Time Stepping for the cable equation, Part 2: Parallel 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

In this Part 2 of our work, we continue the discussion on time-steppers for the cable equation, modeling action potential propagation in cardiac myocyte cells. Here we report results from parallel computations.

The cable equation is a diffusion-reaction PDE for transmembrane voltage, whose reaction (source) term depends on several voltage-dependent quantities (gates), each satisfying a first order, highly nonlinear ODE.

The parallelization, for distributed computing via message passing, was done by domain decomposition and implemented in C using the MPI library.

We compare the performance of several low and high order, explicit and implicit time-stepping numerical schemes on the parallelized 1D cable equation with the Luo-Rudy I (1991) ionic model.

Keywords: Parallel computation, domain decomposition, Super-Time-Stepping, Dufort-Frankel, Runge-Kutta, action potential, Luo-Rudy ionic model, cable equation.

1 MATHEMATICAL MODEL

A full description of the model is given in our companion paper (Li & Alexiades, 2010b) and in an earlier paper (Li&Alexiades,2010a). Only an outline of the model is presented here, for completeness.

The parabolic partial differential equation (1), known as the *cable equation*,

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

is widely used to describe electrical propagation in excitable tissue (Keener & Sneyd, 1998; Plonsey & Barr, 2007) such as nerve fibers and heart muscle. Here $V(x,t)$ is the transmembrane voltage, R_a and C_m are the axial resistance and membrane capacitance of the tissue. I_{ion} represents the total ionic current, which, in the Luo-Rudy I (1991) ionic model we use here, depends on *seven* variables g^i, $i = 1, ..., 7$, known as *gates*. They are functions of local voltage, and each is governed by an ODE of the form

$$\frac{dg^i}{dt} = \alpha_i(V)\ (1 - g^i) - \beta_i(V)\ g^i, \qquad g = g^i, \quad i = 1, ..., 7. \tag{2}$$

The α's and β's, taking values between 0 and 1, are given by explicit formulas as functions of voltage V (Luo & Rudy, 1991).

The stimulus current I_{stim}, is the key to exciting the system. We apply a **single** stimulus at time 10 ms, 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. This instigates a single action potential that propagates along the cable. In our simulations, we use the values $C_m = 1.2\ \mu F/cm^2$ and $R_a = 300\ k\Omega$, same as in (Li & Alexiades, 2010b).

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 $Cai_{init} = 0.000178$, again, same as in (Li & Alexiades, 2010b).

The model consists of the PDE (1), the seven ODEs (2) and the above initial setup.

2 NUMERICAL SCHEMES

As in the companion Part 1 paper, we discretize the cable into M control volumes of uniform length Δx. Following standard Finite Volume discretization of the PDE (1), and applying equation (2) on each control volume, yields a system of $8 \times M$ ODEs, see equations (4)-(6) in the companion paper (Li & Alexiades, 2010b).

We apply the following numerical schemes to solve the ODE system. Again, additional details appear in (Li & Alexiades, 2010b).

- **Super-Time-Stepping (STS) Scheme**

  Super-time-stepping is a simple method to accelerate explicit schemes for parabolic problems, especially forward Euler (Alexiades et al.,1996). The scheme is described in the companion paper (Li & Alexiades, 2010b). In what follows, $\mathbf{STS}_4$ refers to STS scheme with $N = 4$ substeps per superstep and damping $\nu = 0.1$, whereas **Euler** refers to forward Euler scheme obtained by using $N = 1$, $\nu = 0$ in the STS scheme.

- **DuFort-Frankel (DF) Scheme**

  DuFort-Frankel is an explicit, 2-step, second order accurate in space and time scheme, theoretically unconditionally stable (Mayers & Morton,1994). It is described in the companion paper (Li & Alexiades, 2010b).

- **Runge-Kutta (RK) Schemes**

  The GNU Scientific Library (GSL 1.10) from GNU (GNU-GSL) provides various explicit and implicit, non-adaptive and adaptive ODE solvers. Adaptive time-steppers *cannot* be used in parallel computations due to the difficulty of synchronizing the computation on the worker processes. Thus we compare the three non-adaptive GSL solvers, namely, the explicit **rk4** (classical 4th order RK), and the implicit 2nd order **rk2imp** and 4th order **rk4imp**. See (Li & Alexiades, 2010b) for comparison of adaptive and non-adaptive schemes in serial computations.

3 NUMERICAL SIMULATIONS

A C program has been written implementing domain decomposition via message passing with the MPI library.

The full domain (entire cable) is divided evenly into N_w segments. The master MPI process assigns the nodes of each segment to one worker MPI process, which solves the ODEs at those nodes. At each time-step, worker processes get synchronized and they exchange their boundary values with their adjacent neighbors. Thus, all workers evolve at the same pace. Results are sent to the master process for printing out.

We employ the "library" approach, which was shown to be greatly effective in serial computations in (Li & Alexiades, 2010b). Namely, we pre-compute values of the coefficients $\alpha_i(V)$, $\beta_i(V)$, $i = 1, ..., 7$, of the ODEs (2) for the gates over a range of V's, and store them in a "library" file. A copy of this "library" file is loaded into memory on each worker and values of these coefficients at any V in the range are found by interpolation.

A standard measure of parallel performance is the *efficicency of parallelization*, defined as

$$\text{efficiency} = \frac{\text{timing on one worker}}{\text{timing on N}_w \text{ workers * N}_w}.$$

The numerical experiments on a $50mm$ cable reported here were performed on a cluster whose nodes consist of dual Quad-Core AMD Opteron 2378 2.4 GHz processors (8 cores per node). All schemes were run up to time $t_{max} = 2000\ ms$ with identical discretization parameters $\Delta x = 4\ \mu m$ and $\Delta t = 0.01\ ms$.

Table 1 lists CPU timings (total execution time) and efficiencies on 1, 2, 5 and 10 workers. The timings on 2, 5 and 10 workers are graphically illustrated in Fig.1.

Table 1. Timings and efficiencies on 50mm cable.

	1 worker	2 workers		5 workers		10 workers	
solver	CPU(min)	CPU(min)	Efficiency	CPU(min)	Efficiency	CPU(min)	Efficiency
STS_4	27	14	0.98	6	0.96	3	0.92
Euler	43	22	0.98	9	0.96	5	0.92
DF	46	23	0.99	10	0.96	5	0.91
GSLrk4	497	247	1.00	106	0.94	52	0.96
GSLrk2imp	492	245	1.00	102	0.96	59	0.83
GSLrk4imp	898	446	1.00	196	0.91	194	0.46

To convey graphically the speeding-up (reduction in execution time) achieved by parallelization, in Fig.2 we plot the timings of the three explicit, low order, non-RK schemes, $\mathbf{STS}_4$ (with $N = 4,\ \nu = 0.1$), **Euler** and **DF**. Similarly, Fig.3 shows the speeding-up of the three RK schemes **rk4**, **rk2imp**, and **rk4imp**. Note that the RK schemes (Fig.3) are an order of magnitude slower than the non-RK schemes (Fig.2), which is why we plot them in separate plots.

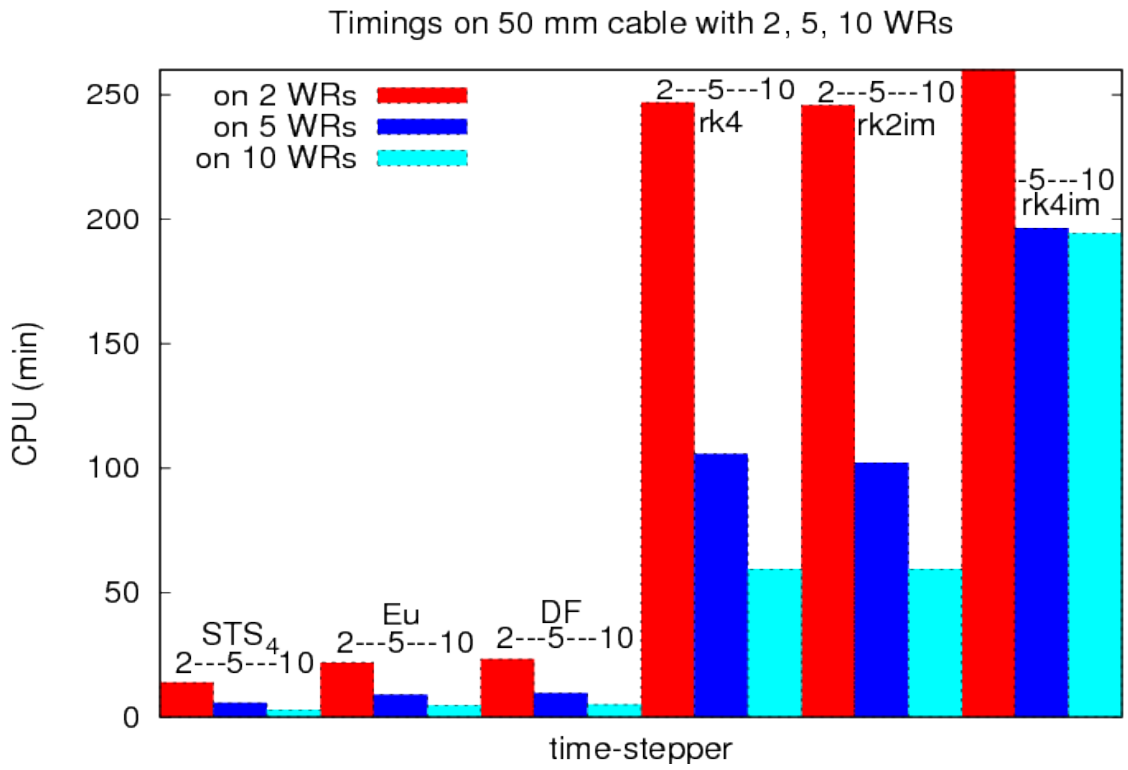

Figure 1: Comparison of timings of six solvers on 50mm cable running on 2, 5, 10 workers.

We observe the following, based on our experiments:

- All parallelized numerical schemes produce identical history and biological values with those obtained from our serial code, presented in (Li & Alexiades, 2010b).

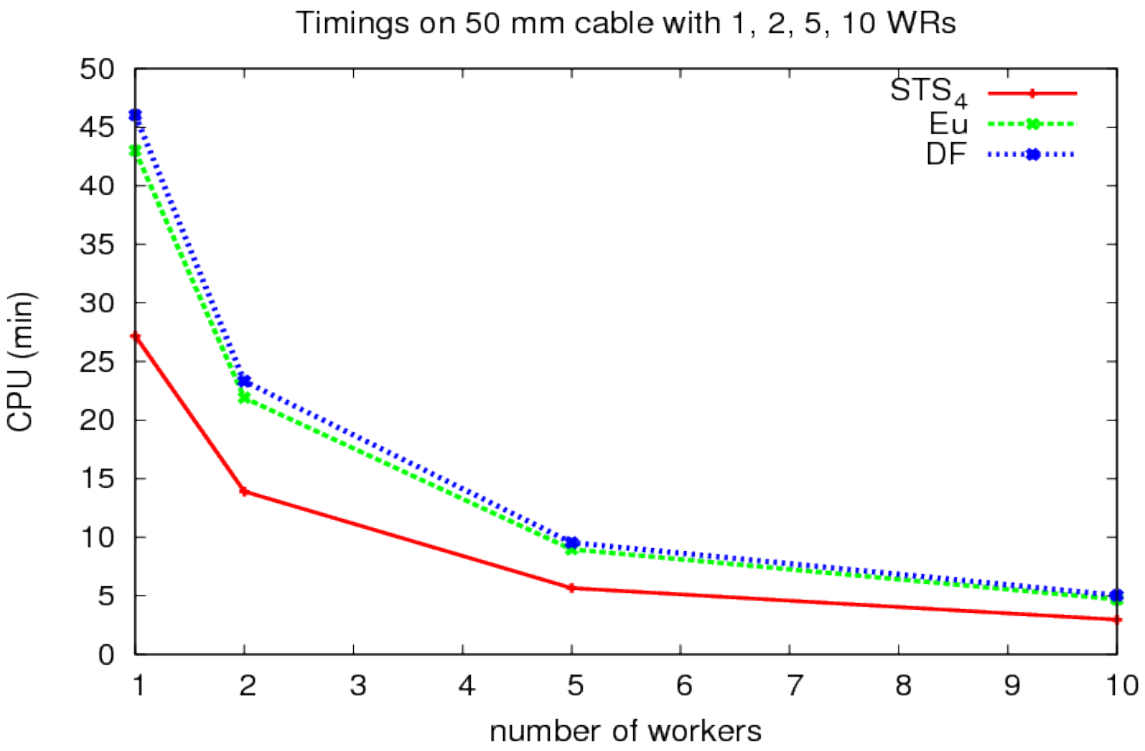

Figure 2: Timing curves of STS_4, Euler, and DF solvers for 50mm cable.

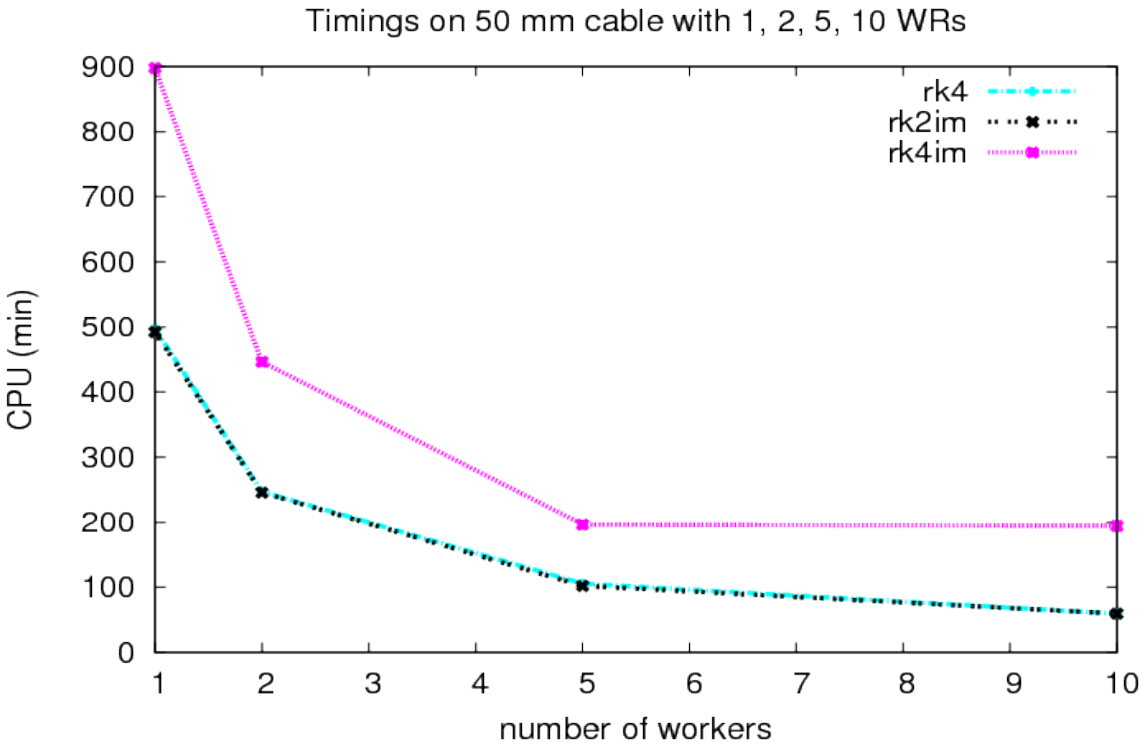

Figure 3: Timing curves of the RK solvers rk4, rk2imp, rk4imp for 50mm cable.

- Increasing the number of worker processes, up to 10 workers, significantly speeds up computation for all schemes, except for **rk4imp**, which costs almost the same time running on 5 and 10 worker processes.
- The low order explicit schemes ($\mathbf{STS}_4$, **Euler** and **DF**), with the same time step, are still $10-25$ times faster than high order schemes after parallelization. The ionic source restricts the time step to $\leq 0.01\ ms$ for all explicit schemes and for **rk2im**, and to $\leq 0.02\ ms$ for **rk4im**.
- On the basis of accuracy and effciency (CPU time), $\mathbf{STS}_4$ (i.e. STS with $N=4$, $\nu=0.1$) is still the winner among the schemes tested. When higher accuracy is needed, **rk4** would be best.

4 FUTURE WORK

We are extending our parallel codes to 2D and 3D, and also developing a space and time parallel code, hoping to achieve further speed ups. Simulating cardiac arrhythmias is the goal.

5 ACKNOWLEDGMENT

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 G, & Gremaud, PA (1996). Super-time-stepping acceleration of explicit schemes for parabolic problems, Commun. Num. Meth. Eng. v.12, pp.12–31.
2. cellML, *luo_rudy_1991_version06*, http://models.cellml.org/luo_rudy_1991_version06.
3. GNU-GSL, GNU Scientific Library (accessed Jan.2010) http://ftp.gnu.org/pub/gnu/gsl
4. Keener, J & Sneyd J (1998). Mathematical physiology, Springer.
5. Li, C & Alexiades, V, (2010a). Comparison of time stepping schemes on the cable equation, Electronic J of Differential Equations, to appear.
6. Li, C & Alexiades, V, (2010b). Time stepping for the cable equation, Part 1: Serial performance, in this volume.
7. 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.
8. Mayers, DF & Morton, KW (1994). Numerical solution of partial differential equations. Cambridge U Press.
9. Plonsey, R & Barr, RC (2007). Bioelectricity, a quantitative approach, 3rd ed., Springer.