Time Stepping for the cable equation, Part 1: Serial performance

Chuan Li[1] **and Vasilios Alexiades**[2]

[1]Mathematics Department, University of Tennessee, Knoxville TN 37996

[2]Mathematics Department, University of Tennessee, Knoxville TN 37996
and Oak Ridge National Laboratory, Oak Ridge TN

Abstract

Electrical propagation (action potentials) in excitable tissue, such as nerve cells and cardiac myocytes, is described by a parabolic diffusion-reaction equation for the transmembrane potential (voltage) $V(x, t)$, known as the *cable equation.* It is driven by a rapidly varying, highly nonlinear (and expensive to evaluate) ionic source term $I_{ion}(V, t)$ representing the total ionic current across the cell membrane, plus a stimulus current. The ionic term is governed by the Hodgkin-Huxley or some other more complicated ionic model, appropriate to the tissue; its evaluation requires solving a system of ODEs for the "gates".

We compare the performance of eleven time-stepping numerical schemes on the 1D cable equation with Luo-Rudy I (1991) ionic source (the evaluation of which involves seven ODEs). The time-steppers include Euler, Super-Time-Stepping, DuFort-Frankel, as well as low and high order, explicit and implicit, non-adaptive and adaptive Runge-Kutta integrators.

Keywords: Explicit schemes, Super-Time-Stepping, Dufort-Frankel, adaptive Runge-Kutta, cardiac action potential, Luo-Rudy ionic model.

1 THE CABLE EQUATION

The parabolic, diffusion-reaction type PDE

$$\frac{1}{R_a}\frac{\partial^2 V}{\partial x^2} = C_m \frac{\partial V}{\partial t} + I_{ion}(V, t) + I_{stim}(t), \tag{1}$$

is known as the *cable equation* (Keener & Sneyd, 1998; Plonsey & Barr, 2007). It describes electrical propagation in excitable tissue, such as nerve fibers and cardiac myocytes. In (1), $V(x, t)$ is the transmembrane potential (voltage), R_a and C_m are the axial resistance and membrane capacitance, I_{ion} represents the total ionic current, and $I_{stim}(t)$ is an applied stimulus current which instigates an action potential.

The first successful ionic current model was introduced in (Hodgkin & Huxley, 1952). In 1991, Luo and Rudy published a model appopriate for cardiac myocytes, which consists of several ionic currents generated by sodium, potassium and calcium ions (Luo & Rudy,1991)

$$I_{ion}(V, t) = I_{Na}(V) + I_{SI}(V) + I_{K1(T)}(V). \tag{2}$$

These currents depend on seven activation and inactivation "gates": $m,\ h,\ j,\ d,\ f,\ X,\ Cai$, each of which is governed by an ODE of the form

$$\frac{dg}{dt} = \alpha_g(V)(1-g) - \beta_g(V)g, \quad g = m,\ h,\ j,\ d,\ f,\ X,\ Cai. \tag{3}$$

The α's and β's, taking values between 0 and 1, are specified by (messy) explicit formulas as functions of voltage V. We used the curated version *luo_rudy_1991_version06* from cellML (http://models.cellml.org/).

The stimulus current $I_{stim}(t)$ is the pacemaker that excites the system and instigates an action potential. In the heart, the stimulus is supplied by the Sino-Atrial Node. To generate a *single* action potential, we apply a *single* stimulus, of duration 1 ms and strength $-200\ \mu A/cm^2$, on a short segment $[0, 10\mu m]$ at one end of the cable.

We use the parameter values: $C_m = 1.2\ \mu F/cm^2$, $R_a = 300\ k\Omega$. This value of R_a is three orders of magnitude greater than the cytoplasmic resistance for human myocytes (Keener & Sneyd, 1998), which is a very low resistance, and applies only within a single cell, whereas we are modeling chains of cells. Gap junctions between cells present much greater resistance to current flow. The larger value used here speeds up computations a thousand-fold, so they can be carried out within more reasonable time (and they are still very long!).

We assume zero voltage gradient at the ends of the cable, and initialize the system from steady state with initial values: $V_{init} = -84.547997\,mV$, $m_{init} = 0.001665$, $h_{init} = 0.983302$, $j_{init} = 0.989522$, $d_{init} = 0.002977$, $f_{init} = 0.999981$, $X_{init} = 0.005643$ and $Cai_{init} = 0.000178$.

The mathematical model consists of the PDE (1), the seven ODEs (3) and the above initial setup.

A typical calculated action potential is shown in Fig.1. Biologically significant quantities computed in the simulations are:

- **Action Potential Duration** (APD) = how long the potential V at a fixed location stays above a certain cut-off value. We set V_{cutoff} as 90% of the initial equilibrium voltage.

- **Conduction Velocity** = speed of propagation of action potential = the difference of starting time of APs at two fixed nodes.

- **Maximum voltage** (V_{max}) and **maximum rate of change of voltage** ($\{dV/dt\}_{max}$) at the nodes (excluding nodes directly stimulated by I_{stim}).

These quantities are characteristic of the cell and I_{ion}, and independent of I_{stim}, cable length, and nodes used for measurement. Thus they also serve as accuracy indicators on the numerical schemes.

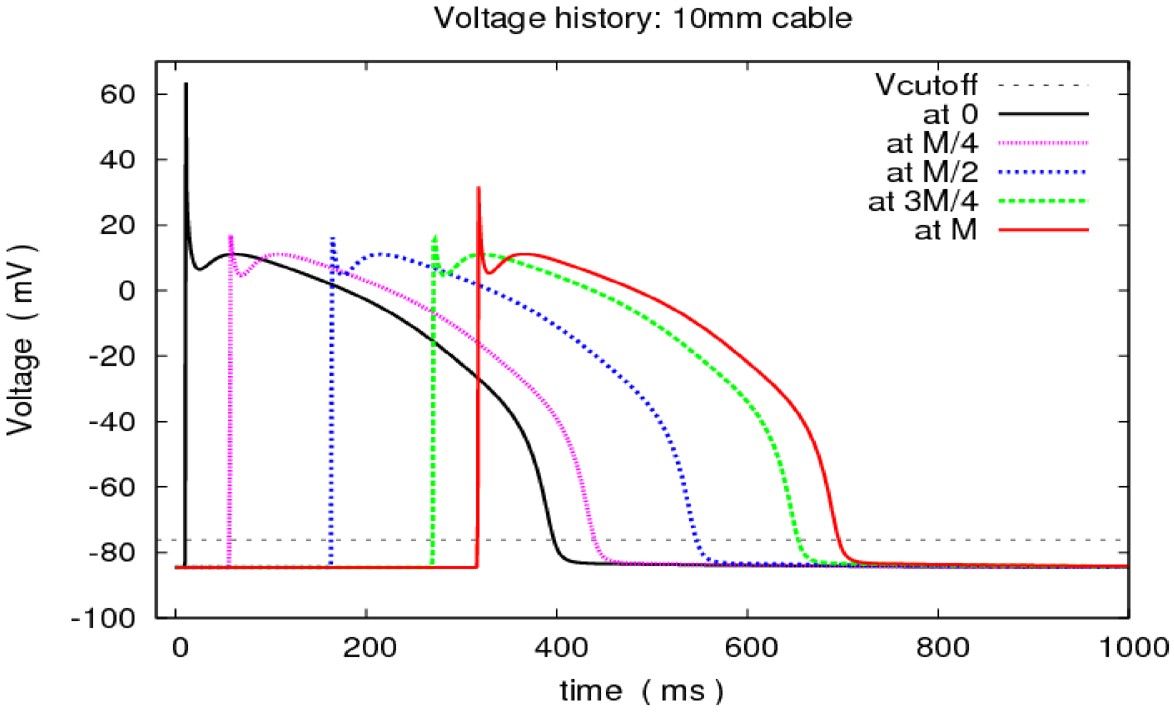

Figure 1: Single action potential propagating along a 10 mm cable. The cable is discretized into $M = 10mm/4\mu m = 2500$ nodes. Voltage histories at five nodes are shown.

2 NUMERICAL SCHEMES

We discretize the cable into M control volumes of uniform length Δx. A cable contains many cells, and each cell contains several control volumes. By standard Finite Volume discretization of the PDE (1), and applying the 7 ODEs (3) on each control volume yields a system of $8 \times M$ ODEs,

$$\frac{dV_k}{dt} = \frac{1}{C_m}\left[\frac{F_{k-\frac{1}{2}} - F_{k+\frac{1}{2}}}{\Delta x} - I_{ion}(V_k, t_n) - I_{stim}(t_n)\right], \tag{4}$$

$$\frac{dg_k}{dt} = \alpha_{g_k}(V_k)(1-g_k) - \beta_{g_k}(V_k)g_k, k = 1, ..., M, \tag{5}$$

where V_k is the voltage, $g_k = m_k, h_k, j_k, d_k, f_k, X_k, Cai_k$ are the corresponding values of gates in the k^{th} control volume, and $F_{k-\frac{1}{2}}$ are the diffusion fluxes,

$$F_{\frac{1}{2}} = 0, \qquad F_{k-\frac{1}{2}} = -\frac{1}{R_{a_{k-\frac{1}{2}}}}\frac{V_k - V_{k-1}}{\Delta x}, \quad k = 2, ..., M, \qquad F_{M+\frac{1}{2}} = 0. \tag{6}$$

We apply the following time-stepping schemes to solve the ODE system (4-5).

2.1 Super-Time-Stepping (STS) Scheme

Super time-stepping is a simple-to-implement method to accelerate existing explicit schemes for parabolic problems (Alexiades et al.,1996). One superstep ΔT consists of N substeps $\Delta\tau_1$,...,$\Delta\tau_N$, with optimal substeps $\Delta\tau_j$ given explicitly by

$$\Delta\tau_j = \Delta t_{expl}\left[(-1+\nu)\cos\left(\frac{2j-1}{N}\frac{\pi}{2}\right) + 1 + \nu\right]^{-1} \qquad j = 1, ..., N, \tag{7}$$

where Δt_{expl} is the time step satisfying the CFL stability condition for the explicit scheme. Thus, we choose an integer N and a small damping parameter $\nu > 0$, and instead of executing N uniform steps Δt_{expl} we execute N Chebyshev steps $\Delta\tau_1$,...,$\Delta\tau_N$. It turns out that $\Delta T \to N^2 \Delta t_{\text{expl}} \quad as \quad \nu \to 0$. Thus, executing a superstep consisting of N substeps covers a time interval N times longer than N explicit steps Δt_{expl} (when $\nu \approx 0$). Thus, superstepping is (up to) N times faster than the standard explicit scheme. Note that the method ensures stability only at the end of each superstep. Only values at the end of a superstep should be printed out.

In the simulations we used $N = 4$, $\nu = 0.1$, denoted as STS_4 in the plots. STS with $N = 1$, $\nu = 0$ reduces to forward Euler, denoted Eu in the plots.

2.2 DuFort-Frankel (DF) Scheme

The DuFort-Frankel scheme is explicit, 2-step, 2nd order in space and time, and theoretically unconditionally stable (Mayers & Morton,1994). It can be obtained by applying Forward Euler to (4) and then replacing V_k^n by the centered time average $(V_k^{n+1} + V_k^{n-1})/2$.

To avoid small oscillations near the steady state, and keep the scheme explicit, the average of voltage at two previous time steps is used to evaluate the ionic current $I_{ion}(V, t)$,

$$\begin{aligned} V_k^{n+1} \quad = \quad & \frac{1}{1 + \frac{\Delta t}{C_m \Delta x^2}\left(\frac{1}{R_{a_{k-\frac{1}{2}}}} + \frac{1}{R_{a_{k+\frac{1}{2}}}}\right)}\left[\frac{2\Delta t}{C_m \Delta x^2}\left(\frac{1}{R_{a_{k-\frac{1}{2}}}}V_{k-1}^n + \frac{1}{R_{a_{k+\frac{1}{2}}}}V_{k+1}^n\right)\right. \\ & \left. + \left(1 - \frac{\Delta t}{C_m \Delta x^2}(\frac{1}{R_{a_{k-\frac{1}{2}}}} + \frac{1}{R_{a_{k+\frac{1}{2}}}})V_k^{n-1}\right) - \frac{2\Delta t}{C_m}\left(I_{ion}(\frac{V_k^n + V_k^{n-1}}{2}, t_n) + I_{stim}(t_n)\right)\right]. \end{aligned} \tag{8}$$

On the other hand, the ODEs (5) for the gates are discretized by forward Euler, and again evaluated at the average of the two previous voltage values,

$$g_k^{n+1} = g_k^n + \Delta t\left[\alpha_{g_k^n}\left(\frac{V_k^n + V_k^{n-1}}{2}\right)(1 - g_k^n) - \beta_{g_k^n}\left(\frac{V_k^n + V_k^{n-1}}{2}\right)g_k^n\right]. \tag{9}$$

2.3 RKSUITE Runge-Kutta Schemes

The RKSUITE package (Brankin et al. 1991), available from Netlib, is a suite of explicit Runge-Kutta methods for first order ODE systems. It provides three adaptive methods, RK23, RK45 and RK78, of orders 2, 4, and 7, respectively. We refer to (Li & Alexiades, 2010a) for their performance. Here we employ only **RK23**, the most efficient of the three. The adaptive time step Δt is controlled by two user-provided parameters: relative tolerance *tol* and threshold *thres*, which we set as $tol = 10^{-3}$ and $thres = 10^{-5}$.

2.4 GSL Runge-Kutta Schemes

The GNU Scientific Library (GSL 1.10) from GNU provides several explicit and implicit, non-adaptive and adaptive, low and high order Runge-Kutta schemes for first order systems of ODEs. Of these, we employ three non-adaptive (fixed time-step) integrators: **rk4, rk2imp, rk4imp**, and four adaptive integrators: **rk2, rkck, rkf45, rk8pd**. Their names suggest their order. They are all explicit except **rk2imp, rk4imp**. **rk4** is the classical 4^{th} order RK, and **rkf45** the embedded Runge-Kutta-Fehlberg(4,5). **rkck** is the embedded Runge-Kutta Cash-Karp(4,5) method, and **rk8pd** the embedded Runge-Kutta Prince-Dormand(8,9) method.

3 NUMERICAL SIMULATIONS

Our code is written in C, and the simulations were performed on AMD Opteron 2378 2.4 GHz processors (single cores of a multicore, multinode linux cluster), with Intel C 11.1 compiler. All schemes were run with $\Delta x = 4\ \mu m$ on cables of various lengths. Timings (in *minutes* of CPU time) are shown in Fig.2 for 10 mm and Fig.3 for 50 mm cables.

It turns out that evaluating the ionic currents is very expensive and, worse, it requires time-steps **no larger than** $0.01ms$. This penalizes all schemes, especially implicit and high order ones. In an attempt to reduce run times, we pre-compute all the $\alpha(V)$, $\beta(V)$ coefficients in the range $[-100, 200]$ with $\Delta V = 0.0001$, and store them in a direct access, binary file (367 MB size), which is loaded into memory at run time. Then, values of $\alpha(V_k^n)$, $\beta(V_k^n)$ at any V_k^n are found by interpolation. This approach is known as the **library method** (Sun et al., 2009). Fig.2 shows that it has high payoff, reducing CPU time to almost half with no loss of accuracy (same propagation speed, V_{max}, $\{dV/dt\}_{max}$, and almost same APD). All other computations discussed here use the pre-computed library.

All non-adaptive schemes used $\Delta t = 0.01\ ms$. Comparison with adaptive solvers is shown in Fig.3 on 50 mm cable.

Our numerical experiments so far lead to the following conclusions about the eleven solvers we tested:

- All the high order schemes produce identical voltage history (action potentials, as in Fig.1), and identical values for the biological quantities. Among them, the explicit adaptive 4th order solver **rkck** is the most efficient, and the implicit non-adaptive 4th order solver **rk4imp** is by far the worst, with no redeeming features.

- The low order schemes (STS, Eu, DF) are much faster than the high order ones, by factors of 10 to 25! and $\mathbf{STS}_4$ (i.e. STS with $N = 4$, $\nu = 0.1$) is the most efficient of all. However, they produce upstrokes somewhat delayed (by 20 - 50 ms at the end of 50 mm cable), and slightly lower propagation speed. When such high accuracy is important, high order solvers should be used.

- Among high order solvers, the adaptive ones outperform the non-adaptive by a factor of 2 or more, and **rkck** is best among them (Fig.3). When adaptivity cannot be used, as is the case for parallel computations (Li & Alexiades, 2010b), then **rk4** would be best among high order solvers.

- In view of the fact that evaluation of the source restricts the time-step to $\Delta t \leq 0.01$, the implicit 2nd order solver **rk2imp** performs surprisingly well, being competitive with **rk4**, whereas rk4imp is hopelessly slow.

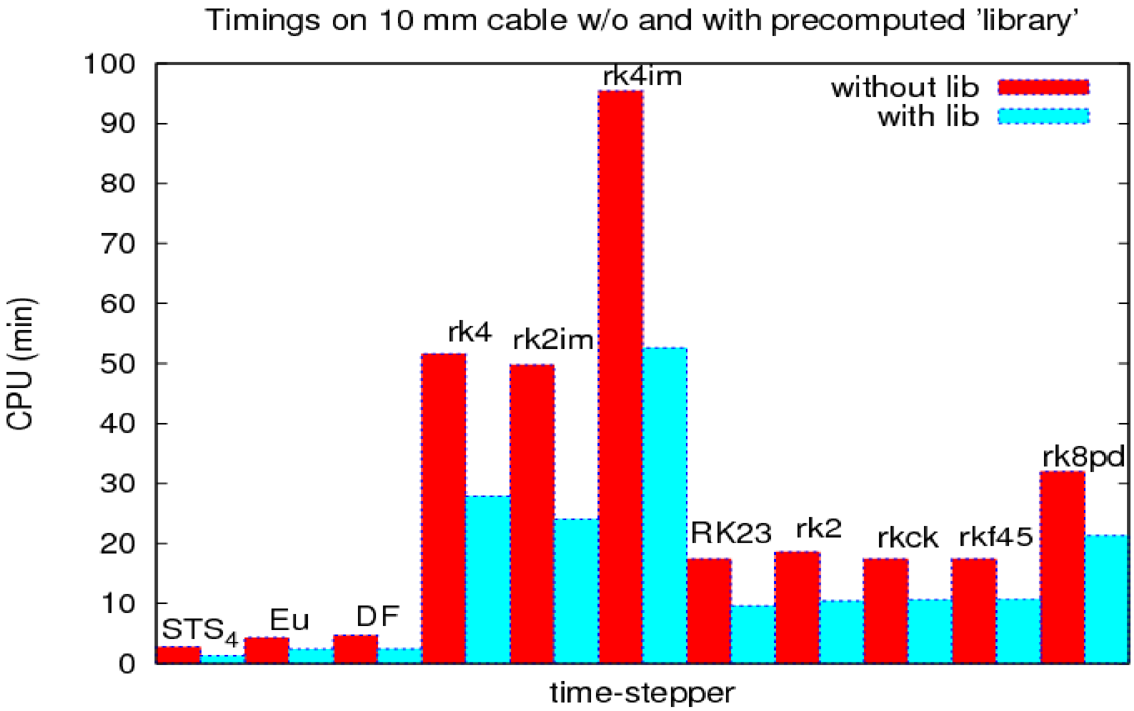

Figure 2: Timings of all eleven schemes on 10mm cable, without (red) and with precomputed library (cyan). Precomputing achieves almost 100% speedup on most schemes.

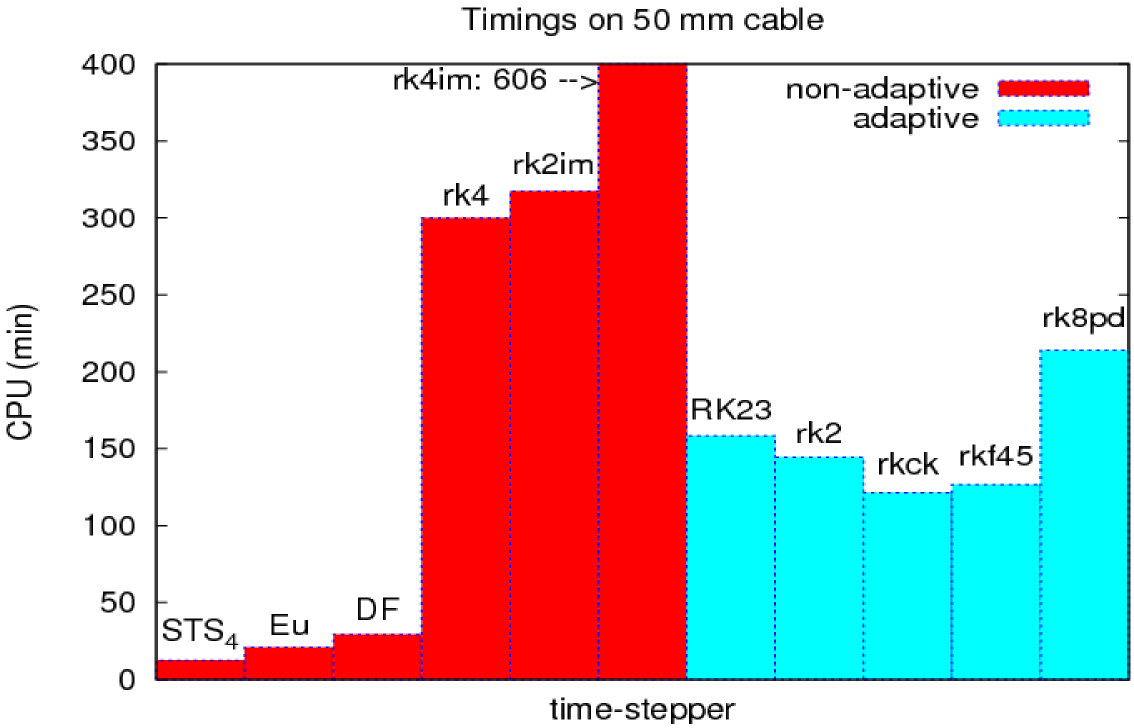

Figure 3: Timings of all eleven schemes on 50mm cable. Non-adaptive (red) and adaptive (cyan) schemes.

4 ACKNOWLEDGMENTS

We thank Dr. Jack Buchanan of the University of Tennessee Health Science Center in Memphis for providing biological parameters and suggestions. This work was supported by NIH grant 1R21GM080698-01A1.

REFERENCES

1. Alexiades, V, Amiez, & Gremaud, PA (1996). Super-time-stepping acceleration of explicit schemes for parabolic problems. Commun. Num. Meth. Eng, v.12, pp.12–31.

2. Brankin, RW, Gladwell, I, & Shampine, LF (1991). *RKSUITE*, obtained from http://www.netlib.org/ode/rksuite/

3. cellML, model *luo_rudy_1991_version06*, http://models.cellml.org/luo_rudy_1991_version06

4. GNU-GSL, GNU Scientific Library (accessed Jan.2010) http://ftp.gnu.org/pub/gnu/gsl

5. Hodgkin, AL & Huxley, AF (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. v.117, pp.500–544.

6. Keener, James & Sneyd, James (1998). Mathematical Physiology. Springer.

7. Li, C & Alexiades, V, (2010a). Comparison of time stepping schemes on the cable equation. Electronic J of Differential Equations, to appear.

8. Li, C & Alexiades, V, (2010b). Time stepping for the cable equation, Part 2: Parallel performance, in this volume.

9. Luo, CH & Rudy, Y (1991). A model of the ventricular cardiac action potential: depolarization, repolarization, and their interaction. Circ. Res. v.68, pp.1501–1526.

10. Mayers, DF & Morton, KW (1994). Numerical Solution of Partial Differential Equations. Cambridge U Press.

11. Plonsey, R & Barr, RC (2007). Bioelectricity, a quantitative approach, 3rd ed., Springer.

12. Sun, Y, Zhou, D, Rangan, AV, Cai D (2009). Library-based numerical reduction of the Hodgkin-Huxley neuron for network simulation. J. Comput. Neurosci, v.27(3), pp.369-390. Epub 2009 Apr 29.