Exponential integrators for a Markov chain model of the fast sodium channel of cardiomyocytes

Tomáš Starý[1], Vadim N. Biktashev[1]

[1]College of Engineering, Mathematics and Physical Sciences, University of Exeter, EX4 4QF, UK

The modern Markov chain models of ionic channels in excitable membranes are numerically stiff. The popular numerical methods for these models require very small time steps to ensure stability. Our objective is to formulate and test two methods addressing this issue, so that the timestep can be chosen based on accuracy rather than stability. Both proposed methods extend Rush-Larsen technique, which was originally developed to Hogdkin-Huxley type gate models. One method, "Matrix Rush-Larsen" (MRL) uses a matrix reformulation of the Rush-Larsen scheme, where the matrix exponentials are calculated using precomputed tables of eigenvalues and eigenvectors. The other, "hybrid operator splitting" (HOS) method exploits asymptotic properties of a particular Markov chain model, allowing explicit analytical expressions for the substeps. We test both methods on the Clancy and Rudy (2002) I_{Na} Markov chain model. With precomputed tables for functions of the transmembrane voltage, both methods are comparable to the forward Euler method in accuracy and computational cost, but allow longer time steps without numerical instability. We conclude that both methods are of practical interest. MRL requires more computations than HOS, but is formulated in general terms which can be readily extended to other Markov Chain channel models, whereas the utility of HOS depends on the asymptotic properties of a particular model. The significance of the methods is that they allow a considerable speed-up of large-scale computations of cardiac excitation models by increasing the time step, while maintaining acceptable accuracy and preserving numerical stability.

***Index Terms*—Markov chain, ion channel, numerical methods, Rush-Larsen method, exponential time-differentiation, operator splitting**

I. Introduction

MATHEMATICAL models are an essential part of the modern cardiac electrophysiology. They are used for hypothesis testing in research and as a guide for clinical decision. A typical definition of such a model is a high-dimensional (tens of equations) system of ordinary differential equations per excitable unit. Detailed simulations of the heart involve solving such systems for each of millions of cells placed in a mesh representing the cardiac tissue. Such large-scale models can be computationally extremely expensive, hence significant efforts are directed to develop efficient numerical methods for solving such systems.

A typical cardiac excitation model is centered around the Kirchhoff circuit law which gives

$$I_{\mathrm{stim}}(t) = C\frac{\mathrm{d}V_m}{\mathrm{d}t} + \sum_{\ell} I_{\ell}, \tag{1}$$

where C is the cell membrane's capacitance, $V_m = V_m(t)$ is the transmembrane potential difference, and I_ℓ, $\ell = 1, \ldots, L$, are currents through ion-specific channels. The currents, in turn, are determined by the Ohm's law,

$$I_\ell = G_\ell P_\ell(t)\left[V_m(t) - E_\ell(\vec{X}(t))\right] \tag{2}$$

where $E_\ell(\vec{X})$ is the ion-specific electromotive force, depending on the ionic concentrations $\vec{X}$ via the Nerst equation, G_ℓ is the total conductance of channels of type ℓ when they are all open, and P_ℓ is the probability of those channels to be open.

The components of the vector $\vec{X}$ are intra- and extra-cellular ionic concentrations, which change in time in the obvious way in accordance with the ionic fluxes and the corresponding volumes; some concentrations in some models are assumed constant. The dynamics of the open probabilities is much more nontrivial, as it reflects the conformation dynamics of the proteins constituting the ion channels .

The classical description of these dynamics, going back to Hodgkin and Huxley (1952) [1], has the form

$$P_\ell(t) = \prod_{i\in\mathcal{I}(\ell)} y_i \tag{3}$$

with a popular, although different from the original Hodgkin and Huxley's, interpretation that the set $\mathcal{I}(\ell)$ corresponds the subunits of the channel, called "gates". These subunits are assumed statistically independent, each of them can be either in an "open" or a "closed" state, and the channel is open if and only if each of the subunits is open. Variables y_i then are open probabilities of the gates, and their dynamics are described by

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = \alpha_i(V_m)(1-y_i) - \beta_i(V_m)y_i, \tag{4}$$

where $\alpha_i(V_m)$ are opening rates and $\beta_i(V_m)$ are closing rates.

For instance, the original Hodgkin-Huxley description of the fast sodium current (I_{Na}) channel uses $\#\mathcal{I}(I_{\mathrm{Na}}) = 4$ gates, three of which, called m-gates, have identical opening $\alpha_1(V_m) = \alpha_2(V_m) = \alpha_3(V_m) = \alpha_m(V_m)$ and closing $\beta_1(V_m) = \beta_2(V_m) = \beta_3(V_m) = \beta_m(V_m)$ rates, and the fourth, called h-gate, has rates $\alpha_4(V_m) = \alpha_h(V_m)$ and $\beta_4(V_m) = \beta_h(V_m)$, hence for this case we have

$$P_{I_{\mathrm{Na}}}(t) = m^3 h, \tag{5}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m, \tag{6}$$

Manuscript received ??; revised ??. Corresponding author: V. N. Biktashev

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h. \tag{7}$$

A more recent approach is modelling the probabilities of the channel molecules, as a whole, to be in specific conformation states, without the restricting assumptions of statistically independent subunits and only two states for any subunit. This gives generic continuous time Markov chain (MC) models. Let the probability of the ℓ'th channel to be in the k'th state at time t be $u_k(t)$ ("state occupancy"), $k \in \mathcal{K}(\ell)$, $K(\ell) = \#\mathcal{K}(\ell)$, and all such probabilities be considered components of the state (column-) vector $\vec{u} = (u_k)^\top = \vec{u}(t)$. Let g_k be the relative permeability of the state k, then we have

$$P_\ell(t) = \sum_{k=\mathcal{K}(\ell)} g_k u_k(t). \tag{8}$$

Typically, $g_k = \delta_{k,k^*}$ where k^* is the "open state". The time evolution of the state vector is described by the system of linear ODEs, known in particular as *Kolmogorov (forward) equations*, or *master equation*, of the form

$$\frac{\mathrm{d}\vec{u}}{\mathrm{d}t} = \boldsymbol{A}(V_m)\vec{u}, \tag{9}$$

where the non-diagonal components of the matrix $\boldsymbol{A}(V_m)$ are the transition rates (TR) between the states, and the diagonal components are defined by the condition $\sum_{k\in\mathcal{K}} u_k = 1$ and consequently sum of any column of $\boldsymbol{A}$ should be zero.

The ODE system for cellular membrane can be solved on a computer using standard numerical solvers. A typical solver iteratively computes the states of the system using time-stepping algorithms, that is computing the states at times $t_n = t_0 + n\Delta t$. The size of the time step Δt is inversely proportional to the computational cost, measured as CPU time required for the computation. Increasing the time step is a straightforward way of reducing the computational cost.

The maximal acceptable time step is limited by considerations of accuracy and stability (see e.g. [2, Sections 5.10, 5.11]). Whereas the former is "relative" in that it depends on the aims of the research, the latter has a more "absolute" character in that if stability conditions are not satisfied, the solution is unusable for any purpose. Typically, when the time step exceeds the stability limit, the numerical solution is characterized by wild oscillations around the exact solution, and quite often will lead to numerical overflow.

Simple explicit solvers suffer from instabilities the most, and implicit, stable methods, applicable to generic systems of ODEs, are complicated and often costly. The motivation for our research was that taking into account the specific properties of the problem can offer some advantages. Specifically, we have in mind two distinct considerations.

One consideration is that the TRs can range through several orders of magnitude, and some of them can be much faster than other processes described by the excitable cell model. This split of the speeds of the variables suggest a possibility to exploit asymptotic methods.

The other consideration is the linearity of the system (9). Here we are inspired by the example of the exponential integrator algorithm developed by Rush and Larsen in 1978 [3]. It is based on the assumption that the transmembrane voltage, on which the TRs in the gate model (4) depend, changes only slightly during one time step Δt. So during one time step, the TRs can be approximated by constants, and the equation (4) is then solved analytically. The solution can be conveniently defined in terms of the "steady state" $\bar{y}_i = \alpha_i(V_m)/[\alpha_i(V_m) + \beta_i(V_m)]$ and the "time constant" $\tau_i = 1/[\alpha_i(V_m) + \beta_i(V_m)]$ at a given potential V_m presumed constant:

$$y_i^{n+1} = \bar{y}_i(V_m) - [\bar{y}_i(V_m) - y_i^n]\exp\left(-\frac{\Delta t}{\tau_i(V_m)}\right). \tag{10}$$

The Rush-Larsen (RL) scheme is easy to implement, gives good results and is very popular in computational cardiac electrophysiology. Its stability and approximation properties have been extensively discussed in literature, including its relation to general exponential integrators family, its extension beyond gating variables by linearization, and improving its approximation properties, see e.g. [4], [5], [6]. However it is designed for a single ODE and is not immediately applicable for MC models which are systems of coupled ODEs. And yet MC models are known to suffer from severe numerical instability issues, just as, or even more than, the gate models (Fig. 1). The classical techniques for numerical solution of continuous-time MC models involve finding the eigenvalues and associated eigenvectors of the transition matrix. Direct implementation of this approach to very large MCs is problematic, see e.g. [7]. However the MCs describing ionic channels are relatively small so the direct approach is feasible.

In this paper, we discuss two methods for numerical solution of MC models based on these two considerations.

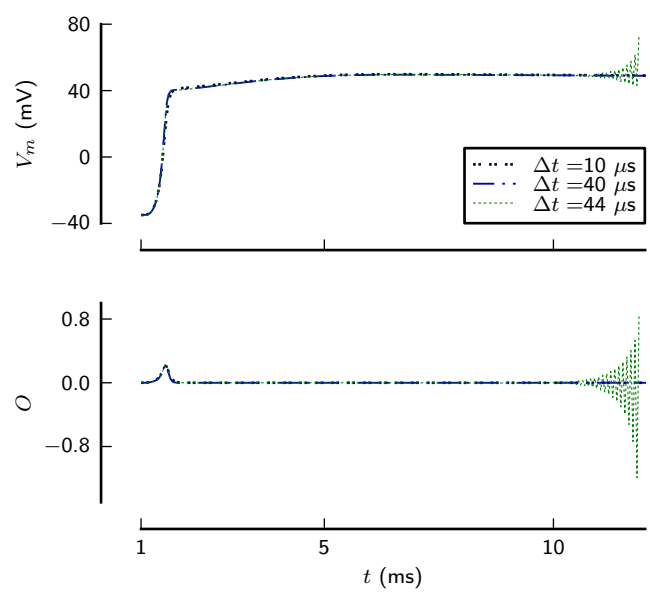

Fig. 1. Instability of I_{Na} Markov chain model at longer time steps. The model was solved using forward Euler method using three different step sizes: $\Delta t = 10\,\mu$s, $\Delta t = 40\,\mu$s, and $\Delta t = 44\,\mu$s. The top panel shows the membrane potential (V_m), the bottom panel shows state occupancy of the open state O.

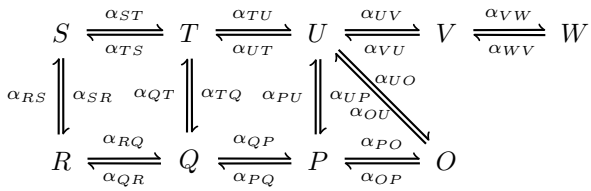

Fig. 2. Markov chain model of I_{Na} channel.

TABLE I
CHANGE OF STATES VARIABLES TERMINOLOGY.

Standard	Our	initial value
O	O	4.386×10^{-8}
C_1	P	5.329×10^{-5}
C_2	Q	1.064×10^{-2}
C_3	R	8.018×10^{-1}
IC_3	S	1.436×10^{-1}
IC_2	T	1.907×10^{-3}
IF	U	1.111×10^{-5}
IM_1	V	8.417×10^{-4}
IM_2	W	4.118×10^{-2}

II. METHODS

A. Models

To test the suggested numerical methods we have chosen the MC model of the I_{Na} channel by Clancy and Rudy [8] (Fig. 2), which is one of the most popular MC models. We used the formulation of the MC model and the whole cell model into which it was incorporated, as implemented in the authors' code kindly provided by C.E. Clancy. It most closely corresponds to the Luo-Rudy model [9] with modifications described in [10], [11], and some further minor differences. For the sake of reproducibility of our results, we describe the whole model in the supplementary material, highlighting all the differences from the published models that we have detected. For the same purpose, we put a simplified version of the C code we used in the simulations described below in the supplementary materials.

For convenience, we changed the notation for the MC states and TRs. The states were named in alphabetical order, starting with O for the open state, in a clockwise direction as in Fig. 2. See Tab. I and Tab. II for the correspondence with the original notation. The model contains 9 interconnected states. The state O represents the conformation of the ion channel that allows the flow of ions between the intracellular and extracellular environment. The remaining states (P, Q, R, S, T, U, V and W) represent non-conductive conformations of the channel, so we can say that for this model $g_k = \delta_{k,1}$, where $u_1 = O$. There are 11 possible bidirectional transitions between states, but some of the corresponding 22 TRs are described by identical functions, so there are only 14 distinct TR definitions. We denote the TRs by α with a subscript showing the direction of the transition, e.g. α_{PO} is the transition rate from state P into state O. See Tab. II for the link with the original notations.

The TRs are shown on Fig. 3 as functions of the transmembrane potential V_m in a physiologically relevant range. The values of TRs vary across several orders of magnitude, from $10^{-11}\,\mathrm{ms}^{-1}$ to $10^{2}\,\mathrm{ms}^{-1}$. Some of the TRs are high at the lower potentials, some are fast at higher potentials, and some are uniformly low.

TABLE II
CHANGE OF TRANSITION RATES (TR) TERMINOLOGY.

Standard	Our
α_{11}	α_{RQ}, α_{ST}
α_{12}	α_{QP}, α_{TU}
α_{13}	α_{PO}
β_{11}	α_{QR}, α_{TS}
β_{12}	α_{PQ}, α_{UT}
β_{13}	α_{OP}
α_2	α_{OU}
β_2	α_{UO}
α_3	α_{UP}, α_{TQ}, α_{SR}
β_3	α_{PU}, α_{QT}, α_{RS}
α_4	α_{UV}
β_4	α_{VU}
α_5	α_{VW}
β_5	α_{WV}

The conductive (open) state O is the only state that has immediate effect on the I_{Na} current. The remaining 8 states of the model can affect the current only indirectly by transitions to the open state O. The time evolution of a generic state occupancy state u_k is described by a differential equation of the form

$$\frac{\mathrm{d}u_k}{\mathrm{d}t} = \sum_{k' \in \mathcal{K}'(k)} \left(\alpha_{k',k} u_{k'} - \alpha_{k,k'} u_k\right), \tag{11}$$

where $\mathcal{K}'(k)$ is the set of all the states interconnected with state k, which can be readily found from the diagram. For example, the occupancy of the open state, O, is described by the following equation:

$$\frac{\mathrm{d}O}{\mathrm{d}t} = \alpha_{PO}\, P + \alpha_{UO}\, U - (\alpha_{OP} + \alpha_{OU})\, O.$$

By taking the sum of the equations (11) for all k, one can see that the sum of the right-hand sides equals to zero, and therefore the system observes a states conservation law, which is consistent with the definition of u_k as probabilities, implying $\sum_{k \in \mathcal{K}} u_k = 1$. This is of course a generic property of a continuous Markov chain. So the differential equations in the model are not independent, which creates a possibility of reducing the number of equations from 9 to 8, by computing one of the occupancies through the conservation law rather than from its differential equation. However the computational gain from this is insignificant, and instead we used any deviations from the conservation law as an indicator of the accuracy of the computations.

B. Numerical Methods

1) Forward Euler

The standard forward Euler (FE) method is the simplest timestepper for differential equations. It defines the solution at the next time step, $\vec{u}_{n+1} = \vec{u}(t_{n+1})$, in terms of the same at the previous time step, $\vec{u}_n = \vec{u}(t_n)$, using one-step forward-time finite different approximation of the time derivative, which for the system (9) gives

$$\vec{u}_{n+1} = \vec{u}_n + \Delta t \boldsymbol{A}(V_m(t_n))\vec{u}_n. \tag{12}$$

The time discretization step $\Delta t = t_{n+1} - t_n$ is presumed here the same for all steps of a simulation.

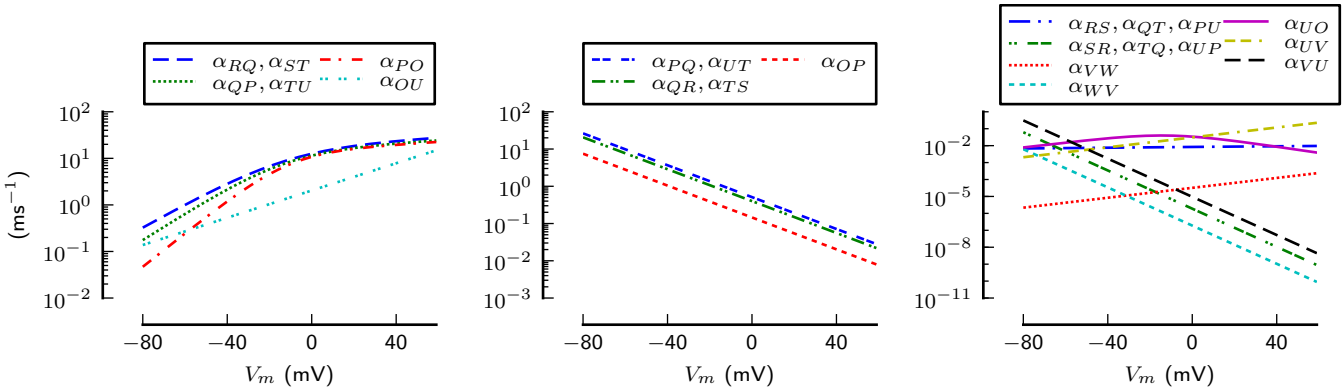

Fig. 3. Transition rates (TR) of I_{Na} Markov chain model. The left panel shows fast TR at high potentials: subsystem $\boldsymbol{A}_0$; the middle panel shows fast TR at low potentials: subsystem $\boldsymbol{A}_1$; and the right panel shows slow TR in the entire range of potentials: subsystem $\boldsymbol{A}_2$.

2) Matrix Rush-Larsen

The proposed Matrix Rush-Larsen method (MRL) assumes that the matrix $\boldsymbol{A}(V_m)$ changes only slightly during one time step and therefore can be approximated by a constant. The solution of (9) can then be written in terms of the matrix exponential,

$$\vec{u}_{n+1} = \exp\left[\boldsymbol{A}(V_m(t_n))\,\Delta t\right]\,\vec{u}_n. \tag{13}$$

We assume that the matrix $\boldsymbol{A}(V_m)$ is diagonalizable, i.e. can be represented in the form $\boldsymbol{A}(V_m) = \boldsymbol{S}(V_m)\boldsymbol{\Lambda}(V_m)\boldsymbol{S}(V_m)^{-1}$, where matrix $\boldsymbol{S}(V_m)$ is composed of the eigenvectors concatenated as column vectors, and matrix $\boldsymbol{\Lambda}(V_m)$ contains eigenvalues placed on the corresponding places on the diagonal. A sufficient condition of diagonalizability of a matrix is that all its eigenvalues are distinct, and this is the generic situation; but we of course check that it actually takes place in every case. Then the matrix exponential is calculated as

$$\vec{u}_{n+1} = \boldsymbol{S}(V_m)\,\exp\left(\boldsymbol{\Lambda}(V_m)\Delta t\right)\,\boldsymbol{S}(V_m)^{-1}\vec{u}_n, \tag{14}$$

where the exponential $\exp\left[\boldsymbol{\Lambda}(V_m)\Delta t\right]$ of the diagonal matrix $\boldsymbol{\Lambda}(V_m)\Delta t$ is obtained by exponentiation of its diagonal elements.

As the numerical solution of the eigenvalue problem is computationally expensive, we precompute the matrices $\boldsymbol{S}(\tilde{V}_j)$, $\boldsymbol{S}(\tilde{V}_j)^{-1}$ and $\boldsymbol{\Lambda}(\tilde{V}_j)$ for a fine grid of physiological potentials, $\tilde{V}_j = V_{\min} + j\Delta V$, $j \in \mathcal{J} = \{0, 1, \ldots j_{\max}\}$, $\tilde{V}_j \leq V_{\max}$, $V_{\min} = -100$, $V_{\max} = 70$, $\Delta V = 0.01$ (all in mV) before compile time and save them in a file.

At start time, the eigenvalue and eigenvector matrices are loaded from the file and we precompute, for Δt used in the particular simulation, the transition matrices

$$\boldsymbol{T}_j = \boldsymbol{T}\left(\tilde{V}_j\right) = \boldsymbol{S}\left(\tilde{V}_j\right)\exp\left[\boldsymbol{\Lambda}\left(\tilde{V}_j\right)\Delta t\right]\boldsymbol{S}\left(\tilde{V}_j\right)^{-1} \tag{15}$$

for all $j \in \mathcal{J}$. At the run time, the solver simply refers to the tabulated transition matrix $\boldsymbol{T}_j$,

$$\vec{u}_{n+1} = \boldsymbol{T}_{j(n)}\vec{u}_n \tag{16}$$

where $\tilde{V}_{j(n)}$ is the tabulated transmembrane potential that is the nearest to $V_m(t_n)$.

Along with the code, we provide precomputed files for a voltage step size of $\Delta V = 0.1$ (size of 4.85 MB), that are sufficient to obtain accurate results. The tables with $\Delta V = 0.01$ mV of 48.5 MB size, used for the simulations presented, are available from the authors upon request.

The method of tabulation (tab.) can be applied to all the presented numerical methods. However, its benefit is most essential in the MRL method, because matrix exponentiation is computationally expensive. The accuracy of the tabulation is dependent on the voltage step (here 0.01 mV) which is a matter of choice depending on memory availability and allowable precompile and start-time computation time.

3) Hybrid Operator Splitting

The MRL achieves the purpose in principle but is relatively costly as multiplication by a dense $K \times K$ matrix $\boldsymbol{T}_{j(n)}$ is required at each time step. On the other hand, it did not at all exploit the specific structure of the TR, illustrated by Fig. 3, that is, that the matrix $\boldsymbol{A}(V_m)$ is sparse and some TRs are much faster than others for some voltage ranges. Hence we propose a hybrid operator splitting method (HOS), which combines FE and MRL, and exploits the asymptotic structure of the TRs. In this method, we set

$$\boldsymbol{A}(V_m) = \boldsymbol{A}_0(V_m) + \boldsymbol{A}_1(V_m) + \boldsymbol{A}_2(V_m)$$

as described in Fig. 3: $\boldsymbol{A}_0$ contains only TRs that are fast at high values of V_m (α_{RQ}, α_{ST}, α_{QP}, α_{TU}, α_{PO} and α_{OU}); $\boldsymbol{A}_1$ contains only TRs that are fast at low values of V_m (α_{PQ}, α_{UT}, α_{QR}, α_{TS} and α_{OP}); and $\boldsymbol{A}_2$ contains only uniformly slow TRs (α_{RS}, α_{QT}, α_{PU}, α_{SR}, α_{TQ}, α_{UP}, α_{VW}, α_{WV}, α_{UO}, α_{UV} and α_{VU}). Explicit expressions for $\boldsymbol{A}_j$ are given in the Supplement.

Every timestep is then done in three substeps,

$$\vec{u}_{n+1/3} = \exp(\Delta t\boldsymbol{A}_0(V_m(t_n)))\,\vec{u}_n, \tag{17}$$

$$\vec{u}_{n+2/3} = \exp(\Delta t\boldsymbol{A}_1(V_m(t_n)))\,\vec{u}_{n+1/3}, \tag{18}$$

$$\vec{u}_{n+1} = \vec{u}_{n+2/3} + \Delta t\boldsymbol{A}_2(V_m(t_n))\,\vec{u}_{n+2/3}. \tag{19}$$

In our case the matrix exponentials in the two fast subsystems (17) and (18) are found analytically, through solving the corresponding ODE systems. This is possible because some of the equations corresponding to the matrices $\boldsymbol{A}_0(V_m)$ and $\boldsymbol{A}_1(V_m)$ are coupled in a specific manner and can be solved one by one where solution of one equation is substituted in the next etc. The full expressions and the method of derivation

are given in the Supplement; here we present the solution for state O in the equation (17) as an example:

$$O_{n+1/3} = \mu_{OU} O_n + K_{PO} P_n + K_{QO} Q_n + K_{RO} R_n,$$

where

$$\begin{aligned}
K_{PO} &= \frac{\alpha_{PO}(\mu_{PO} - \mu_{OU})}{\alpha_{OU} - \alpha_{PO}}, \\
K_{QO} &= \frac{\alpha_{PO}\alpha_{QP}(\mu_{QP} - \mu_{OU})}{(\alpha_{PO} - \alpha_{QP})(\alpha_{OU} - \alpha_{QP})} \\
&\quad - \frac{\alpha_{PO}\alpha_{QP}(\mu_{PO} - \mu_{OU})}{(\alpha_{PO} - \alpha_{QP})(\alpha_{OU} - \alpha_{PO})}, \\
K_{RO} &= -\frac{\alpha_{PO}\alpha_{QP}\alpha_{RQ}(\mu_{QP} - \mu_{OU})}{(\alpha_{QP} - \alpha_{RQ})(\alpha_{PO} - \alpha_{QP})(\alpha_{OU} - \alpha_{QP})} \\
&\quad + \frac{\alpha_{PO}\alpha_{QP}\alpha_{RQ}(\mu_{PO} - \mu_{OU})}{(\alpha_{QP} - \alpha_{RQ})(\alpha_{PO} - \alpha_{QP})(\alpha_{OU} - \alpha_{PO})} \\
&\quad + \frac{\alpha_{PO}\alpha_{QP}\alpha_{RQ}(\mu_{RQ} - \mu_{OU})}{(\alpha_{QP} - \alpha_{RQ})(\alpha_{PO} - \alpha_{RQ})(\alpha_{OU} - \alpha_{RQ})} \\
&\quad - \frac{\alpha_{PO}\alpha_{QP}\alpha_{RQ}(\mu_{PO} - \mu_{OU})}{(\alpha_{QP} - \alpha_{RQ})(\alpha_{PO} - \alpha_{RQ})(\alpha_{OU} - \alpha_{PO})}, \\
\mu_{jk} &= \mathrm{e}^{-\alpha_{jk}\Delta t}.
\end{aligned}$$

The slow subsystem (19) uses FE, and since it contains only uniformly slow TRs, it can tolerate large time-steps, allowed by other components of the cell model, without loss of stability.

C. A priori error estimates

Estimates by standard methods (e.g. [2, Chapter 5], see details in the Supplement), show that all three numerical schemes have local truncation errors of the second order, i.e. $\mathcal{E}\Delta t^2 + \mathcal{O}\left(\Delta t^3\right)$, although the coefficients $\mathcal{E}$ vary: for FE we have $\mathcal{E}_{\mathrm{FE}} \leq \frac{1}{2}\left(\|\boldsymbol{A}\|^2 + \|\mathrm{d}\boldsymbol{A}/\mathrm{d}V_m\||\mathrm{d}V_m/\mathrm{d}t|\right)$, for MRL we have $\mathcal{E}_{\mathrm{MRL}} = \frac{1}{2}\|\mathrm{d}\boldsymbol{A}/\mathrm{d}V_m\||\mathrm{d}V_m/\mathrm{d}t|$, and for HOS it is composed of contributions of the three substeps plus the error due to operator splitting, $\mathcal{E}_{\mathrm{HOS}} \leq \frac{1}{2}|\mathrm{d}V_m/\mathrm{d}t|\left(\|\mathrm{d}\boldsymbol{A}_0/\mathrm{d}V_m\| + \|\mathrm{d}\boldsymbol{A}_1/\mathrm{d}V_m\| + \|\mathrm{d}\boldsymbol{A}_2/\mathrm{d}V_m\|\right) + \frac{1}{2}||\boldsymbol{A}_2||^2 + \mathcal{E}_{\mathrm{OS}}$, where $\mathcal{E}_{\mathrm{OS}} = \frac{1}{2}\|[\boldsymbol{A}_1, \boldsymbol{A}_0] + [\boldsymbol{A}_2, \boldsymbol{A}_0] + [\boldsymbol{A}_2, \boldsymbol{A}_1]\|$, and $[\boldsymbol{X}, \boldsymbol{Y}] \equiv \boldsymbol{XY} - \boldsymbol{YX}$. So comparison of MRL and HOS with FE depends on the solution, but in any case accuracy of HOS it contingent on $\boldsymbol{A}_0$, $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ not being large at the same time, to ensure relative smallness of $\mathcal{E}_{\mathrm{OS}}$.

D. Implementation

Most of the algorithms described here were implemented in C language in double precision floating point arithmetics and compiled using GNU Compiler Collection (GCC) (version 4.7.2). The exception is computation of eigenvalues and eigenvector tables, which was done using mathematical software Sage [12] (version 5.9). Simulation were performed on Intel Core i5-3470 CPU with the clock frequency 3.20GHz under GNU/Linux operating system (distribution Fedora 18).

III. Results

Figure 4 shows the detail of the first millisecond of simulated cardiac excitation, the onset of an action potential (AP). The I_{Na} Markov chain model was solved using the three suggested integration methods: forward Euler (FE), matrix Rush-Larsen (MRL), and hybrid operator splitting (HOS), as described in the methods section. The model was solved with time step $\Delta t = 10\,\mu$s, $\Delta t = 40\,\mu$s, and $\Delta t = 100\,\mu$s, except for FE, which was also solved for $\Delta t = 1\,\mu$s, to be used as a reference, but not for $\Delta t = 100\,\mu$s, due to instability.

The model excitable cell was initially at the resting state, and at the time $t = 1$ ms, an AP was initialized by instantaneous injection of potassium ions, raising the membrane potential to $V_m = -35$ mV. The initial conditions of the states of the I_{Na} MC model are specified in Tab. I and the initial states of the remaining variables of the model can be found in the supplementary material. Before the initiation, more than 90% of the channels reside in the states R and S, which require at least three transitions to get to the open state. After the initiation, the channels start to transit rapidly towards the open state O and then to the state U. Within about 0.7 ms almost all channels reside in the state U. Then, the channels slowly transit to the state V, where they stay until the resting potential V_m is recovered. The states U and V are similar to the situation when the inactivation gate h is closed in the gate model. The states S and T have less than 10% occupancy during all the stages of the action potential.The plot of W is omitted, as this variable changes very little during the time interval shown.

The results for the time step $\Delta t = 10\,\mu$s are consistent in all panels. The FE is still stable at time step $\Delta t = 40\,\mu$s, however, compared to MRL and HOS, the FE solution is less accurate, resulting in a higher peak and faster decay of both the open state O occupancy and the resulting I_{Na} current.

Comparison of the solutions for $O(t)$ with the reference $O^*(t)$, obtained by FE with $\Delta t = 1\,\mu$s, is shown on Fig. 4 (first column, fourth row). We see that MRL and HOS approximate O^* better than FE at the same time steps. This is consistent with results of evaluation of the error estimates over the AP solution: we have $\max(\mathcal{E}_{\mathrm{FE}}) \approx 2700$, $\max(\mathcal{E}_{\mathrm{MRL}}) \approx 118$, $\max(\mathcal{E}_{\mathrm{HOS}}) \approx 125$, with $\max(\mathcal{E}_{\mathrm{OS}}) \approx 19$, all in ms^{-2}, and $\min(\mathcal{E}_{\mathrm{FE}}/\mathcal{E}_{\mathrm{HOS}}) \approx 2.3$, $\min(\mathcal{E}_{\mathrm{FE}}/\mathcal{E}_{\mathrm{MRL}}) \approx 3.2$ (see the Supplement). This suggests that exponential integrators can be useful, for their accuracy, even when instability is not a concern, say in systems with slower dynamics, such as I_{Ks}.

At longer time steps, FE is unstable (Fig. 1 illustrates a mild case of the instability), while MRL and HOS continue to provide stable solutions. At $\Delta t = 100\,\mu$s, the peak of the most important component of $\vec{u}$, the occupancy of the open state O, is slightly lower than at shorter timesteps. On the other hand, the decrease of the peak of the total I_{Na} current in these two methods (HOS, MRL at $\Delta t = 100\,\mu$s) is relatively small compared to the decrease of the open state occupancy. Also, the decay of the I_{Na} current in the MRL $\Delta t = 100\,\mu$s is slower than in the other cases. Note that the lead of the APs onsets at $\Delta t = 100\,\mu$s against smaller time steps is comparable to the value of Δt.

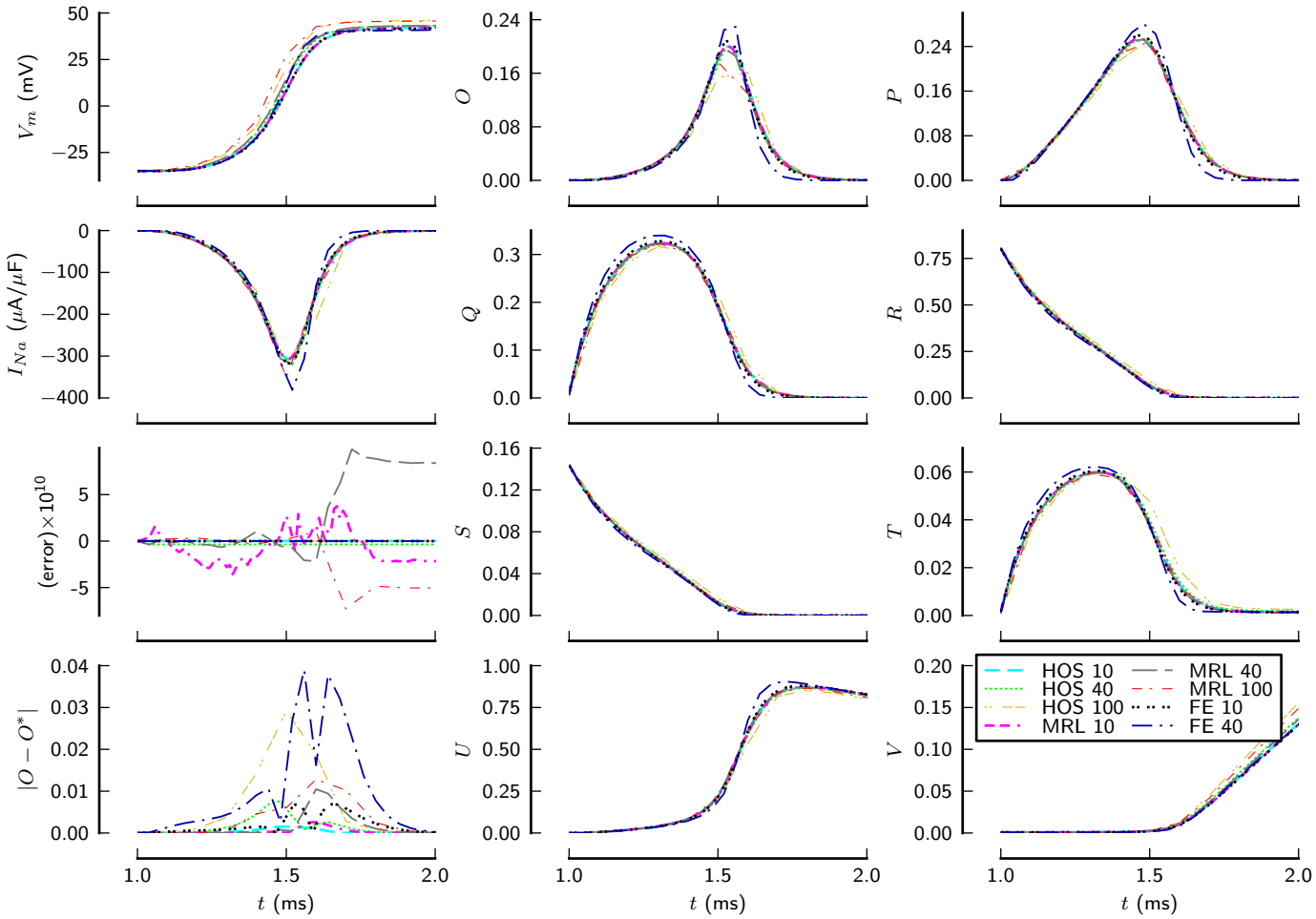

Fig. 4. Cardiac cell action potential (AP) simulations (detail of the first second) with I_{Na} Markov chain model solved using: hybrid operator splitting method (HOS) – light colours; matrix Rush-Larsen (MRL) – semi-dark colours; and forward Euler (FE) – dark colours; The left column shows: membrane potential (V_m), sodium current (I_{Na}), error calculated as deviation from the states conservation law, and deviation of the state occupancy O from the solution using FE with $\Delta t = 1\,\mu$s (O^*); the other two columns show the state occupancies. The model was solved with time step $\Delta t = 10\,\mu$s, $\Delta t = 40\,\mu$s, and $\Delta t = 100\,\mu$s represented by thick, middle thick, thin lines respectively.

TABLE III
ELAPSED SIMULATION TIME [S]. CELL MODEL [8] 100 PULSES WITH CL=1000 MS.

	$\Delta t = 10\,\mu$s		$\Delta t = 40\,\mu$s		$\Delta t = 100\,\mu$s	
I_{Na} Model	I_{Na}	Total	I_{Na}	Total	I_{Na}	Total
FE	4.88	22.34	1.24	5.59		
FE (tab.)	2.48	19.98	0.60	5.01		
MRL (tab.)	2.96	20.45	0.74	5.16	0.28	2.06
HOS	8.11	25.71	2.01	6.43	0.81	2.58
HOS (tab.)	2.81	20.31	0.71	5.11	0.29	2.05

Approximation of the whole APs rather than just their onsets is illustrated in Fig. 5.

Further increase of the time steps (not shown) in MRL and HOS gives significant errors in the AP, e.g. at $\Delta t = 200\,\mu$s there is a 30 mV overshoot. Stability persists for much longer: for HOS the solution becomes unphysical (a negative concentration) at about $\Delta t = 2$ ms without loss of stability, and for MRL an instability occurs at about 7.5 ms, although the solution is then also very different from the true AP.

Table III illustrates the efficiency of the three methods at three different time steps Δt. This was done by measuring time taken by simulations consisting of 100 pulses with a cycle length (CL) of 1000 ms without any output. The pulses were initialized by an instantaneous injection of potassium ions of a sufficient amount to set the membrane potential to $V_m = -35$ mV. The table shows times taken by the whole cell model ("Total") and by the Markov Chain model computations ("I_{Na}"). The times shown are median values from six separate simulations in each case to minimize the effect of other processes running on the computer.

At $\Delta t = 10\,\mu$s, FE is the most efficient method. Computation of I_{Na} accounts for 21.8% and 31.5% of the overall computation cost in FE and HOS respectively. Tabulation allows reduction of the computation cost of I_{Na} by 49.1% and 65.3% in FE and HOS. MRL was used only with tabulation using the precomputed eigenvalues and eigenvectors matrices and the computational cost at the $\Delta t = 10\,\mu$s is comparable with FE. These proportions are consistent with the results at time step $\Delta t = 40\,\mu$s and $\Delta t = 100\,\mu$s for MRL and HOS. So, at the same time step, the computational costs of the proposed methods are slightly higher, but the accuracies are somewhat better, compared to FE. The most important benefit of HOS

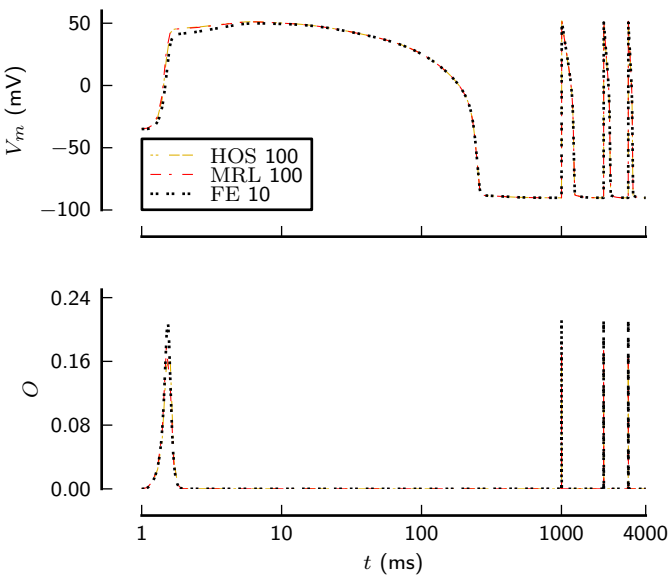

Fig. 5. Approximation properties of HOS ($\Delta t = 100\,\mu$s) and MRL ($\Delta t = 100\,\mu$s) compared to FE ($\Delta t = 10\,\mu$s)during a single cell simulation of 4 action potentials with cycle length (CL) of 1000 ms (logarithmic scale): membrane potential V_m (top panel), open state occupancy O (bottom panel).

and MRL is, however, the possibility of using larger time steps.

IV. Conclusion

Both proposed methods maintain stability at larger time steps, and improve the accuracy of the solution at the same time step, compared to the explicit ODE solver (FE). When tabulated, those methods are comparable to FE in computational cost. As expected, using larger time steps results in reduction of computational cost.

MRL method extends the popular RL method, developed for gate models, to Markov chain models. MRL is more universal than HOS, and may be made "automatic". The only restriction of our implementation is the assumption of diagonalizability of matrix $\boldsymbol{A}(V_m)$ for all voltages. If in another model this happens not to be the case, then some more sophisticated approach would be needed. If non-diagonalizability is a regular feature, say due to identical definitions of some of the TRs, then a Jordan form can be used instead; if it only happens at selected voltages, then interpolation of matrices $\boldsymbol{T}(V_m)$ may be sufficient.

HOS method depends on the possibility to split the transition rates to multiple (three in our case) sub-systems according to their speeds, and solve each of the subsystems on its own. Our solution benefits from the possibility of solving the fast subsystems analytically. Implementing the analytical solution results in even better speed-up as the resulting timestepping matrices are sparse. However, the possibility of a suitable analytical solution is not guaranteed for a general MC model. In this case, the fast time subsystems can be solved using diagonalization like in the MRL method, which might require additional computational time.

Finally we comment on the order of approximation. In this paper we considered first order schemes, and they are most popular in practice. However, the approximation order can be improved by using more sophisticated methods, both for the whole cell model (say using Runge-Kutta approach) and for the exponential solvers. For the original Rush-Larsen scheme, higher-order variants have been proposed and tested [4], [5], and the same ideas can be extended to the matrix case as well. Naturally, HOS method may then need to involve a more sophisticated operator splitting method to correspond.

Another appealing direction for further research is application of the proposed methods to other important MC models. MRL is straightforward for any MC where TRs depend only on one variable, otherwise tabulation will be a bit more problematic. The success of the HOS approach will depend on the asymptotic properties of the TRs.

Acknowledgment

During this study, TS was funded by University of Exeter PhD Studentship and VNB was partly supported by EPSRC grant EP/I029664/1. We are grateful to C.E. Clancy for the permission to use the original authors code for this study and S. Sherwin for encouraging discussions.

References

[1] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol*, vol. 117, no. 4, pp. 500–544, 1952.

[2] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed. Boston: Brooks/Cole, 2011.

[3] S. Rush and H. Larsen, "A practical algorithm for solving dynamic membrane equations," *IEEE Trans. Biomed. Eng.*, vol. 25, no. 4, pp. 389–392, 1978.

[4] J. Sundnes, R. Artebrant, O. Skavhaug, and A. Tveito, "A second-order algorithm for solving dynamic cell membrane equations," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 10, pp. 2546–2548, 2009.

[5] M. Perego and A. Veneziani, "An efficient generalization of the Rush-Larsen method for solving electro-physiology membrane equations," *Electronic Transactions on Numerical Analysis*, vol. 35, pp. 234–256, 2009.

[6] M. E. Marsh, S. T. Ziaratgahi, and R. J. Spiteri, "The secrets to the success of the Rush-Larsen method and its generalizations," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 9, pp. 2506–2515, 2012.

[7] A. Reidman and K. Trivedi, "Numerical transient analysis of Markov models," *Comput. Opns Res.*, vol. 15, no. 1, pp. 19–36, 1988.

[8] C. E. Clancy and Y. Rudy, "Na^+ channel mutation that causes both Brugada and long-QT syndrome phenotypes: a simulation study of mechanism," *Circulation*, vol. 105, no. 10, pp. 1208–1213, 2002.

[9] C. H. Luo and Y. Rudy, "A dynamic-model of the cardiac ventricular action-potential. 1. Simulations of ionic currents and concentration changes," *Circulation Research*, vol. 74, no. 6, pp. 1071–1096, 1994.

[10] J. L. Zeng, K. R. Laurita, D. S. Rosenbaum, and Y. Rudy, "Two components of the delayed rectifier K^+ current in ventricular myocytes of the guinea-pig type. theoretical formulation and their role in repolarization," *Circulation Research*, vol. 77, no. 1, pp. 140–152, 1995.

[11] P. C. Viswanathan, R. M. Shaw, and Y. Rudy, "Effects of I_{Kr} and I_{Ks} heterogeneity on action potential duration and its rate dependence. a simulation study," *Circulation*, vol. 99, no. 18, May 1999.

[12] W. Stein *et al.*, *Sage Mathematics Software (Version 5.9)*, The Sage Development Team, 2013, http://www.sagemath.org.