order to put our asynchronous algorithm into perspective.

1. Adaptive mesh size refinement algorithm

The Adaptive Mesh size Refinement Algorithm has been applied recently to wave propagation in cardiac models in one and two dimensions by Cherry *et al.*[2] This algorithm is a multigrid size/time step algorithm where three different resolutions are fixed at the start of the simulation [three pairs of $(\delta x, \delta t)$]. Using error estimation, the algorithm dynamically fixes the resolution which must be used. Their results show how the algorithm is able to use the coarse and medium resolutions in the smoothest parts of the system. Reconnection between the different resolutions is carried out using interpolation of the missing time values.

They simulate a complex 2D regime where propagating wave break-up leads to a dynamical multispiral state. The integration is eleven times faster than the Euler scheme at the finest resolution. In the one-dimensional simulation, the front speeds agree to within 0.1%. This result was obtained using a very small time step ($\delta t = 0.003$ ms) relative to the grid size ($\delta x = 0.0125$ cm). The typical diffusion coefficient of the heart excitable medium is about 10^{-3} cm^2 ms^{-1}. This gives $\delta t_{\text{stab}} = \delta x^2/(2D) = 0.08$ ms, i.e., 26 times the value used in their simulation. The use of very small time step avoids problems related to numerical instability, but has to be justified on the basis of a discussion of ρ. A factor of 10 smaller in the time step than the pure diffusive theoretical limit is frequently employed and is not specific to this paper (see Ref. 7 for instance).

A linear variation of the time step relative to the grid size is recommended in Ref. 2. Since the error function varies like δx^2 and δt, the weight of the temporal error term decreases with respect to the spatial error term as one goes from the finest to the coarsest spatial resolution. From our point of view, this constraint plays the same role as our parameter $e_{\text{tol}}^{\text{first}}$ which must be chosen smaller than e_{tol} to limit the systematic error propagation.

2. Semi-implicit schemes

Fully implicit integration schemes are rarely used since they require minimization of a very large and nonlinear set

of equations, potentially leading to convergence difficulties. Semi-implicit schemes, also called mixed explicit–implicit schemes, are widely used for simulation of stiff systems of equations. In such schemes only few terms in Eq. (1) are implicitly integrated. For the reaction rates,[1,4] one may use immediate implicit integration when possible for a specific type of reaction term, or an iterative procedure which is usually efficient for a few-variable system. The use of implicit schemes for the gradient terms requires that one consider technical issues like matrix inversion[3] or multigrid iterations.[5]

References 1 and 4 describe an algorithm where a semi-implicit scheme is used, allowing a large time step for coarse resolution (Region *B* in Fig. 2). This algorithm is very powerful since it covers a very large range of grid sizes, from the partial differential equation continuous limit to the cellular automaton limit (the local variables are reduced to few discrete states). The method is limited because it relies on a specific form of the local reaction term used to carry out efficient implicit integration. The low accuracy of the implicit scheme is compensated by using a second-order implicit scheme; even the reaction term is integrated explicitly by a first order Euler scheme.

In the continuous limit, schemes of this type are dominated by the instability in the diffusion term. Implicit integration of the discretized diffusion term can be used to perform simulations in region *D*. Quantitative comparisons at constant error between the Euler scheme and semi-implicit schemes have not yet been carried out. Keener and Bogar[5] show that using a Crank–Nicolson scheme to integrate the bidomain equations in cardiac tissue, the simulation is about 40 times faster than that using a simple Euler scheme. The error in the front velocity is about 2.5% relative to the Euler scheme with same grid size and a time step close to the stability limit. They used an elegant and efficient form of the bidomain equation leading to a very fast algorithm that reduces the computational overhead by a factor of 2. Further analysis of the results is difficult since neither the solution in the continuous time and/or space limit nor the weight of the spatial discretization error term (criterion $\mathcal{R}_1$) are known.

3. Asynchronous mixed schemes

Although there are several differences, our algorithm belongs to the same class as that described by Quan *et al.*[6] (adaptive variable time step with fixed grid size). One of the main differences is the utilization of an implicit scheme for the first synchronous step. They show that this choice can be efficient for a system where the number of active sites is very small.

Their asynchronous evolution steps are different from ours since they allow any time step values between 0 and Δt for the sub-iterations requiring the management of ''a priority queue.'' Their algorithm is more complex because it mixes two different schemes, an Euler scheme and a Cooley–Dodge scheme with a modified alternating-direction-implicit method. Also, the efficient implementation of an error control procedure with an implicit scheme requires investigation.

TABLE VII. CPU times and front speeds of different algorithms for the 1D pulse solution using the same parameters as in Fig. 1 with $\delta x = 0.5$. The system size is 100 and less than 40% of the system is in the rest state. The ratio $\delta t/\delta_{\text{sta}} = 1/100$. RK2 refers to the second order fixed time step Runge–Kutta scheme. Sync.n refers to the synchronous multistep error control procedure using a Runge–Kutta scheme of order *n*. The third-order synchronous scheme requires on average five steps to reach the expected accuracy instead of 100 using first order scheme. The parameters are $e_{\text{tol}}^{\text{first}} = 0.1\, e_{\text{tol}}$ for the asynchronous scheme.

Scheme	Δt	e_{tol}	CPU/E	CPU/S	Vel
Euler	**0.00125**	···	**1**	**0.33**	**0.255%**
RK2	0.00125	···	1.97	0.65	0.004%
RK2	**0.0125**	···	**0.20**	**0.07**	**0.018%**
Sync.1	0.125	$1.5\,10^{-3}$	3.04	1	0.004%
Sync.3	0.125	$1.5\,10^{-3}$	0.53	0.18	0.004%
Sync.1	0.125	$1.5\,10^{-2}$	0.37	0.12	0.022%
Sync.3	0.125	$1.5\,10^{-2}$	0.28	0.09	0.004%
Async	**0.125**	$\mathbf{1.5\,10^{-3}}$	**0.19**	**0.07**	**0.033%**

A direct comparison with their algorithm is difficult since they have not considered the continuous time *and* space limit and model cardiac tissue by a discrete network of excitable cells connected by resistors. Time steps are estimated using the temporal derivative and a threshold corresponding to the maximum variation of the action potential through one iteration. They also include model specific criteria related to some of the ion gate variables.

In a regime with complex dynamics they obtain an acceleration factor of ~ 4 with respect to a synchronous implicit scheme. The overhead introduced by this scheme in comparison to the simple Euler method is not discussed.

B. Higher-order schemes

As shown in Table II, asynchronous schemes using an explicit first iterate are efficient for homogeneous systems in the continuous limit if $\mathcal{R}_1 \ll 1$. In this case utilization of a higher order scheme must be considered as a possible way to improve significantly the numerical integration.

Table VII summarizes the relative CPU times and relative errors in the front speed using the same parameters as in Fig. 1 for a relatively small system ($L = 100$) in order to decrease the number of grid points in the rest state. The table shows that higher-order temporal schemes perform very well in terms of CPU time versus relative error in the front speed.

Since the order of the temporal scheme is larger than the ''coherence'' value with respect to error function and the stability limit criteria for the explicit scheme, the time step is quickly dominated by the stability limit, fixing the maximum acceleration factor to $1/\mathcal{R}_1$ divided by the overhead induced by the high-order scheme. For instance, the RK2 scheme with a time step ten times larger than that for the Euler scheme executes five times faster and provides more accurate results. Note that we assume in this case that $\mathcal{R}_1 = 1/100$. This implies that the error in the front speed arising from the spatial discretization is also about 0.25%. Synchronous schemes also perform well when one reduces the accuracy from $1.5\,10^{-3}$ to $1.5\,10^{-2}$. Finally, our asynchronous scheme yields results equivalent to the best fixed time step scheme, demonstrating its efficiency. One cannot easily ex-

tend the performance of higher-order schemes to cases where $\mathcal{R}_1 \sim 1$ since a prefactor in the stability condition must be taken into account.

C. Remarks

We have presented a simple asynchronous algorithm and discussed the dynamical regimes where it will yield efficient simulations. The efficiency was defined with respect to the ''maximum or theoretical acceleration factor,'' taking into account its variation with system parameters (size, local stiffness, stability ratios $\mathcal{R}_i$, ...).

We have shown that the scheme will be efficient in three different cases:

(i) For homogeneous media, with coarse resolution, when the pulse solution is stiffer than rest state and most of the system is in the rest state,
(ii) for homogeneous/inhomogeneous media in the continuous limit when $\mathcal{R}_1 \ll 1$ and the error function is sharply peaked,
(iii) for inhomogeneous media in the continuous limit when localized inhomogeneities lead to localized unstable eigenmodes.

We have mainly limited our results to cases where $\mathcal{R}_1 \approx 1$ and, indeed, it is difficult to find other examples for excitable media. Should this be a general property of excitable media, this would constrain significantly the type of efficient simulation one may perform for such systems.

The method described in this paper does not rely on the form of the local dynamics, is easily implemented for arbitrary geometries and should prove useful for simulations of a variety of problems such as wave propagation in cardiac tissue which is an inhomogeneous medium with complex local stiff dynamics and a complex geometry.

ACKNOWLEDGMENT

This work was supported in part by a grant from the Canadian Network of Centers of Excellence on Mathematics of Information Technology and Complex Systems (MITACS).

APPENDIX A: VARIABLE TIME STEP ALGORITHM

1. Single time step

One may estimate the error in a simulation by performing two integrations with different time steps as follows: Let $\boldsymbol{w}_A$ be the solution at t. To obtain the solution at $t+\delta t$ one may take one step using the Euler scheme with time step δt, or two steps using the time step $\delta t/2$. We let $\boldsymbol{w}_B$ and $\boldsymbol{w}_C$, respectively, be these two estimates:

$$\boldsymbol{w}_B(x,t+\delta t)=\boldsymbol{w}_A(x,t)+\delta t\frac{\partial \boldsymbol{w}_A}{\partial t}(x,t),$$

$$\boldsymbol{w}_C(x,t+\delta t)=\boldsymbol{w}_A(x,t)+\frac{\delta t}{2}\frac{\partial \boldsymbol{w}_A}{\partial t}(x,t)+\frac{\delta t}{2}\frac{\partial}{\partial t}\left(\boldsymbol{w}_A(x,t)+\frac{\delta t}{2}\frac{\partial \boldsymbol{w}_A}{\partial t}(x,t)\right).$$

The following relations among $\boldsymbol{w}_A(x,t+\delta t)$, $\boldsymbol{w}_B(x,t+\delta t)$ and $\boldsymbol{w}_C(x,t+\delta t)$ apply

$$\boldsymbol{w}_A(x,t+\delta t)=\boldsymbol{w}_B(x,t+\delta t)-\frac{\delta t^2}{2}\frac{\partial^2 \boldsymbol{w}_A(x,t)}{\partial t^2}+\mathcal{O}(\delta t^3),$$

$$\boldsymbol{w}_A(x,t+\delta t)=\boldsymbol{w}_C(x,t+\delta t)-\frac{\delta t^2}{4}\frac{\partial^2 \boldsymbol{w}_A(x,t)}{\partial t^2}+\mathcal{O}(\delta t^3).$$

The difference between $\boldsymbol{w}_B$ and $\boldsymbol{w}_C$ is approximately the same as between $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$, or $\boldsymbol{w}_A$ and $\boldsymbol{w}_C$. One can impose the condition that distance between $\boldsymbol{w}_B$ and $\boldsymbol{w}_C$ must never be larger than a predetermined amount e_{tol}. If the distance between $\boldsymbol{w}_B$ and $\boldsymbol{w}_C$ is too large, then the solution is not accepted and the time step is decreased. Since the error function can be very inhomogeneous for extended systems, one may consider the maximum value of the difference over the grid points, $e_{\max}=\max_{\boldsymbol{r}}|\boldsymbol{w}_B(\boldsymbol{r})-\boldsymbol{w}_C(\boldsymbol{r})|$.

Taking into account the first-order nature of the Euler scheme the new time step $\delta t'$ is given by

$$\delta t' = c\,\delta t\sqrt{\frac{e_{\mathrm{tol}}}{e_{\max}}}. \tag{A1}$$

The prefactor $c<1$ (typically $c\approx 0.9$) will suppress the contributions of higher-order terms if they are not too large. If $e_{\max}$ is less than e_{tol}, the solution is accepted but the time step is increased using the same formula since it could be too small if $e_{\max}\ll e_{\mathrm{tol}}$. Finally, these two solutions can be used to obtain more accurate estimates of $\boldsymbol{w}_A(x,t+\delta t)=2\boldsymbol{w}_C(x,t+\delta t)-\boldsymbol{w}_B(x,t+\delta t)+\mathcal{O}(\delta t^3)$, by canceling δt^2 terms.

For a temporal scheme of order n_t, the instantaneous and cumulative first error terms, respectively, vary as

$$\delta t^{n_t+1}\frac{\partial^{n_t+1}\boldsymbol{w}}{\partial t^{n_t+1}} \quad \text{and} \quad \Delta T\,\delta t^{n_t}\frac{\partial^{n_t+1}\boldsymbol{w}}{\partial t^{n_t+1}}.$$

Then the estimate of $\delta t'$ is

$$\delta t' = c\,\delta t\left(\frac{e_{\mathrm{tol}}}{e_{\max}}\right)^{1/(n_t+1)}, \tag{A2}$$

and the general linear combination of $\boldsymbol{w}_B$ and $\boldsymbol{w}_C$ is given by

$$\boldsymbol{w}_A(x,t+\delta t)=\frac{2^{n_t}}{2^{n_t}-1}\boldsymbol{w}_C(x,t+\delta t)-\frac{1}{2^{n_t}-1}\boldsymbol{w}_B(x,t+\delta t)+\mathcal{O}(\delta t^{n_t+2}).$$

2. Multitime step control

The previous solution will lead to an average time step over which the error is bounded by e_{tol}. This average time step depends on the order of the scheme. It is then more useful to fix the maximum error E_{tol} over the time step ΔT. A simple way to do this is to impose a maximum error varying like $e_{\mathrm{tol}}=E_{\mathrm{tol}}\,\delta t/\Delta T$ over each time step δt. Then Eq. (A2) becomes

$$\delta t' = c\,\delta t\left(\frac{\delta t_{\mathrm{old}}E_{\mathrm{tol}}}{\Delta T e_{\max}}\right)^{1/n_t}. \tag{A3}$$

In this case a linear combination of solutions is made each time an iteration δt is accepted. We may also form a linear combination of solutions every ΔT assuming a linear accumulation of the highest order error term. If ΔT is too large this method will fail. In the algorithm we used, the error is controlled only at $t+\Delta T$ and one must fix the number m of time steps into which Δt will be split. The extrapolation of the new value of m, m', is done assuming linear accumulation of the error terms as described above and for a temporal scheme of order n_t we have

$$m' = 1 + m\left(\frac{e_{\max}}{e_{\text{tol}}}\right)^{1/n_t},$$

where the $+1$ term serves the same function as the c prefactor.

To insure that errors do not accumulate because of the possibly large number of asynchronous steps, we limited (on average) the number of refinement levels N_s by placing a limit on the value of m'. Defining

$$\mathcal{R} = \left(\frac{m_l}{m_f}\right)^{1/(N_s-1)}, \tag{A4}$$

where m_f is the first and m_l the last value of m in the previous iteration, we choose,

$$m' = \min\left(1 + m\left(\frac{e_{\max}}{e_{\text{tol}}}\right)^{1/n_t}, 1+\mathcal{R}m\right). \tag{A5}$$

The control of the accuracy of the solution is done only in time. Therefore, the error tolerance must decrease as the continuous limit is approached. A simple way to accomplish this consists in fixing the grid size δx^0 and choosing e_{tol}^0 so that the numerical solution is on the physical branch and the simulation is efficient. Then e_{tol} just has to be decreased relative to the order of spatial discretization scheme n_s, i.e.,

$$e_{\text{tol}} = e_{\text{tol}}^0\left(\frac{\delta x}{\delta x^0}\right)^{n_s+1}.$$

This procedure will not apply for larger values of δx since the solution may not remain on the physical branch. The guideline we adopt is that e_{tol} should not be larger than a few percent of the typical value of $\boldsymbol{w}$ since perturbations were assumed to be small.

APPENDIX B: CONVERGENCE TO THE CONTINUOUS LIMIT

The finite-resolution numerical solution must be sufficiently close to the continuous space and time solution of the reaction diffusion equation for applications to physical problems. The convergence to the continuous limit must be carried out by taking into account both spatial and temporal discretization. One would also like to determine the most appropriate time step that is consistent with an accurate solution but with minimal computational cost. The following procedure may be used to determine the most effective way to approach the continuum limit.

The determination of the error terms arising from discretization are difficult to estimate in general since one requires a knowledge of the complete solution. Consider first a simple Euler scheme with a first-order finite difference approximation to the Laplacian. The cumulative error E using linear response has the form

$$E \propto A\,\delta t + B\,\delta x^2. \tag{B1}$$

A natural way to converge to the continuous limit is to decrease the grid size while adapting the time step to satisfy the stability criteria. One can see that fixing $\delta t = \rho\,\delta x^2$ both error terms vary like δx^2 and the error will vanish as the grid size goes to zero. Here ρ is a constant which must be fixed so that the solution always lies on the physical branch. This defines a ''coherent'' convergence to the continuous limit.

Generalizing the problem to higher dimensions, the computational cost using Euler scheme and first-order finite differences is proportional to

$$\text{Cost} \propto 1/(\delta x^d\,\delta t) = 1/(\delta x^{d+2}\rho). \tag{B2}$$

Substituting the expression for δt and δx as function of ρ and the cost variable we find

$$E \propto (\text{Cost})^{-2/d+2}(A\rho + B)\rho^{-2/d+2}. \tag{B3}$$

Keeping the computational cost constant we can minimize the error function as a function of ρ giving $\rho = 2B/Ad$, and thus

$$\delta t = \frac{2B}{Ad}\,\delta x^2. \tag{B4}$$

In this development, we used only the coherence between the temporal and spatial schemes, so that convergence to the continuous limit is carried out without excessive accuracy in the space or time integrations.

We can generalize the previous demonstration to a temporal scheme of order n_t and a spatial discretization of order n_s: The cumulative error function should behave as

$$E \propto A\,\delta t^{n_t} + B\,\delta x^{n_s+1}, \tag{B5}$$

which in terms of $\rho = \delta t^{n_t}/\delta x^{n_s+1}$ and CPU cost gives

$$E \propto (A\rho + B)\rho^{-\gamma_1}(\text{Cost})^{-\gamma_2},$$

where

$$\gamma_1 = \frac{n_s+1}{n_t d + n_s + 1}, \gamma_2 = \frac{n_t(n_s+1)}{n_t d + n_s + 1}.$$

Then the appropriate ratio between the time step and the grid size is

$$\frac{\delta t^{n_t}}{\delta x^{n_s+1}} = \frac{B}{A}\frac{n_s+1}{d n_t}.$$

The stability limit provides a strong limitation on this result since, depending of the values of A and B, we may have a very inefficient simulation with a time step that is much smaller than necessary. One should note that ρ varies as the inverse of the spatial dimension. This is consistent with the variation of the stability limit for the continuous limit with the inverse of the dimension d. For the Euler scheme and first-order finite difference form of the Laplacian operator, one has $\delta t = \delta x^2/(2d)$.

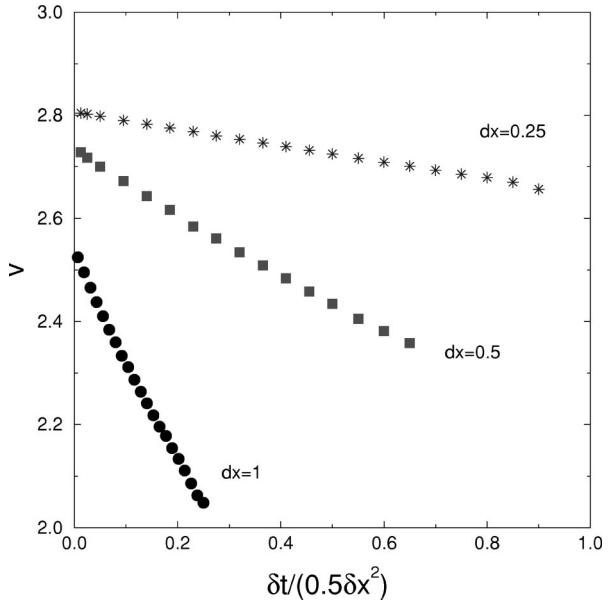

FIG. 11. Velocity of the excited wave versus the time step divided by the stability limit of the diffusion equation 0.5 δx^2, for three different values of the grid size. The other parameters are the same as Fig. 1.

As a result of these considerations it is important to find an appropriate value for ρ that leads to the smallest error for the lowest computational cost.

We now illustrate the application of these ideas to the FHN model computations. Figure 11 gives results for the velocity of the pulse for $n_t=1$ and $n_s=1$ in one dimension as a function of δt for several values of δx. We can interpolate the velocity curves for $\delta x=0.5$ and $\delta x=0.25$ (the two upper curves) for values of δt between 0 and $\delta x^2/4$. Assuming a quadratic dependence in δx and linear dependence in δt, we can estimate the ratio $B/A\simeq 0.08$. In this case the appropriate value of δt is

$$\delta t \simeq \frac{0.16}{d}\,\delta x^2.$$

For the three different values of δx used to construct Fig. 3, the numerical solution obtained with this time step lies on the physical branch.

This estimate of ρ requires a knowledge of the velocity of the pulse for four values of δt and δx (note that 1D results can be used for calculations in higher dimensions). Instead, one may prefer to use the one-step error expression which, for an Euler scheme with first order finite differences, is given by

$$e_k(x,t,\delta t,\delta x)=\frac{\delta t^2}{2}\,\frac{\partial^2 w_k}{\partial t^2}+\frac{\delta t D_k \delta x^2}{12}\,\frac{\partial^4 w_k}{\partial x^4}. \qquad \text{(B6)}$$

Knowing one ''physically'' acceptable solution, one can estimate the fourth order spatial derivative and the second order temporal derivative using the equality for a propagating wave, $\partial^2 w_k/\partial t^2 = V^2\,\partial^2 w_k/\partial x^2$, and then compare the two functions $(\frac{1}{2})\,\partial^2 w_k/\partial t^2$ and $(D_k/12)\,\partial^4 w_k/\partial x^4$. Using a variety of criteria such as the ratio of the maxima, the ratio of the integral of the absolute value, or the average of the ratio at each point where these functions are not zero, one obtains the estimate, $B/A\approx 0.12$, which is in good agreement with the previous estimate. The difference arises from the sensitivity of the neutral mode to the two different kinds of perturbation.

In the absence of such information one may obtain a crude approximation to ρ from a knowledge of the physical characteristics of the excited wave. The up-stroke part of the wave gives rise to the dominant contribution to the error function. In the case under consideration, the transition between the rest and excited states occurs over a typical length scale as $\Delta x\approx 1$ and the velocity of propagation is $V\approx 2.8$. From dimensional considerations one may estimate B/A by

$$\frac{B}{A}\sim\frac{1}{6}\,\frac{D_k}{V^2\Delta x^2}\approx 0.02.$$

While crude, this estimate of B/A, and thus ρ, is obtained at almost no cost.

As an example consider wave propagation in the heart. The conduction velocity in cardiac muscle is about 0.5 m/s, the diffusion coefficient is of order 10^{-4} m^2/s, and the gap in the membrane potential between the rest state and the excited state is about 0.1 V. Using 100 V/s for the up-stroke velocity we find $B/A\approx 200$, which must be compared to the equivalent stability limit ratio, i.e., $1/D\approx 10^4$. This estimate indicates that both error terms in Eq. (B5) contribute to the approximation to the continuous limit solution. This is probably a common property of an excited pulse, since both diffusion and reaction terms are significant in the up-stroke part of the pulse solution.

[1] D. Barkley, Physica D **49**, 61 (1991).
[2] E. M. Cherry, H. S. Greenside, and C. S. Henriquez, Phys. Rev. Lett. **84**, 1343 (2000).
[3] F. Fenton and A. Karma, Chaos **8**, 20 (1998).
[4] M. Dowle, R. M. Mantel, and D. Barkley, Int. J. Bif. Chaos **7**, 2529 (1997).
[5] J. P. Keener and K. Bogar, Chaos **8**, 234 (1998).
[6] W. Quan, S. Evans, and M. Hastings, IEEE Trans. Biomed. Eng., **45**, 372 (1998).
[7] A. Xu and M. R. Guevara, Chaos **8**, 157 (1998).
[8] See, W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1996), for a discussion and implementation of implicit and explicit schemes.
[9] A. M. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis* (Cambridge University Press, Cambridge, 1996).
[10] The discrete dynamical system we consider as an approximation of the continuous system is subject to a series of bifurcations which may even lead to chaotic dynamics with positive Lyapunov exponents. These bifurcations are the precursors of the numerical instability which occurs for larger values of δt.
[11] We confirmed numerically that the gradient term in Eq. (1) which appears because the diffusion coefficient is no longer constant does not significantly change the properties we discuss in this section. For the sake of simplicity we shall neglect it and concentrate on the pure reaction-diffusion form.
[12] G. Rousseau, H. Chaté, and R. Kapral, Phys. Rev. Lett. **80**, 5671 (1998); ''Twisted vortex filaments in the three-dimensional complex Ginzburg–Landau equation'' (in preparation).
[13] *Cardiac Electrophysiology: From Cell to Bedside*, edited by D. P. Zipes and J. Jalife (Saunders, Philadelphia, 1995).