SIAM J. Sci. Comput.
Vol. 35, No. 3, pp. B623–B643

EXPONENTIAL TIME DIFFERENCING FOR HODGKIN–HUXLEY-LIKE ODES[*]

CHRISTOPH BÖRGERS[†] AND ALEXANDER R. NECTOW[‡]

Abstract. Several authors have proposed the use of exponential time differencing (ETD) for Hodgkin–Huxley-like partial and ordinary differential equations (PDEs and ODEs). For Hodgkin–Huxley-like PDEs, ETD is attractive because it can deal effectively with the stiffness issues that diffusion gives rise to. However, large neuronal networks are often simulated assuming "space-clamped" neurons, i.e., using the Hodgkin–Huxley ODEs, in which there are no diffusion terms. Our goal is to clarify whether ETD is a good idea even in that case. We present a numerical comparison of first- and second-order ETD with standard explicit time-stepping schemes (Euler's method, the midpoint method, and the classical fourth-order Runge–Kutta method). We find that in the standard schemes, the stable computation of the very rapid rising phase of the action potential often forces time steps of a small fraction of a millisecond. This can result in an expensive calculation yielding greater overall accuracy than needed. Although it is tempting at first to try to address this issue with adaptive or fully implicit time-stepping, we argue that neither is effective here. The main advantage of ETD for Hodgkin–Huxley-like systems of ODEs is that it allows underresolution of the rising phase of the action potential without causing instability, using time steps on the order of one millisecond. When high quantitative accuracy is not necessary and perhaps, because of modeling inaccuracies, not even useful, ETD allows much faster simulations than standard explicit time-stepping schemes. The second-order ETD scheme is found to be substantially more accurate than the first-order one even for large values of Δt.

Key words. Hodgkin–Huxley equations, computational neuroscience, exponential time differencing, stiffness

AMS subject classifications. 92-08, 65L04

DOI. 10.1137/120883657

1. Introduction and overview.

Systems of Hodgkin–Huxley-like ordinary differential equations (ODEs), modeling neurons or neuronal networks, are commonly solved in computational neuroscience with simple explicit numerical methods, often using a fixed time step Δt. The value of Δt is usually on the order of 0.01 ms—much shorter than even a voltage spike (Table 1). Why does it have to be so small? We investigate this question through computational experiments for Hodgkin–Huxley-like model neurons, as well as networks of such model neurons. We specify our test problems and the numerical methods that we use to solve them in section 2.

Great quantitative precision may not, at this point, be a sensible aim in computational neuroscience, since there is substantial uncertainty about model parameters, translating into even greater uncertainty about the solutions of the model equations; see section 3.2 for illustrations of this point. The proper goal of the computational simulation of neuronal networks is, therefore, at the present time more likely to be qualitative insight than quantitative precision.

[*] Submitted to the journal's Computational Methods in Science and Engineering section July 6, 2012; accepted for publication (in revised form) March 13, 2013; published electronically June 4, 2013.
http://www.siam.org/journals/sisc/35-3/88365.html

[†] Department of Mathematics, Tufts University, Medford, MA 02155 (cborgers@tufts.edu). This author's work was supported in part by the Collaborative Research in Computational Neuroscience (CRCNS) program through NIH grant 1R01 NS067199.

[‡] Department of Biomedical Engineering, Tufts University, Medford, MA 02155. Current address: Rockefeller University, New York, NY 10021 (anectow@mail.rockefeller.edu).

Table 1
Examples of methods and time steps used in the literature to solve Hodgkin–Huxley-like systems.

Reference	Method	Δt [ms]
Börgers, Epstein, and Kopell [1]	midpoint method (2nd-order)	0.01
Hasegawa [9]	4th-order Runge–Kutta	0.01
Ho et al. [14]	Euler's method (1st order)	0.001–0.01
Hodgkin and Huxley [13]	Hartree's method [8]	0.02–1.0
Kopell et al. [16]	midpoint method (2nd-order)	0.02
Rubin and Wechselberger [24]	4th-order Runge–Kutta	≤ 0.01
Tiesinga, José, and Sejnowski [27]	2nd- and 4th-order Runge–Kutta	0.01
Traub et al. [28]	2nd-order Taylor series method	0.002
Wang and Buzsáki [31]	4th-order Runge–Kutta	0.05

However, even though there is probably no need for great accuracy in most contexts in computational neuroscience, simple, explicit numerical methods for Hodgkin–Huxley-like systems often do require Δt to be on the order of 0.01 ms. Significantly larger values of Δt can easily lead to catastrophic breakdown of the computations; see section 4. On the other hand, we also demonstrate in section 4 that time steps on the order of 0.01 ms often give more accuracy than is likely to be useful, in view of the modeling uncertainties. Thus stability constraints force us into paying for more accuracy than we need.

Not surprisingly, the time step is constrained primarily by the voltage spikes. In section 5, we demonstrate that between voltage spikes, $\Delta t = 1$ ms gives perfectly adequate accuracy for our model problems. Ironically, therefore, what makes the solution of Hodgkin–Huxley-like ODEs expensive is the need to compute spike shapes, over and over again, even though those shapes are largely stereotypical, i.e., almost the same for each spike, and well known a priori. Furthermore, in many situations, it is unnecessary to know the precise spike shape; what matters is mostly whether or not there is a spike. This, of course, is the reason for the popularity of integrate-and-fire models, in which spike shapes are not approximated at all. In section 6, we show that the breakdown of the standard explicit methods for $\Delta t \gg 0.01$ ms is caused specifically by the *rising phase* of the voltage spike: The rapid rise of the membrane potential often lasts no longer than a few times 0.01 ms. We present numerical results in section 6 suggesting that the fundamental cause of trouble with larger values of Δt is overshoot of the membrane potential during the rising phase, triggering instability.

In view of these observations, a natural approach would be to use adaptive time steps, so that at least one could use larger time steps between spikes. However, in a network containing many neurons, conventional step size control strategies would force small time steps for the whole network whenever a single neuron spikes, thereby destroying much of the advantage of using adaptive time-stepping, unless the spiking is synchronous. We give a numerical example illustrating this point in section 7. An alternative is to abandon the requirement that the spike shapes be resolved accurately and to look for a method which, while capable of producing high accuracy when Δt is very small, quickly and efficiently produces crude approximations to the spike shapes and good accuracy between spikes with much larger Δt. Fully implicit methods are not a good option here because of the difficulty of solving the nonlinear systems that would arise in each time step: To achieve convergence of iterative methods used for solving those nonlinear systems, we would need to impose the same time step constraint that simple explicit methods require for stability. In section 8, we illustrate this point with a numerical experiment for a Hodgkin–Huxley-like system and analyze it for the logistic equation.

The idea of exponential time differencing (ETD) is to freeze, in each time step, some of the variables, in such a way that the equations become linear, and then solve analytically over the time interval of duration Δt [5, 11]. To derive ETD schemes for Hokdgin-Huxley-like systems of ODEs, we exploit a special feature of such systems: Each variable appears linearly in the equation governing its time evolution.[1]

The simplest ETD scheme for Hodgkin–Huxley-like ODEs, called the "exponential Euler method," is the default time-stepping method in the software packages CSIM [19] and GENESIS [2]. We prove in section 9.2 that it is unconditionally stable for Hodgkin–Huxley-like ODEs, i.e., guaranteed to prevent the kind of overshoot that constrains the time step in the simpler methods. Its accuracy in the limit as $\Delta t \to 0$ was analyzed by Oh and French [22]. Using numerical experiments, we demonstrate in section 9 that it allows, for our model equations, time steps as large as 1 ms (or even greater), of course at the expense of not resolving the spike shapes accurately. We also propose a second-order accurate "exponential midpoint" method. Oh and French [22] suggested and analyzed a similar method; their method has good, but not unconditional, stability. We propose a modification of their method which restores unconditional stability by preventing voltage overshoot even in the preliminary half time step of the midpoint method. We present numerical results indicating that this method, too, allows the use of time steps as large as 1 ms and that it often yields substantially better results, even for large Δt, than the exponential Euler method.

In summary, we conclude that the advantage of ETD schemes for Hodgkin–Huxley-like ODEs lies in their unconditional stability specifically during the rising phase of the action potential. With time steps many times larger than those commonly used, ETD can still produce qualitatively correct, even if quantitatively somewhat crude, results, taking a fraction of the time required for a simulation with schemes such as Euler's method, the midpoint method, or the classical fourth-order Runge–Kutta method (RK4). Furthermore, the ETD schemes are not much more complicated or costly than the simple, standard explicit schemes.

In section 10, we discuss a "semi-implicit" (SI) version of Euler's method (see also section 2.3.3) which has, for Hodgkin–Huxley-like systems, properties similar to those of the exponential Euler method. However, we have not been able to generalize this method to second-order accuracy, preserving unconditional stability. In section 11, we analyze the exponential and SI methods for a one-dimensional model equation, confirming the conclusions from our numerical experiments, and in section 12, we put our results into the context of related work by others.

2. Test problems and numerical methods.

2.1. Neuronal models.

We report on numerical experiments with Hodgkin–Huxley-like model neurons, and networks of such model neurons, in this paper. Since the models are taken from the literature, we will not state all the details here but will give references for some of them. Both model neurons are of the form of the classical Hodgkin–Huxley ODEs [13] with the simplifying assumption that the activation

[1] Strictly speaking, in our Hodgkin–Huxley-like models, the membrane potential v appears nonlinearly in the evolution equation for v because the models rely on the simplifying assumption that the gating variable m is a direct function of v; see (2.1). We hide this fact by writing "m" on the right-hand side of (2.1), not "$m_\infty(v)$." No complications arise as a result.

variable, m, of the sodium current is a direct function of v:

$$C\frac{dv}{dt} = g_{Na}m^3h(v_{Na}-v) + g_K n^4(v_K - v) + g_L(v_L - v) + I, \tag{2.1}$$

$$\frac{dx}{dt} = \frac{x_\infty(v) - x}{\tau_x(v)} \qquad \text{for} \quad x = h, n, \qquad m = \left(m_\infty(v)\right)^3. \tag{2.2}$$

The letters v, t and τ, C, g, and I denote voltage (membrane potential), time, capacitance density, conductance density, and current density, respectively, measured in mV, ms, μF/cm^2, mS/cm^2, and μA/cm^2. For simplicity, we will often omit units from here on. The functions x_∞ and τ_x always satisfy

$$x_\infty(v) \in (0,1) \quad \text{and} \quad \tau_x(v) > 0 \quad \text{for all } v. \tag{2.3}$$

As in the classical Hodgkin–Huxley model, the gating variables m and n are "activation variables," i.e., $m_\infty(v)$ and $n_\infty(v)$ are increasing functions of v, while h is an "inactivation variable," i.e., $h_\infty(v)$ is a decreasing function of v. We assume that

$$C,\ g_{Na},\ g_K,\ g_L \ \text{ are positive.} \tag{2.4}$$

These parameters, as well as v_{Na}, v_K, and v_L, will be specified next.

2.1.1. Reduced Traub–Miles neuron. The reduced Traub–Miles (RTM) model, due to Ermentrout and Kopell [7], is a reduction of a model of a pyramidal cell in rat hippocampus proposed by Traub and Miles in [30]. We use a variation stated in complete detail in [16, Appendix 1] (see also [23]). The parameters are $C = 1$, $g_K = 80$, $g_{Na} = 100$, $g_L = 0.1$, $v_K = -100$, $v_{Na} = 50$, and $v_L = -67$. Our choices of I will be specified later. For the definitions of $x_\infty(v) \in (0,1)$ $(x = m, h, n)$ and $\tau_x(v) > 0$ $(x = h, n)$, we refer to [23] or [16, Appendix 1].

2.1.2. Wang–Buzsáki neuron. In the Wang–Buzsáki (WB) model of an inhibitory basket cell in rat hippocampus [31], the parameters are $C = 1$, $g_{Na} = 35$, $g_K = 9$, $g_L = 0.1$, $v_{Na} = 55$, $v_K = -90$, and $v_L = -65$. Note in particular that the conductance densities g_{Na} and g_K are much smaller than in the RTM model. For the definitions of x_∞ and τ_x, see [31] or [16, Appendix 1].

2.2. E/I networks. We will also study networks of 160 RTM and 40 WB neurons, which we refer to as "E-cells" (for "excitatory cells") and "I-cells" (for "inhibitory cells"), respectively. We adopt the synaptic model of [7] with parameter values as in [16, Appendix 1]. In particular, the rise and decay times are 0.1 ms and 3 ms for excitatory synapses and 0.3 ms and 9 ms for inhibitory ones, and the reversal potentials are 0 mV for excitatory synapses and −80 mV for inhibitory ones. Connectivity is chosen at random: For any pair of neurons, A and B, the probability that B receives synaptic input from A is 1/4, provided that at least one of the two neurons is inhibitory; we omit E → E-connections. Other parameters are chosen to produce a "gamma frequency" (~ 40 Hz) network oscillation. The drive to the jth E-cell is

$$I_{E,j} = 2 + 0.25X_j,$$

where the X_j $(j = 1, 2, \ldots, 160)$ are independent Gaussians with mean 0 and standard deviation 1. The drives to all I-cells are zero. Using the notation of [16], the strengths of the synapses are characterized by

$$g_{EE} = 0, \quad g_{EI} = 0.2, \quad g_{IE} = 0.5, \quad g_{II} = 0.1.$$

For instance, $g_{EI} = 0.2$ means that the sum of the maximal conductances associated with all excitatory synapses affecting a given I-cell has the expected value 0.2. All

E → I synapses have the same strength, but the total number of E-cells giving input to a given I-cell is random because connectivity is sparse and random.

2.3. Numerical methods.

We use the explicit and implicit Euler, midpoint, and classical RK4 methods, as well as the exponential and SI integrators defined below. We fix the time step $\Delta t > 0$ throughout, except in section 7, where we use the `ode23` function of MATLAB, which is adaptive. We write v_j, n_j, and h_j for the numerical approximations for $v(j\Delta t)$, $n(j\Delta t)$, and $h(j\Delta t)$. We also write $m_j = m_\infty(v_j)$ and $t_j = j\Delta t$.

2.3.1. Exponential Euler method. In the exponential Euler method, given v_j, h_j, and n_j, one analytically solves the linear initial-value problem

$$C\frac{d\tilde{v}}{dt} = g_{Na} m_j^3 h_j (v_{Na} - \tilde{v}) + g_K n_j^4 (v_K - \tilde{v}) + g_L(v_L - \tilde{v}) + I, \tag{2.5}$$

$$\frac{d\tilde{x}}{dt} = \frac{x_\infty(v_j) - \tilde{x}}{\tau_x(v_j)}, \quad x = h, n, \tag{2.6}$$

$$\tilde{v}(j\Delta t) = v_j, \quad \tilde{h}(j\Delta t) = h_j, \quad \tilde{n}(j\Delta t) = n_j,$$

and then sets

$$v_{j+1} = \tilde{v}((j+1)\Delta t), \quad h_{j+1} = \tilde{h}((j+1)\Delta t), \quad n_{j+1} = \tilde{n}((j+1)\Delta t).$$

2.3.2. Exponential midpoint method. In our version of the exponential midpoint method, given v_j, h_j, and n_j, we analytically solve

$$C\frac{d\tilde{v}}{dt} = g_{Na} m_{j+1/2}^3 h_{j+1/2} (v_{Na} - \tilde{v}) + g_K n_{j+1/2}^4 (v_K - \tilde{v}) + g_L(v_L - \tilde{v}) + I, \tag{2.7}$$

$$\frac{d\tilde{x}}{dt} = \frac{x_\infty(v_{j+1/2}) - \tilde{x}}{\tau_x(v_{j+1/2})}, \quad x = h, n, \tag{2.8}$$

$$\tilde{v}(j\Delta t) = v_j, \quad \tilde{h}(j\Delta t) = h_j, \quad \tilde{n}(j\Delta t) = n_j,$$

where $v_{j+1/2}$, $h_{j+1/2}$, and $n_{j+1/2}$ are computed using a step of the exponential Euler method with time step $\Delta t/2$, and $m_{j+1/2}$ stands for $m_\infty(v_{j+1/2})$. We then define

$$v_{j+1} = \tilde{v}((j+1)\Delta t), \quad h_{j+1} = \tilde{h}((j+1)\Delta t), \quad n_{j+1} = \tilde{n}((j+1)\Delta t).$$

Oh and French [22] used the explicit Euler method for the preliminary half step.

2.3.3. SI Euler method. We consider a variation of the Euler method in which each dependent variable is treated implicitly in the equation describing its time evolution but explicitly in all other equations:

$$C\frac{v_{j+1} - v_j}{\Delta t} = g_{Na} m_j^3 h_j (v_{Na} - v_{j+1}) + g_K n_j^4 (v_K - v_{j+1}) + g_L(v_L - v_{j+1}) + I, \tag{2.9}$$

$$\frac{x_{j+1} - x_j}{\Delta t} = \frac{x_\infty(v_j) - x_{j+1}}{\tau_x(v_j)}, \quad x = h, n. \tag{2.10}$$

Note that these equations are simple and inexpensive to solve, since they are linear in v_{j+1} and x_{j+1}, respectively. (Note that on the right-hand side of (2.9), $m_j = m_\infty(v_j)$ appears, not m_{j+1}.)

We do not discuss an "SI midpoint method" here because we have not been able to construct one that has stability properties similar to those of the ETD schemes and the SI Euler method (see Proposition 9.1, sections 10 and 11).

2.3.4. Numerical computation of firing frequency. When the goal is to track the membrane potentials of spiking neurons accurately, there is no question that very small time steps are needed. However, often less detailed information is desired in computational neuroscience. The simplest example is the computation of the frequency f of a periodically firing neuron. To compute f, we simulate a sufficiently long time interval (in the calculations presented in this paper, we take it to be a 300 ms interval) and determine the difference T between the second-to-last and last spike times. (See below for a discussion of how we define and compute spike times.) T is the period of the neuron. The frequency f is computed from the formula $f = 1000/T$. The factor of 1000 is needed because we follow the convention, common in neuroscience, of measuring time in ms, but frequency in Hz = s^{-1}.

We define the spike times of a neuron to be the times at which the action potential v is 0 with $dv/dt > 0$. The slight arbitrariness of this convention does not, of course, affect the computed periods. Since we will compute frequencies using the RK4 method in some of our numerical experiments and would like to verify fourth-order accuracy, we need to approximate spike times with at least fourth-order accuracy. When computing firing frequencies of individual neurons, we therefore determine spike times as follows. Suppose that $t_j = j\Delta t$, and v_j is the computed approximation for $v(t_j)$, $j = 0, 1, 2, \dots$. If $k \geq 1$ and $v_k < 0 \leq v_{k+1}$, we define $p = p(t)$ to be the cubic polynomial with $p(t_j) = v_j$ for $j = k-1$, k, $k+1$, and $k+2$, and use the bisection method to find a solution t^* of $p(t) = 0$ with $t_k < t^* \leq t_{k+1}$, with rounding error accuracy. Because the interpolating polynomial p is cubic, this procedure computes spike times with fourth-order accuracy, provided that the v_j are computed with fourth-order accuracy as well.[2]

3. Properties of solutions of Hodgkin–Huxley-like ODEs.

3.1. Bounding box. Since we are interested in whether discretizations of the differential equations allow over- or under-shoot (see section 1), we first state simple bounds on v, h, and n valid for the differential equations themselves: Under reasonable assumptions on I, the trajectory (v, h, n) cannot leave the box $(v_K, v_{Na}) \times (0,1) \times (0,1)$ if it starts in this box.

PROPOSITION 3.1. *Let* (v, h, n) *be a solution of* (2.1) *and* (2.2), *and assume* (2.3), (2.4), *and*

$$-g_L(v_L - v_K) < I < g_L(v_{Na} - v_L). \tag{3.1}$$

If $(v(0), h(0), n(0)) \in (v_K, v_{Na}) \times (0,1) \times (0,1)$, *then* $(v(t), h(t), n(t)) \in (v_K, v_{Na}) \times (0,1) \times (0,1)$ *for all* $t \geq 0$.

Proof. Using (2.3), (2.2) implies that $dx/dt > 0$ when $x = 0$, and $dx/dt < 0$ when $x = 1$, $x = h, n$. Therefore $x(t) \in (0,1)$ for all $t \geq 0$ if $x(0) \in (0,1)$. Then (2.1), together with (3.1), implies $dv/dt > 0$ when $v = v_K$ and $dv/dt < 0$ when $v = v_{Na}$. Thus the vector field points into the box in all points on the boundary of the box. This implies the assertion. ▯

[2] When computing and plotting spike *rastergrams*, it is unnecessary to determine spike times with that much care. Our *network* codes therefore compute the spike time, more conventionally, by determining the time at which the line connecting (t_k, v_k) with (t_{k+1}, v_{k+1}) crosses the line $v = 0$.

Table 2
Percentage change in neuronal spiking frequency when one model parameter is multiplied by 1.01.

	C	g_K	g_{Na}	g_L	v_K	v_{Na}	v_L	I
RTM	$-.95$	$-.03$	.19	.16	$-.79$	.08	-6.20	.63
WB	$-.22$	$-.46$	.33	$-.52$	-1.67	$-.04$	-8.38	.89

Values of I outside the range given by the inequalities in (3.1) are of very little interest. For the RTM neuron, (3.1) becomes $-3.3 < I < 11.7$; the spiking threshold is slightly above 0.1, and for $I = 11.7$, the firing frequency is about 232 Hz—much higher than typical for neurons in the brain under most circumstances. For the WB neuron, (3.1) becomes $-2.5 < I < 12$; the spiking threshold is slightly above 0.15, and for $I = 12$, the firing frequency is about 314 Hz.

3.2. Parameter dependence. Hodgkin–Huxley-like models describe biological reality only approximately, not with great quantitative precision. The proper goal of most numerical simulations in neuroscience should therefore, at this point, be qualitative insight, not quantitative accuracy. Fairly large numerical errors may be acceptable, as long as the computed solutions are qualitatively correct.

This reasoning is, of course, not always right. For instance, sometimes the differential equations themselves are the primary subject of interest, and then it is important to be able to obtain solutions with high accuracy. Ideally, a numerical method should therefore be able to obtain high accuracy if one needs it and is willing to pay the computational price for it, but also be able to quickly and inexpensively obtain rough but qualitatively correct approximations. This is useful, for instance, for a quick preliminary exploration of a high-dimensional parameter space.

To illustrate the uncertainty in the models of section 2.1, we consider the sensitivity of the firing frequency to changes in the eight parameters C, g_K, g_{Na}, g_L, v_K, v_{Na}, v_L, and I. We start with the parameter values of section 2.1, using (arbitrarily) $I = 0.7$. This yields firing frequencies of $f_0 \approx 35$ Hz for the RTM neuron, and $f_0 \approx 44$ Hz for the WB neuron. We then multiply one of the eight parameters by 1.01, while leaving all others unchanged, determine the resulting firing frequency f, and compute the percentage change, $(f - f_0)/f_0 \times 100$. The results, recorded in Table 2, show that the firing frequency is in fact remarkably insensitive to many of the parameters: Often the firing frequency changes by less than one percent when a parameter is changed by one percent. The firing frequency is, however, quite sensitive to the reversal potential of the leak current, v_L. One percent uncertainty about the value of v_L translates into 6.20 percent uncertainty about the firing frequency for the RTM neuron and 8.38 percent uncertainty for the WB neuron. There is no reason to think that, for instance, the RTM neuron with $v_L = -67$ is a better model of reality than the same model with $v_L = -67 \times 1.01 = -67.67$. One may therefore be content, for many purposes, with numerical simulations that come within 5 percent accuracy or so, as long as they are qualitatively correct.

4. Often stability, not accuracy, constrains the time step.

Figure 4.1 shows projections into the (v, n)-plane of approximate solutions of the RTM and WB models, with $I = 0.7$, obtained with time steps Δt increasing from left to right. The top three rows of the figure show results for the RTM model, obtained using the explicit Euler method (first row), the midpoint method (second row), and RK4 (third row). The bottom three rows show analogous results for the WB model. The rightmost panel in each row shows results of a highly inaccurate and nearly unstable

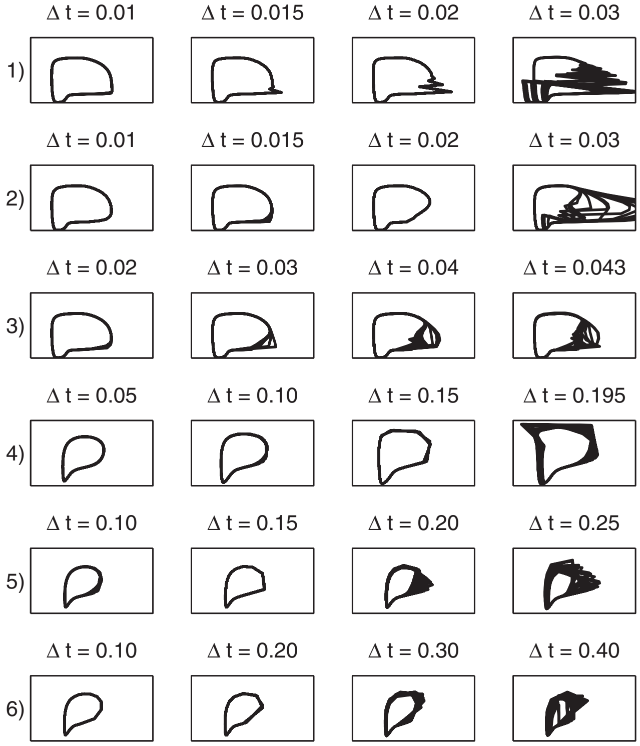

FIG. 4.1. *Projections into the (v, n)-plane of computed approximations of limit cycles of the RTM and WB neurons, obtained using three different numerical methods and various values of Δt.* (1) *RTM, explicit Euler,* (2) *RTM, midpoint method,* (3) *RTM, RK4,* (4) *WB, explicit Euler,* (5) *WB, midpoint method, and* (6) *WB, RK4. In all cases, $I = 0.7$. If Δt were raised just slightly in any of the plots in the right-most column, an overflow error would result.*

calculation; in each case, a very slight further increase in Δt would lead to an overflow error.

The results show that for the three standard methods, applied to the RTM model, time steps Δt much greater than 0.01 ms result in catastrophic instability. For the WB neuron, there is a similar instability, but it occurs at significantly greater values of Δt; see section 6 for an explanation of this difference between the two models.

Nothing would be wrong with time steps Δt on the order of 0.01 ms if accuracy requirements dictated so small a time step anyway. However, we will now present results suggesting that as soon as Δt is so small that the calculation is stable, the accuracy may be greater than necessary for most purposes. This is illustrated by Figure 4.2, panel A, which shows the percentage error in the computed frequency of an RTM neuron ($I = 0.7$) as a function of Δt, in a log-log plot. As Δt increases, the accuracy deteriorates, but just before the calculations break up as a result of catastrophic instability, accuracy is still very good—the errors are much smaller than, say, five percent (indicated by the dashed horizontal line in Figure 4.2), and therefore probably smaller than necessary (see section 3.2). Thus the time step size is dictated by stability, not accuracy.

However, Figure 4.2, panel B, demonstrates that stability considerations are not *always* the most constraining factor. The figure shows numerical experiments for the WB model. For this model, if the goal is to reach about five percent accuracy, the

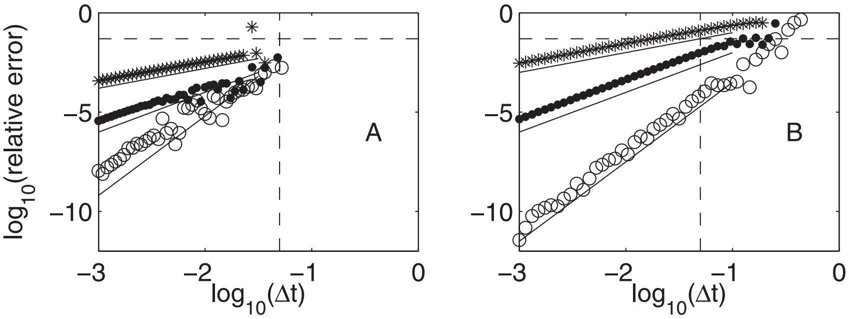

FIG. 4.2. *Log-log plots of relative error in computed firing frequency, as a function of Δt, for the explicit Euler method (stars), the midpoint method (dots), and RK*4 *(circles), for the RTM (panel A) and WB (panel B) models. The solid lines are of slopes* $1, 2,$ *and* 4*, and were added to confirm that the three methods give first, second, and fourth-order accurate approximations to the frequency. (Since the scaling is not the same on both axes, actual slopes seen in the figures differ from* $1, 2,$ *and* 4*.) Dashed horizontal line: relative error* $= 0.05$ *(i.e., five percent error). Dashed vertical line:* $\Delta t = 0.05$*. For the RTM neuron, calculations with* $\Delta t > 0.05$ *result in catastrophic instability in all cases.*

choice of time step is dictated by accuracy, not stability: As Δt increases and the accuracy deteriorates, a five-percent error level is reached *before* stability is lost.

5. Between action potentials, large time steps yield good accuracy.

Panel A of Figure 5.1 shows a voltage trace of the RTM neuron with $I = 0.7$ as a solid line, and an approximation computed using $\Delta t = 1$ as dots. The computation with $\Delta t = 1$ was started immediately after a voltage spike and gave very good results up to the time of the next spike. At that time, an overflow error occurred. Panel B of the same figure shows results of an analogous numerical experiment with the WB model, where $\Delta t = 2$ was used in between spikes.

These figures demonstrate that the need to resolve spike shapes dictates the choice of Δt; between spikes, much larger time steps give adequate accuracy. Thus most of the effort is spent on computing voltage spikes. This should not and need not be the case: The voltage spike shapes are sterotypical, i.e., they are known a priori with good accuracy. There should be no need to expend significant computational resources on computing them over and over again.

In both panels of Figure 5.1, we took Δt close to the limit: Increasing Δt by 1 ms (to 2 ms in panel A, 3 ms in panel B) would result in catastrophic instability. This

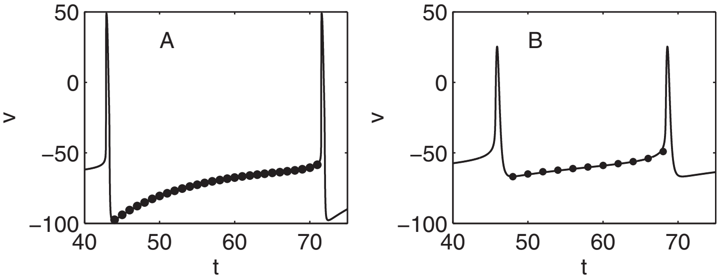

FIG. 5.1. A: *Voltage trace of RTM neuron computed using the midpoint method with* $\Delta t = 0.002$ *(solid line) and* $\Delta t = 1$ *(dots).* B: *Voltage trace of the WB neuron computed using the midpoint method with* $\Delta t = 0.002$ *(solid line) and* $\Delta t = 2$ *(dots).*

is a different instability than that shown in Figure 4.1. It is related to the fact that the motion *toward* the subthreshold part of the limit cycle is fast in comparison with the motion *along* the subthreshold part of the limit cycle. By contrast, the instability shown in Figure 4.1, which makes itself felt at much smaller values of Δt already and therefore constrains Δt much more severely, is related to the very fast motion *along* the limit cycle during the upstroke of the action potential; see section 6.

6. The rising phase of the action potential is the primary source of instability. A closer look at the instability shown in Figure 4.1 suggests that the main difficulty is the *rising phase* of the action potential. For the RTM model, the rising phase of the action potential is extremely brief, on the order of 0.03 ms; see Figure 6.1, panel A. The dashed horizontal lines in Figure 6.1 and in subsequent figures indicate v_K and v_{Na}, the bounds on v in the continuous case; see Proposition 3.1. Figure 6.2 shows voltage traces computed using the explicit Euler method with $\Delta t = 0.01$, 0.02, 0.03. For $\Delta t = 0.04$, a catastrophic instability sets in, and there is an overflow error. It is interesting to look more closely at the computed voltage spikes when the voltage overshoots. The bottom panel of Figure 6.2 shows a close-up look at the computation with $\Delta t = 0.02$. There is zig-zagging, very much like that seen when solving a stiff problem using an explicit method with slightly too large a value of Δt.

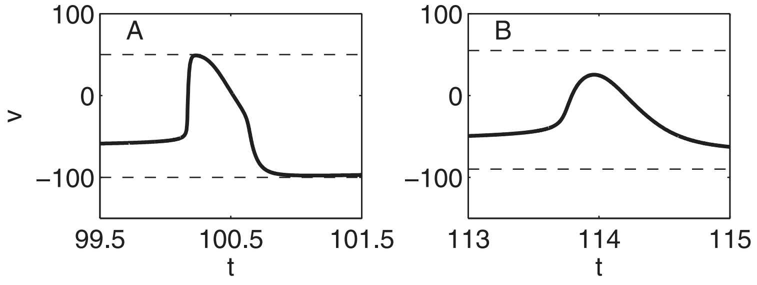

FIG. 6.1. *A single voltage spike computed using the midpoint method with* $\Delta t = 0.002$. A: *RTM neuron,* B: *WB neuron, with* $I = 0.7$ *in both cases.*

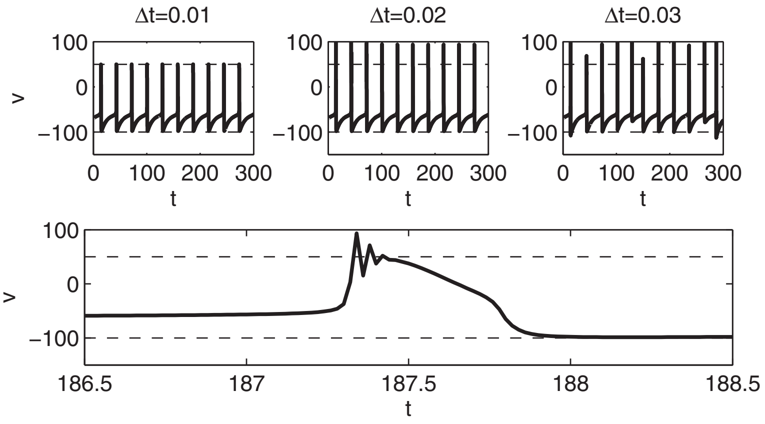

FIG. 6.2. *Voltage traces of the RTM neuron* ($I = 0.7$)*, computed using the explicit Euler method with various values of* Δt*. The bottom panel shows a close-up of the computation with* $\Delta t = 0.02$.

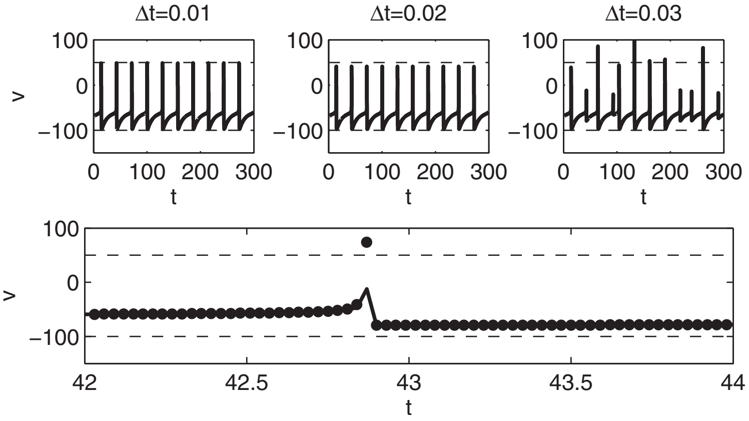

Fig. 6.3. *Voltage traces of the RTM neuron* ($I = 0.7$)*, computed using the midpoint method with various values of* Δt*. The bottom panel is a close-up for* $\Delta t = 0.03$*, showing both the computed values* v_j *of the membrane potential (solid) and* $v_{j+1/2}$ *(dots) as functions of* $t_j = j\Delta t$*.*

The breakup of the midpoint method, as Δt increases, looks different in detail. Figure 6.3 is the analogue of Figure 6.2, computed with the midpoint method. When $\Delta t = 0.03$, the membrane potential overshoots (that is, shoots above v_{Na}) during some of the spikes, but close-up views of those spikes (not shown here) do not reveal any zig-zag behavior near the peak voltage values. Furthermore, there is now what appears to be a new problem, namely, instances of spikes during which the membrane potential rises *too little*, not too much. However, examination of the computed spikes during which the peak membrane potential remains much smaller than v_{Na} shows that the cause of the trouble is that the membrane potential $v_{j+1/2}$, computed in the *preliminary* (half) step of the midpoint method, overshoots. The term $g_K n^4(v_K - v)$ in (2.1) then aborts the rise in v prematurely. This is illustrated by the bottom panel of Figure 6.3, which shows both v_j (solid line) and $v_{j+1/2}$ (dots) as functions of $t_j = j\Delta t$. Thus, even for the midpoint method, the fundamental cause of trouble is overshoot during the rising phase of the action potential.

Voltage spikes of the WB model are considerably smoother, with a longer rising phase; see Figure 6.1, panel B. This explains why larger values of Δt can be used to integrate the WB model equations; compare Figure 4.1.

7. Adaptive time-stepping is of questionable use here.

A natural conclusion from section 5 and section 6 would be that time-stepping ought to be adaptive: Between action potentials, one should use much larger values of Δt than during action potentials. However, in a network of neurons, standard strategies for adapting time steps will refine the time step for the entire network each time any of the neurons in the network spikes, unless one develops a sophisticated strategy using different values of Δt for different neurons. This point is illustrated by Figure 7.1. In Figure 7.1, panel A, we show the result of simulating a single RTM neuron ($I = 0.7$) using `ode23` with `options=odeset('Reltol',0.02)`. Figure 7.1, panel B, shows the time steps Δt chosen by the code, as a function of t. Not surprisingly, Δt varies greatly over the course of a period, from about 0.003 ms during an action potential to about 4.1 ms just prior to an action potential. This variation, of course, is desirable: It reflects efficiency of the adaptive time-stepping strategy. In Figure 7.1, panel C, we show results of a simulation of 500 uncoupled RTM neurons. The drive to the kth neuron is

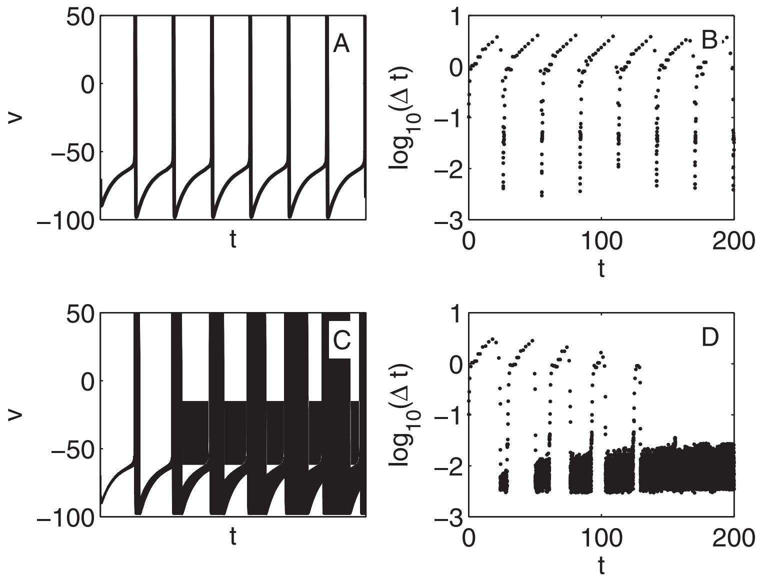

Fig. 7.1. A: *Voltage trace of a spiking RTM neuron.* B: *Time steps chosen by* ode23 *of MATLAB for the simulation in A, plotted on a logarithmic scale. The time step varies by three orders of magnitude.* C: *Voltage traces of* 500 *uncoupled RTM neurons spiking at slightly different frequencies.* D: *Time steps chosen by* ode23 *for the simulation in C, plotted on a logarithmic scale. The time step variation is reduced from three orders of magnitude to one.*

$0.6 + k/2500$, thus the drive varies uniformly from 0.6 to 0.8. The neurons are started in synchrony, but because of the heterogeneity in drives, they desynchronize. As a result, the time step variations become much less pronounced. During the final 40 ms of the simulation, the time step varies only from about 0.003 ms to about 0.028 ms. Thus much of the advantage offered by adaptive time-stepping is erased.

8. Fully implicit time-stepping is not useful here.

A typical approach to overcoming stability issues constraining Δt is to use fully implicit time-stepping. However, fully implicit time-stepping, of course, requires the solution of a nonlinear system of equations in each time step. Simple iterative methods for solving these systems, such as fixed point iteration or Newton's method, require sufficiently small Δt to converge, and the constraint on Δt that appears here as a convergence condition can be just as severe as the one that we were trying to escape by using fully implicit time-stepping to begin with. We illustrate this point first with a numerical experiment for the RTM model, then with analysis for the logistic equation.

8.1. Numerical experiments.

As an example, we apply the implicit Euler method to solve the RTM model equations. To compute v_{j+1}, h_{j+1}, and n_{j+1} from v_j, h_j, and n_j, we have to solve a nonlinear system of equations. We do this using $\nu > 0$ steps of either fixed point iteration or Newton's method, starting with the initial guesses v_j, h_j, and n_j. (It is easy to implement Newton's method using exact, analytically computed derivatives in this example.) In the limit as $\nu \to \infty$, if the iteration converges, the implicit Euler method is obtained. In practice, one might fix a fairly small value of ν, obtaining, in effect, an explicit method that approximates the implicit Euler method if the iteration converges rapidly enough. All of these methods require time step constraints, which are summarized in Table 3. These constraints

Table 3

Largest value of Δt, rounded to one significant digit, for which the approximate implicit Euler method, using ν steps of fixed point iteration or Newton's method per time step, produces voltages bounded by v_K and v_{Na} for the RTM model with $I = 0.7$. In each case, the initial guess for the iteration at a given time step is the approximation at the previous time step.

	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
Fixed point iteration	0.01	0.02	0.01	0.01	0.01
Newton's method	0.004	0.005	0.004	0.004	0.004

are more severe for implicit Euler with fixed point iteration than for explicit Euler, and even more severe for implicit Euler with Newton's method.

8.2. Analysis for the logistic equation. Following Oh and French [22], we consider, as a model problem, the initial-value problem

$$\frac{dx}{dt} = rx(1-x), \qquad x(0) = x_0, \tag{8.1}$$

with $r > 0$, $0 < x_0 < 1$. The equation drives x toward 1 monotonically. If r is large, the ascent towards 1 is rapid, and for explicit schemes, Δt must be small to prevent overshoot. Implicit methods can overcome this constraint, but as soon as one introduces an iterative method for solving the nonlinear algebraic equations arising in each time step, the same time step constraint typically returns. (Of course, in this simple example, the nonlinear algebraic equations are quadratic and can therefore be solved explicitly, but the same is not the case for most nonlinear equations, and we therefore disregard this point here.)

Proposition 8.1. (a) *The equation*

$$x_{j+1} = x_j + r\Delta t\, x_j(1-x_j) \quad (j = 0, 1, 2, \ldots), \tag{8.2}$$

defining the explicit Euler method for (8.1), *assures $x_{j+1} \in [0,1]$ for all $x_j \in [0,1]$ if and only if*

$$\Delta t \le \frac{1}{r}. \tag{8.3}$$

(b) *The equation*

$$x_{j+1} = x_j + r\Delta t\, x_{j+1}(1-x_{j+1}) \quad (j = 0, 1, 2, \ldots), \tag{8.4}$$

defining the implicit Euler method for equation (8.1), *has for any $r > 0$, $x_j \in (0,1)$, and $\Delta t > 0$, two solutions x_{j+1}^- and x_{j+1}^+ with $x_{j+1}^- < 0$ and $x_{j+1}^+ \in (x_j, 1)$. Thus, if we define $x_{j+1} = x_{j+1}^+$, over- and undershoot are prevented without any constraint on Δt.*

(c) *Fixed point iteration for* (8.4) *is locally convergent to x_{j+1}^+ if and only if* (8.3) *holds.*

(d) *Newton's method for* (8.4), *starting with the initial guess x_j, converges to x_{j+1}^+ for any $x_j \in (0,1)$ if and only if* (8.3) *holds.*

Proof. (a) We write $g(x) = x + r\Delta t x(1-x)$, so (8.2) becomes $x_{j+1} = g(x_j)$. Note that $g(0) = 0$, $g(1) = 1$, and the maximum of g is attained at $x^* = (1+r\Delta t)/(2r\Delta t) > 0$. Therefore $x_j \in [0,1]$ guarantees $x_{j+1} \in [0,1]$ if and only if $x^* \ge 1$, which is equivalent to (8.3).

(b) For any $x_j \in (0,1)$, there are two real solutions x_{j+1} of (8.4):

$$x_{j+1}^{\pm} = -\frac{1-r\Delta t}{2r\Delta t} \pm \sqrt{\left(\frac{1-r\Delta t}{2r\Delta t}\right)^2 + \frac{x_j}{r\Delta t}} \ .$$

It is straightforward to verify that $x_{j+1}^- < 0$ and $0 < x_{j+1}^+ < 1$, and therefore, because x_{j+1}^+ is a solution of (8.4), $x_{j+1}^+ > x_j$.

(c) The derivative of $x_j + r\Delta t\, x(1-x)$ with respect to x is $r\Delta t\,(1-2x)$. Thus for x_{j+1}^+ to be a stable fixed point, we need $|r\Delta t(1-2x_{j+1}^+)| \le 1$. This condition holds for all $x_j \in (0,1)$ if and only if (8.3) holds.

(d) We write (8.4) in the form

$$r\Delta t\, x_{j+1}^2 + (1-r\Delta t)x_{j+1} - x_j = 0 \tag{8.5}$$

and consider Newton's method to solve it for x_{j+1}, starting with the initial guess x_j. The left-hand side of (8.5) is a quadratic function in x_{j+1}. Its local minimum occurs at $x^\dagger = -(1-r\Delta t)/(2r\Delta t)$. Note that $x_{j+1}^+ > x^\dagger$ and $x_{j+1}^- < x^\dagger$. This implies that Newton's method, starting with the initial guess x_j, converges to x_{j+1}^+ if and only if $x_j > x^\dagger$. For this to hold for any $x_j \in (0,1)$, we need $0 \ge x^\dagger$, or $0 \ge -(1-r\Delta t)/(2r\Delta t)$, i.e., $\Delta t \le 1/r$, so again the constraint (8.3) has returned. □

9. Stability and accuracy of ETD schemes.

9.1. Numerical results for a single neuron. Figure 9.1, panels A–C, show voltage traces of the RTM neuron computed using the exponential Euler method.

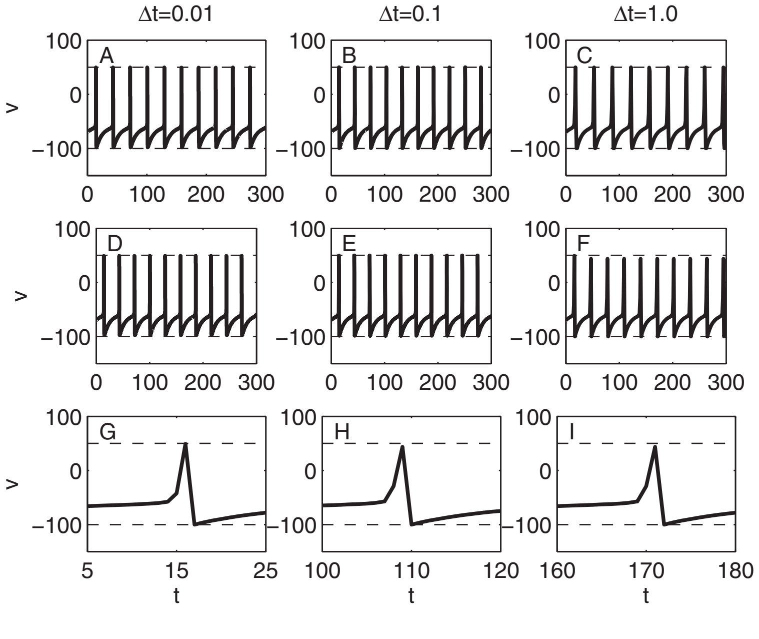

FIG. 9.1. *Voltage traces of the RTM neuron* ($I = 0.7$), *computed using ETD schemes. Note that the values of* Δt *used here are very much larger than those in Figure* 6.2. *Panels A–C: Exponential Euler. Panels D–F: Exponential midpoint method. Panels G–I: Close-ups of panel F, showing three different computed voltage spikes, demonstrating that even for* $\Delta t = 1$, *the computed voltage spikes have a stereotypical shape.*

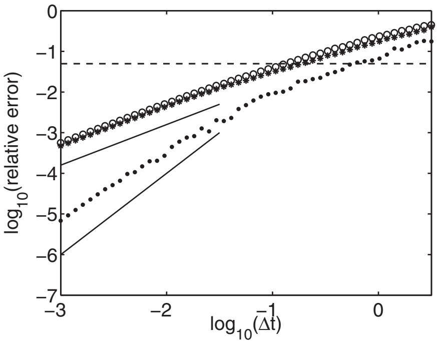

FIG. 9.2. *Log-log plot of relative error in computed frequency for RTM neuron ($I = 0.7$), as a function of Δt, for exponential Euler (stars), midpoint method (dots), and SI Euler (circle). The solid lines are of slopes* 1 *and* 2. *(Since the scaling is not the same on both axes, actual slopes seen in the figures differ from* 1 *and* 2.*) Dashed horizontal line: relative error* $= 0.05$ *(i.e., five percent error).*

These plots should be compared with Figure 6.2, which shows similar results obtained with the explicit Euler method. With the exponential Euler method, spike trains that look qualitatively correct (though with too low a frequency) are obtained even when $\Delta t = 1\,$ms. Figure 9.1, panels D–F, show similar results for the exponential midpoint method. Figure 9.2 shows the percentage error in the computed frequency of an RTM neuron ($I = 0.7$) as a function of Δt, in a log-log plot, for the exponential Euler and midpoint methods, as well as for the SI Euler method (see section 10). This should be compared with Figure 4.2, panel A, where similar results are shown for the explicit Euler and midpoint methods, and for RK4. In Figure 9.2, no instability is visible even for Δt as large as $10^{0.5} \approx 3.2$. Five percent accuracy is obtained with $\Delta t \approx 10^{-0.75} \approx 0.18$ when the exponential Euler method is used, and with $\Delta t \approx 1$ when the exponential midpoint method is used.

It is interesting to look at the shapes of the spikes computed with $\Delta t = 1$ more closely. Figure 9.1, panels G–I, show close-ups of Figure 9.1, panel F. Note that the computed voltage spikes, while much broader than real voltage spikes (compare Figure 6.1, panel A), look alike, and the voltage rises to nearly v_{Na}, and then drops to nearly v_K, just as in the real spikes.

9.2. Bounding box for the ETD schemes. The following discrete analogue of Proposition 3.1 shows that the ETD schemes are stable.

PROPOSITION 9.1. *For both exponential Euler and the exponential midpoint method, if inequalities* (3.1) *hold and* $(v_0, h_0, n_0) \in (v_K, v_{Na}) \times (0, 1) \times (0, 1)$*, then* $(v_j, h_j, n_j) \in (v_K, v_{Na}) \times (0, 1) \times (0, 1)$ *for all* $j \geq 0$.

Proof. We assume

$$(v_j, h_j, n_j) \in (v_K, v_{Na}) \times (0, 1) \times (0, 1) \tag{9.1}$$

and will show

$$(v_{j+1}, h_{j+1}, n_{j+1}) \in (v_K, v_{Na}) \times (0, 1) \times (0, 1). \tag{9.2}$$

This will then imply our assertion by induction.

We first prove (9.2) for the exponential Euler method. Equations (2.5) and (2.6), together with (3.1) and (9.1), imply $d\tilde{v}/dt > 0$ for $\tilde{v} = v_K$, $d\tilde{v}/dt < 0$ for $\tilde{v} = v_{Na}$,

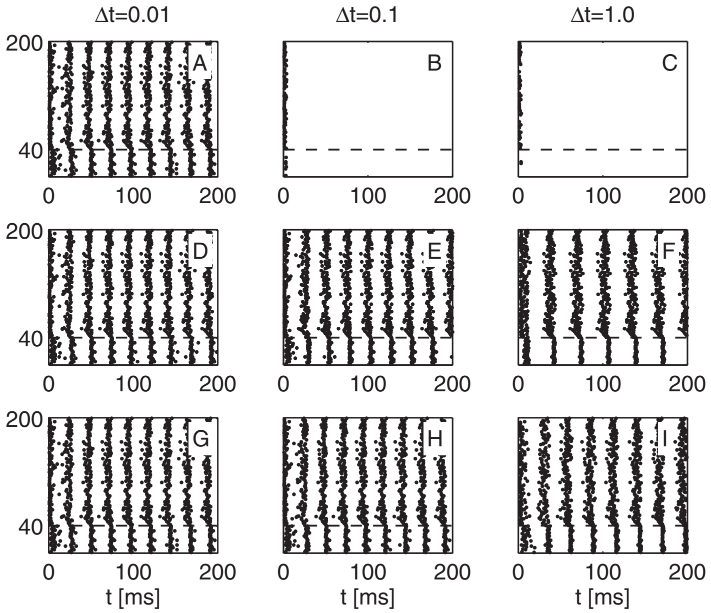

Fig. 9.3. *Results of simulating an E-I network using the midpoint method (A–C), the exponential Euler method (D–F), and the exponential midpoint method (G–H) with various values of* Δt*. The horizontal axis denotes time in milliseconds, and the vertical axis denotes neuronal index. Cells* 1–40 *(below the dashed line) are I-cells, and cells* 41–200 *(above the dashed line) are E-cells.*

$d\tilde{h}/dt > 0$ for $\tilde{h} = 0$, $d\tilde{h}/dt < 0$ for $\tilde{h} = 1$, $d\tilde{n}/dt > 0$ for $\tilde{n} = 0$, and $d\tilde{n}/dt < 0$ for $\tilde{n} = 1$. Therefore $(\tilde{v}, \tilde{h}, \tilde{n}) \in (v_K, v_{Na}) \times (0,1) \times (0,1)$ for all $t \geq t_j$, and therefore (9.2) holds.

Now we prove (9.2) for the exponential midpoint method. Since the preliminary half step is carried out with the exponential Euler method, what we have already shown implies that $(v_{j+1/2}, h_{j+1/2}, n_{j+1/2}) \in (v_K, v_{Na}) \times (0,1) \times (0,1)$. But then (2.7) and (2.8) imply again that $d\tilde{v}/dt > 0$ for $\tilde{v} = v_K$, $d\tilde{v}/dt < 0$ for $\tilde{v} = v_{Na}$, $d\tilde{h}/dt > 0$ for $\tilde{h} = 0$, $d\tilde{h}/dt < 0$ for $\tilde{h} = 1$, $d\tilde{n}/dt > 0$ for $\tilde{n} = 0$, and $d\tilde{n}/dt < 0$ for $\tilde{n} = 1$, and therefore $(\tilde{v}, \tilde{h}, \tilde{n}) \in (v_K, v_{Na}) \times (0,1) \times (0,1)$ for all $t \geq t_j$, and therefore (9.2). □

9.3. Numerical results for networks. We now consider a network of E- and I-cells as described in section 2.2. Panels A–C of Figure 9.3 show results of simulating the network using the (standard) midpoint method with $\Delta t = 0.01$, 0.1, and 1.0. For $\Delta t = 0.1$ and 1.0, there is a catastrophic instability, resulting in an overflow error. Panels D–F and G–I of the figure show similar simulations using the exponential Euler and exponential midpoint methods, respectively. In both cases, even the results with $\Delta t = 1\,\text{ms}$ are qualitatively reasonable, although the oscillation frequency is too low, especially for the exponential Euler method (panel F of Figure 9.3).

The exponential methods do require somewhat more work per time step than the standard methods. However, this extra cost is more than compensated for by the ability to use larger values of Δt with the exponential methods. Table 4 shows some timing results, obtained using a MacBook Pro (3.06 GHz Intel Core 2 Duo). We also indicate in the table the frequency, in Hz, of the first I-cell, estimated as $1000/\overline{T}$, where $\overline{T}$ denotes the mean interspike interval of the first I-cell. (Since each cell fires once

Table 4

Simulation times in seconds on a MacBook Pro (3.06 GHz Intel Core 2 Duo) for the network of Figure 9.3. Asterisks indicate overflow errors. Estimated network oscillation frequencies, rounded to the nearest Hz, are given to indicate, approximately, the accuracy of the simulations.

	0.005	0.01	0.02	0.05	0.1	0.5	1.0
Euler	13.0 s 42 Hz	6.45 s 42 Hz	3.24 s 42 Hz	* * *	* * *	* * *	* * *
Exp. Euler	20.0 s 43 Hz	9.01 s 42 Hz	4.44 s 42 Hz	1.96 s 42 Hz	0.920 s 41 Hz	0.203 s 35 Hz	0.119 s 31 Hz
Midpoint	25.3 s 43 Hz	13.1 s 43 Hz	6.56 s 43 Hz	* * *	* * *	* * *	* * *
Exp. midpoint	35.8 s 43 Hz	17.6 s 43 Hz	8.76 s 43 Hz	3.40 s 43 Hz	1.74 s 43 Hz	0.545 s 39 Hz	0.198 s 38 Hz
RK4	52.7 s 43 Hz	28.8 s 43 Hz	12.9 s 43 Hz	* * *	* * *	* * *	* * *

per oscillation period, this is also an estimate of the population oscillation frequency.) The results indicate that exponential methods can easily yield speed-ups by an order of magnitude, albeit at the expense of some loss in accuracy.

It may seem pointless to accelerate a calculation that takes only a few seconds on a laptop to begin with. However, if one wants to simulate much larger networks for much longer times, or explore high-dimensional parameter spaces—typical situations in computational neuroscience—acceleration by an order of magnitude becomes significant.

10. Stability and accuracy of the SI Euler method. The performance of the SI Euler method is very similar to that of the exponential Euler method: The computed voltage traces (not shown here) are qualitatively correct, albeit with too low a firing frequency, even when $\Delta t = 1$ ms. The circles in Figure 9.2 show the percentage error in the frequency of an RTM neuron ($I = 0.7$) computed using the SI Euler method as a function of Δt, demonstrating that the SI Euler method is just as stable as the exponential Euler method, and of very similar accuracy. Results for E/I networks are very similar to those in Figure 9.3, panels D–F, and are therefore not shown here. For SI Euler, the analogue of Proposition 9.1 is true and can be derived analogously.

11. Further analysis for a model equation. During the ascending phase of the voltage spike, a rise in v causes the opening of sodium channels in the cell membrane (that is, a rise in the gating variable m), which in turn accelerates the rise in v. We consider here a model equation that is a simple caricature of this mechanism and discuss its numerical solution. This will help explain our findings in earlier sections. Our model equation is

$$\frac{dv}{dt} = g(v)(1-v), \quad v(0) = 0, \tag{11.1}$$

where $g = g(v) > 0$ is a differentiable, increasing function of $v \geq 0$. With $g(v) = rv$ (disregarding the fact that, strictly speaking, this does not satisfy our assumptions because $g(0) = 0$), we get the logistic equation, which was considered as a model equation in section 8.2. We assume that $\lim_{v \to 1} g(v)$ is finite and denote it by $1/\tau$, where $\tau > 0$. Here τ is the time constant characterizing the convergence of v to 1. We think of τ as the analogue of the duration of the rising phase of the action potential.

For (11.1), we define the exponential Euler method by

$$\frac{d\tilde{v}}{dt} = g(v_j)(1-\tilde{v}), \qquad \tilde{v}(t_j) = v_j, \qquad v_{j+1} = \tilde{v}(t_{j+1}).$$

The exponential midpoint method is defined by

$$\frac{d\tilde{v}}{dt} = g(v_{j+1/2})(1-\tilde{v}), \qquad \tilde{v}(t_j) = v_j, \qquad v_{j+1} = \tilde{v}(t_{j+1}),$$

where $v_{j+1/2}$ is computed using a step of the exponential Euler method with step size $\Delta t/2$. The SI Euler method is defined by

$$\frac{v_{j+1}-v_j}{\Delta t} = g(v_j)(1-v_{j+1}).$$

PROPOSITION 11.1. (a) *The solution of* (11.1) *is strictly increasing and converges to* 1 *as* $t \to \infty$.

(b) *For stability, the explicit Euler and midpoint methods for* (11.1) *require* $\Delta t \leq 2\tau$, *and RK*4 *requires* $\Delta t \leq \alpha\tau$, $\alpha \approx 2.785$.

(c) *The exponential Euler, exponential midpoint, and SI Euler methods generate strictly increasing sequences* v_0, v_1, $v_2, \ldots$ *with* $\lim_{j\to\infty} v_j = 1$ *for all* $\Delta t > 0$.

Proof. (a) This follows from $g(v)(1-v) > 0$ for $v < 1$. (b) These are standard stability conditions for these three methods. (c) It is easy to verify that

$$v_{j+1} = \gamma_j v_j + 1 - \gamma_j \tag{11.2}$$

with $\gamma_j = e^{-g(v_j)\Delta t}$ for the exponential Euler method, $\gamma_j = e^{-g(v_{j+1/2})\Delta t}$ for the exponential midpoint method, and $\gamma_j = (1+g(v_j)\Delta t)^{-1}$ for the SI Euler method. In each case, $0 < \gamma_j < 1$, and therefore $\{v_j\}$ is a strictly increasing sequence bounded from above by 1. Also, in each case there is an upper bound on γ_j that is independent of j and less than 1, obtained by replacing $g(v_j)$ or $g(v_{j+1/2})$ by $g(0) > 0$ in the definition of γ_j. This implies $\lim_{j\to\infty} v_j = 1$. □

The most straightforward second-order SI midpoint method would be

$$v_{j+1/2} = v_j + \frac{\Delta t}{2}\, g(v_j)(1-v_{j+1/2}), \tag{11.3}$$

$$v_{j+1} = v_j + \Delta t\, g(v_{j+1/2})\left(1 - \frac{v_j + v_{j+1}}{2}\right). \tag{11.4}$$

Equation (11.4) can be written in the form (11.2) with

$$\gamma_j = \frac{1 - g(v_{j+1/2})\,\Delta t/2}{1 + g(v_{j+1/2})\,\Delta t/2}.$$

To ensure that this number does not become negative, we need $\Delta t \leq 2\tau$. Thus a time step constraint similar to those for the fully explicit methods has returned here.

12. Summary and discussion.

The work presented here provides an understanding of why exponential time differencing is a good idea not only for Hodgkin–Huxley-like PDEs, but even for the model equations for "space-clamped" neurons, the Hodgkin–Huxley-like ODEs. We have demonstrated that for Hodgkin–Huxley-like ODEs, standard explicit time-stepping methods, such as Euler's method, the midpoint method, or RK4, require very small time steps, often on the order of a hundredth of a millisecond, because of the rising phase of the action potential, which often lasts only a few hundredths of a millisecond. When one uses larger time steps in these methods, there is overshoot during the rising phase of the action potential, triggering instability. By contrast, the exponential Euler method, the exponential midpoint method in the form proposed here, and the SI Euler method allow arbitrarily large time steps. With $\Delta t \approx 1\,\text{ms}$, computed voltage spikes are, of course, broader than the real ones, but the solutions are otherwise qualitatively similar to the correct solutions. Thus the exponential methods, in particular the exponential midpoint method, seem useful for the preliminary exploration of large parameter spaces or large networks.

Exponential time differencing is a large field of current research [5, 6, 10, 11, 12, 15, 18, 20, 21, 32]. Most of the work on ETD has focused on PDEs of the form

$$\frac{du}{dt} = Lu + N(u,t) \tag{12.1}$$

with L linear (often a second-order elliptic partial differential operator) and N nonlinear. This is a natural decomposition of the right-hand side reaction-diffusion problems, such as Hodgkin–Huxley-like PDEs. However, here we consider Hodgkin–Huxley-like ODEs, which cannot naturally be written in the form (12.1). We demonstrate that, and explain why, it is a good idea to use ETD even for the Hodgkin–Huxley ODEs, exploiting the fact that in Hodgkin–Huxley-like systems, each dependent variable appears linearly in the equation governing its time evolution. For the Hodgkin–Huxley equations with space dependence, we believe that one would want to combine an ETD method designed for reaction-diffusion problems with ideas of the sort discussed here in order to overcome both the diffusive and the reactive time step constraints, and we plan to make this the subject of future work.

The idea of using this kind of exponential method for neuronal simulations is not ours. In fact, the exponential Euler method is the default time-stepping method in the software packages CSIM [19] and GENESIS [2]. Its accuracy in the limit as $\Delta t \to 0$ was analyzed by Oh and French [22]. We have made a small modification to the second-order method of Oh and French [22]: We use the exponential Euler method for the preliminary half step, whereas Oh and French used the explicit Euler method [22, equation (4)], which reintroduces stability issues during the ascending phase of the action potential. For instance, for the RTM neuron, the second-order method of Oh and French allows $\Delta t = 0.5\,\text{ms}$ but becomes unstable for $\Delta t = 0.8\,\text{ms}$. Although the method of Oh and French is not unconditionally stable for Hodgkin–Huxley-like systems, it can easily be seen to be unconditionally stable for the model problem of section 11. In fact, Oh and French presented numerical results for their method applied to the logistic equation [22, Figure 1], a special case of the model equation in section 11.

An alternative approach to performing neuronal network simulations without a need for extremely small time steps, likely more accurate but also very much more complicated than ETD with large Δt, has been proposed by Sun, Zhou, and Cai [26]. In their method, voltage spike shapes are precomputed and then inserted when needed during the simulation. Stewart and Bair [25] applied a Picard-iteration algorithm to the RTM neuron. (Their model equations, taken from [3], differ slightly from ours.) They found that the scheme still has a stability threshold, although one that is significantly less stringent than that of RK4 [25, p. 128].

It is often acceptable not to resolve the spike shape in detail. The widely used integrate-and-fire model does not even model spike shapes at all. There are, however, situations in which detailed spike shapes do matter. One example that we are aware of is that of gap-junctionally (electrically) coupled neurons. Spike shapes, and in particular spike widths, contribute to determining whether gap junctions are synchronizing, which is the usual situation [17, 29], or antisynchronizing, which is at least a mathematical possibility [4].

Each figure in this paper was generated by a stand-alone MATLAB code, all of which is available from the first author upon request.

Acknowledgments. We thank Shane Lee for pointing out that in the network simulations of [16] (midpoint method, $\Delta t = 0.02\,\mathrm{ms}$), there are subtle indications of near-instability. This observation motivated the work described here.

REFERENCES

[1] C. Börgers, S. Epstein, and N. Kopell, *Background gamma rhythmicity and attention in cortical local circuits: A computational study*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 7002–7007.

[2] J. M. Bower and D. Beeman, *The Book of GENESIS*, Springer-Verlag, New York, 1998.

[3] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, M. Zirpe, T. Natschläger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. P. Davison, S. E. Boustani, and A. Destexhe, *Simulation of networks of spiking neurons: A review of tools and strategies*, J. Comput. Neurosci., 23 (2007), pp. 349–398.

[4] C. C. Chow and N. Kopell, *Dynamics of spiking neurons with electrical coupling*, Neural Comput., 12 (2000), pp. 1643–1678.

[5] S. M. Cox and P. C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys., 176 (2002), pp. 430–455.

[6] F. de la Hoz and F. Vadillo, *An exponential time differencing method for the nonlinear Schrödinger equation*, Comput. Phys. Comm., 179 (2008), pp. 449–456.

[7] G. B. Ermentrout and N. Kopell, *Fine structure of neural spiking and synchronization in the presence of conduction delay*, Proc. Natl. Acad. Sci. USA, 95 (1998), pp. 1259–1264.

[8] D. R. Hartree, *A practical method for the numerical solution of differential equations*, Mem. Manch. Lit. Phil. Soc., 77 (1933), pp. 91–107.

[9] H. Hasegawa, *Responses of a Hodgkin-Huxley neuron to various types of spike-train inputs*, Phs. Rev. E, 61 (2000), pp. 718–726.

[10] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.

[11] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286.

[12] M. Hochbruck and A. Ostermann, *Exponential multistep methods of Adams-type*, BIT, 51 (2011), pp. 889–908.

[13] A. L. Hodgkin and A. F. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, J. Physiol. (London), 117 (1952), pp. 500–544.

[14] E. C. Y. Ho, M. Strüber, M. Bartos, L. Zhang, and F. K. Skinner, *Inhibitory networks of fast-spiking interneurons generate slow population activities due to excitatory fluctuations and network multistability*, J. Neurosci., 32 (2012), pp. 9931–9946.

[15] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.

[16] N. Kopell, C. Börgers, D. Pervouchine, P. Malerba, and A. B. L. Tort, *Gamma and theta rhythms in biophysical models of hippocampal circuits*, in Hippocampal Microcircuits: A Computational Modeler's Resource Book, V. Cutsuridis, B. Graham, S. Cobb, and I. Vida, eds., Springer-Verlag, New York, 2010.

[17] N. Kopell and B. Ermentrout, *Chemical and electrical synapses perform complementary roles in the synchronization of interneuronal networks*, Proc. Natl. Acad. Sci. USA, 101 (2004), pp. 15482–15487.

[18] S. Maset and M. Zennaro, *Unconditional stability of explicit exponential Runge-Kutta methods for semi-linear ordinary differential equations*, Math. Comp., 78 (2009), pp. 957–967.

[19] T. Natschläger, *CSIM: A Neurel Circuit SIMulator*, http://www.lsm.tugraz.at/csim/index.html (19 April 2008).

[20] Q. Nie, F. Y. M. Wan, Y.-T. Zhang, and X.-F. Liu, *Compact integration factor methods in high spatial dimensions*, J. Comput. Phys., 227 (2008), pp. 5238–5255.

[21] Q. Nie, Y.-T. Zhang, and R. Zhao, *Efficient semi-implicit schemes for stiff systems*, J. Comput. Phys., 214 (2006), pp. 521–537.

[22] J. Oh and D. A. French, *Error analysis of a specialized numerical method for mathematical models from neuroscience*, Appl. Math. Comput., 172 (2006), pp. 491–507.

[23] M. Olufsen, M. Whittington, M. Camperi, and N. Kopell, *New functions for the gamma rhythm: Population tuning and preprocessing for the beta rhythm*, J. Comput. Neurosci., 14 (2003), pp. 33–54.

[24] J. Rubin and M. Wechselberger, *The selection of mixed-mode oscillations in a Hodgkin-Huxley model with multiple timescales*, Chaos, 18 (2008), 015105.

[25] R. D. Stewart and W. Bair, *Spiking neural network simulation: Numerical integration with the Parker-Sochacki method*, J. Comput. Neurosci., 27 (2009), pp. 115–133.

[26] Y. Sun, D. Zhou, and A. V. R. D. Cai, *Library-based numerical reduction of the Hodgkin-Huxley neuron for network simulation*, J. Comput. Neurosci., 27 (2009), pp. 369–390.

[27] P. H. Tiesinga, J. V. José, and T. J. Sejnowski, *Comparison of current-driven and conductance-driven neocortical model neurons with Hodgkin-Huxley voltage-gated channels*, Phys. Rev. E, 62 (2000), pp. 8413–8419.

[28] R. D. Traub, D. Contreras, M. O. Cunningham, H. Murray, F. E. N. LeBeau, A. Roopun, A. Bibbig, W. B. Wilent, M. J. Higley, and M. A. Whittington, *Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts*, J. Neurophysiol., 93 (2005), pp. 2194–2232.

[29] R. D. Traub, N. Kopell, A. Bibbig, E. H. Buhl, F. E. N. Lebeau, and M. A. Whittington, *Gap junctions between interneuron dendrites can enhance long-range synchrony of gamma oscillations*, J. Neurosci., 21 (2001), pp. 9478–9486.

[30] R. D. Traub and R. Miles, *Neuronal Networks of the Hippocampus*, Cambridge University Press, Cambridge, UK, 1991.

[31] X.-J. Wang and G. Buzsáki, *Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network model*, J. Neurosci., 16 (1996), pp. 6402–6413.

[32] Y. Xia, Y. Xu, and C.-W. Shu, *Efficient time discretization for local discontinuous Galerkin methods*, Discrete Contin. Dyn. Syst. Ser. B, 8 (2007), pp. 677–693.