Multirate method for co-simulation of electrical-chemical systems in multiscale modeling

Ekaterina Brocke[1,2,3] · Mikael Djurfeldt[4] · Upinder S. Bhalla[2] · Jeanette Hellgren Kotaleski[1,5] · Michael Hanke[6]

Received: 12 October 2016 / Revised: 21 February 2017 / Accepted: 26 February 2017 / Published online: 7 April 2017

Abstract Multiscale modeling by means of co-simulation is a powerful tool to address many vital questions in neuroscience. It can for example be applied in the study of the process of learning and memory formation in the brain. At the same time the co-simulation technique makes it possible to take advantage of interoperability between existing tools and multi-physics models as well as distributed computing. However, the theoretical basis for multiscale modeling is not sufficiently understood. There is, for example, a need of efficient and accurate numerical methods for time integration. When time constants of model components are different by several orders of magnitude, individual dynamics and mathematical definitions of each component all together impose stability, accuracy and efficiency challenges for the time integrator. Following our numerical investigations in Brocke et al. (*Frontiers in Computational Neuroscience, 10*, 97, 2016), we present a new multirate algorithm that allows us to handle each component of a large system with a step size appropriate to its time scale. We take care of error estimates in a recursive manner allowing individual components to follow their discretization time course while keeping numerical error within acceptable bounds. The method is developed with an ultimate goal of minimizing the communication between the components. Thus it is especially suitable for co-simulations. Our preliminary results support our confidence that the multirate approach can be used in the class of problems we are interested in. We show that the dynamics ofa communication signal as well as an appropriate choice of the discretization order between system components may have a significant impact on the accuracy of the coupled simulation. Although, the ideas presented in the paper have only been tested on a single model, it is likely that they can be applied to other problems without loss of generality. We believe that this work may significantly contribute to the establishment of a firm theoretical

Action Editor: D. Jaeger

Mikael Djurfeldt and Upinder S. Bhalla contributed equally to this work

Electronic supplementary material The online version of this article (doi:10.1007/s10827-017-0639-7) contains supplementary material, which is available to authorized users.

✉ Ekaterina Brocke
ekaterina.brocke@scilifelab.se

Mikael Djurfeldt
mikael@djurfeldt.com

Upinder S. Bhalla
bhalla@ncbs.res.in

Jeanette Hellgren Kotaleski
jeanette.h.kotaleski@scilifelab.se

Michael Hanke
hanke@kth.se

[1] Science for Life Laboratory, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden

[2] National Centre for Biological Sciences, Bangalore, India

[3] Manipal University, Manipal, India

[4] PDC Center for High-Performance Computing, KTH Royal Institute of Technology, Stockholm, Sweden

[5] Department of Neuroscience, Karolinska Institute, Stockholm, Sweden

[6] Department of Mathematics, School of Engineering Sciences, KTH Royal Institute of Technology, Stockholm, Sweden

basis and to the development of an efficient computational framework for multiscale modeling and simulations.

Keywords Multiscale modeling · Multiscale simulation · Co-simulation · Coupled system · Multirate integration · Adaptive time step integration · Backward differentiation formula · Parallel numerical integration · Coupled integration

1 Introduction

Multiscale modeling and simulations provide an entirely new capability for scientists to explore complex brain phenomena in neuroscience (Bhalla 2014; Hirakis et al. 2015). Multiscale modeling may contribute to understanding of how the hierarchy of biological levels (Djurfeldt and Lansner 2007) integrates to produce certain brain functions. This knowledge may have a significant impact for elucidation of disease states and drug development.

In the paper we refer to multiscale modeling as the process of defining multiple models on the conceptual, mathematical and computational levels (Delalondre et al. 2010) as well as their assembly into a larger model. Our ultimate goal is to be able to simulate these models simultaneously, in different simulation tools, while exchanging data during runtime. We refer to this methodology as co-simulation (Park and Felippa 1983; Felippa et al. 2001). In computational neuroscience the co-simulation idea is used by several software tools. For instance, the MUlti-SImulation Coordinator (MUSIC) tool promotes interoperability between different event based simulators by allowing existing models defined by different formalisms to be simulated in parallel (Djurfeldt et al. 2010).

In computational neuroscience the mathematical domain is usually represented by a particular set of mathematical formalisms among which are systems of Ordinary Differential Equations (ODEs). During the co-simulation of mutually dependent systems the coupling of the models described by different formalisms or even the same mathematical formalism is not a trivial task. One of the problems the scientist may observe is numerical artifacts such as numerical instability as well as inaccuracy. Thus the potential of a co-simulation methodology depends on the coupling method between the components, the choice of an appropriate organization strategy and the approximation of the exchange variables (Brocke et al. 2016). Moreover, the co-simulation technique imposes additional restrictions. In order to minimize the communication overhead during a simulation the exchange of data should be limited to certain points in time; an intermediate communication is prohibited.

Brocke et al. (2016) present an efficient method for coupling problems that can be formulated by systems of Ordinary Differential Equations (ODEs). In order to cope with numerical stiffness the method implements the decoupled Backward Differentiation Formulae (BDF). The integration step size is predicted by the adaptive controller in Söderlind and Wang (2006). The controller is based on the error estimation mechanism proposed and analyzed in Skelboe (2000). Brocke et al. (2016) show a significant advantage of the method both with respect to accuracy and computational cost while simulating the electrical-chemical test case. The method uses a *singlerate* coupling concept, that is it works with step sizes that are varying in time but are the same for each component per step. In contrast, in a *multirate* approach each component can be solved with its own discretization time step (Gear and Wells 1984). This can reduce the number of communication points between the components, facilitate the throughput and gain computation efficiency during co-simulation.

Skelboe (2000) mentions that the decoupled integration formula can be used in a multirate mode with a waveform relaxation method. Waveform relaxation is an iterative method. It is an expensive method and suitable only for a certain class of problems (Sand and Skelboe 1992). Sometimes it is enough to have only one iteration during the relaxation but this is only possible if a macro step size (i.e. the time step between two successive communication points) is sufficiently small. To the best knowledge of the authors, the choice of multirate methods applicable to the physical systems of our interest is limited. Bartel and Günther (2002) propose a multirate method suitable for highly integrated electric circuits that usually exhibit the variables of different time scales. This integration approach employs the concept of a constant step size integration over a varying in time macro time step. However, the results showed a relatively high amount of rejected macro steps for some stiff test cases. Another multirate method for a system of ODEs is discussed in Engstler and Lubich (1997). The method belongs to a class of extrapolation methods in which the reduction of the number of function evaluations is achieved by intermediate interpolations. These methods require additional communication between the components that can become expensive in co-simulations and should be avoided. Similar ideas are presented in Savcenco (2008).

Brocke et al. (2016) design the method mathematically justified for a singlerate approach. Here we try to relax the requirement of a singlerate integration by applying the idea of a multirate integration to gain more efficiency during co-simulations of electrical-chemical systems. The presented algorithm defines a sequence of actions to be taken while simulating mutually dependent components of a multiscale system in parallel. In particular, it introduces a new strategy for the error control in the multirate integration approach. The algorithm allows each component to be solved independently during the macro time step while keeping the accuracy withing acceptable bounds. One of the goals of the algorithm is to minimize communication between the components during a simulation. Thus it is especially tailored to suit co-simulations.

In the Section 2 we address the questions "which communication signal to choose?", "when do the components have to communicate or to be synchronized?", "in which order can the components of a multiscale system be solved?". Then we present the multirate algorithm and describe the details of its implementation in the Section 2.2. In the Sections 2.3 and 2.4 we describe the multiscale test case and an evaluation technique, respectively. Finally, we present and discuss the results in the Sections 3 and 4 sections.

2 Materials and methods

2.1 Problem definition

For expository purposes we define the problem as a system of ODEs:

$$\begin{aligned}\frac{d}{dt}x_1 &= f_1(t, x_1, g_1(x_2))\\ \frac{d}{dt}x_2 &= f_2(t, g_2(x_1), x_2), \end{aligned} \tag{1}$$

where x_1, x_2 are solution vectors of the component *1* and the component *2* respectively. g_1 and g_2 are transformation functions of the respective solution vector (or a part of it). In the simplest case the transformation function may represent a uniform scaling operation between the variables.

2.1.1 Communication signal

Coupling the components of a multiscale system involves decisions about communication signals. The choice can be crucial not only from the conceptual perspective, but it may also have a significant impact on the computational side of a simulation, e.g., with respect to efficiency.

In neuroscience, the cellular level is a level of communications, signal integration and filtering that usually acts on a timescale of about a few microseconds. In the context of synaptic plasticity, learning and memory, the subcellular level covers long term processes of physiological changes in the cell that can span timescales from a few seconds to months and years. How does the dynamics of a communication signal contribute to the numerical properties and overall simulation performance? Which time course should a communication signal follow in this case?

In Brocke et al. (2016) we developed a multiscale test case where the fast electrical signal is scaled and communicated to the biochemical component. From a computational perspective, the fast changing signal can introduce stiffness in the whole coupled system (Gear and Wells 1984). However, if stiffness is isolated in a separate component, it can be tackled locally by choosing an appropriate integration method. Consequently the choice of having an approximated communication signal with time course of a slower component seems to be more favorable unless there is a need to model systems on a very small spatial scale.

2.1.2 Organization of system components

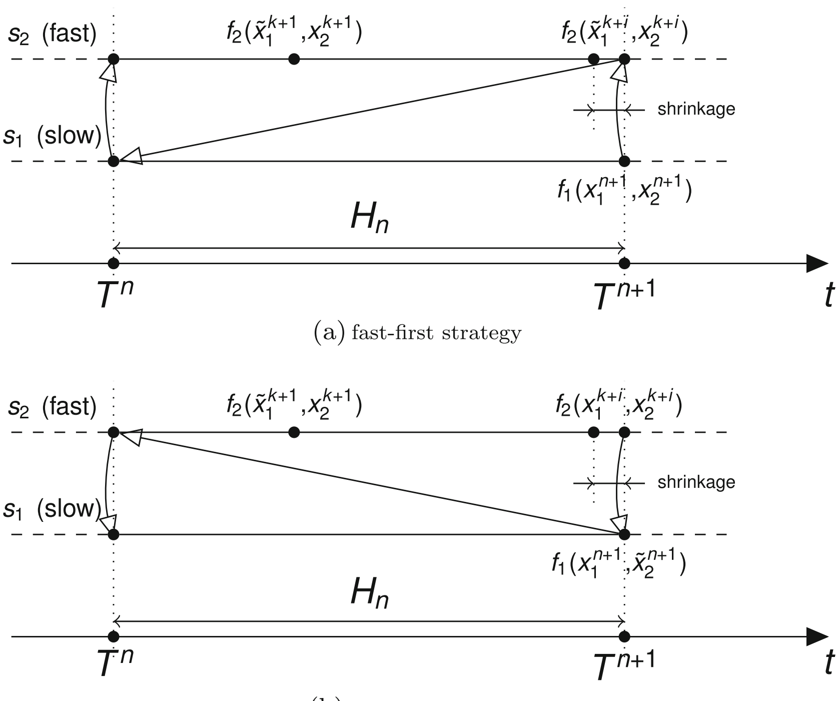
In co-simulations the components can be solved using either *Jacobi* or *Gauss-Seidel* organization between the components (Brocke et al. 2016). The organization of the components will define which approximation, interpolation, extrapolation, or both, shall be used. The choice may have a crucial impact on the accuracy of the coupled simulation. In general, during a multirate integration when approximation is required, it is preferable to choose interpolation instead of extrapolation over a large integration step. Gauss-Seidel organization allows us to eliminate the error introduced by the approximation of one component as described in Brocke et al. (2016). Therefore, we will consider only Gauss-Seidel organization with the fast component or the slow component solved first. We will refer to these as fast-first and slow-first strategies respectively.

Figure 1 shows the discretization of two components over one macro time step H_n for the fast-first (Fig. 1a) and slow-first (Fig. 1b) strategies. The slow system values x_1^{k+l}, needed at each micro time step $k+1 \dots k+i$, are not known. Then depending on the strategy either an extrapolation or interpolation can be used to calculate $\tilde{x}_1^{k+1}$.

Each strategy has its pros and cons. In the fast-first strategy when a macro time step is rejected at least some of the computations performed at the micro time steps will be discarded. In this case it is crucial to use a well adjusted step size controller in order to minimize the number of rejected steps. In contrast, when a slow component is solved first the memory usage and computation costs are relatively negligible. However, an extrapolation over a large macro step can introduce a significant extrapolation error. In principle, the error can be mitigated when the coupling from a fast component to a slow component is weak (Sand and Skelboe 1992). We expect that this can be accomplished by using a slow changing communication signal in our test case.

2.1.3 Communication time points

Co-simulation usually imposes the problem of the variables' availability in the mutually dependent components. Moreover, in a multirate integration approach there will be points on the time grid when not all required variables are available due to the independent discretization of the components. With *synchronous communication* we refer to a


s_2 (fast)
$f_2(\tilde{x}_1^{k+1}, x_2^{k+1})$
$f_2(\tilde{x}_1^{k+i}, x_2^{k+i})$
shrinkage
s_1 (slow)
$f_1(x_1^{n+1}, x_2^{n+1})$
H_n
T^n
T^{n+1}
t
s_2 (fast)
$f_2(\tilde{x}_1^{k+1}, x_2^{k+1})$
$f_2(x_1^{k+i}, x_2^{k+i})$
shrinkage
s_1 (slow)
$f_1(x_1^{n+1}, \tilde{x}_2^{n+1})$
H_n
T^n
T^{n+1}
t

(a) fast-first strategy

(b) slow-first strategy

Fig. 1 Discretization of two components s_1 and s_2 over one macro time step $[T^n, T^{n+1}]$. f1(...) and f2(...) are the function evaluations required to evolve the system. The *arrows* correspond to the communication of exchanged variables. The variables marked with a *tilde* denote variables obtained either by extrapolation or interpolation. In the fast-first strategy (1a) the slow variables x_1^n are communicated first, then for each micro time step the variables $\tilde{x}_1^{k+i}$ are obtained by extrapolation x_1^n at each t_2^{k+i}. The calculated solution x_2^{k+i} at T^{n+1} is communicated backwards. Finally, the slow component evolves to T^{n+1}. In the slow-first strategy (1b) the sequence of computations is similar. The difference is in how the exchanged variables of the slow component are approximated at each micro time step. $\tilde{x}_1^{k+i}$ can be obtained by interpolation since it is the slow component that makes the first step. Both figures show a synchronous communication between the components where the last micro time step $[t_2^{k+i-1}, t_2^{k+i}]$ is shrunk (see the discussion in Section 2.1.3). This interval is marked with *two arrows* and the label "shrinkage"

communication protocol organized in a such way that the time points of the discretization grid of any component is a subset of the grids of the faster components. In contrast, in *asynchronous* communication, the discretization time grids of the components are not synchronized as depicted in Fig. 2. It is worth noting that the latter type of communication requires an additional interpolation step.

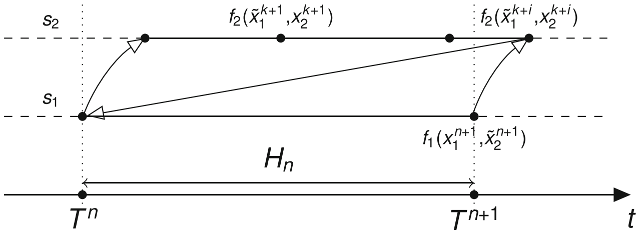

s_2
$f_2(\tilde{x}_1^{k+1}, x_2^{k+1})$
$f_2(\tilde{x}_1^{k+i}, x_2^{k+i})$
s_1
$f_1(x_1^{n+1}, \tilde{x}_2^{n+1})$
H_n
T^n
T^{n+1}
t

Fig. 2 Discretization of two components s_1 and s_2 with asynchronous communication over one macro time step $[T^n, T^{n+1}]$. The *arrows* correspond to the communication of exchanged variables in the fast-first integration strategy. The variables marked with a *tilde* denote variables obtained by extrapolation. Asynchronous communication requires an interpolation of x_2^{k+i} at T^{n+1} that is denoted by $\tilde{x}_2^{n+1}$ in the figure

Several tactics are known for synchronous communication. For instance, one strategy is to use a fixed step size ratio between the micro and macro time steps (Bartel and Günther 2002). One of the potential problems with this approach is a relatively high amount of rejected macro time steps. Another option can be to shrink the last micro time step at the end of each macro time step as indicated in Fig. 1. However, a sharp reduction of the step size can cause inefficiency of the step size controller that will require a number of additional small steps to be taken before the step size will be increased again. Moreover, it was shown that the BDF method has better stability properties with a smoother behavior of step sizes (Hairer et al. 2010, pp. 402ff). The latter was the key argument that inclined us towards the choice of asynchronous communication.

2.2 Multirate algorithm

Here we give a detailed description of the multirate algorithm. Let T be an integration interval and S an ordered set of p components of a system to be integrated over T. Then the system can be solved by the SOLVE_INTERVAL(T,S) function. A sequence of necessary operations is presented in Algorithm 1 together with the following explanations:

(1) The algorithm admits a possible variance in the temporal dynamics of a component that may appear due to an external signal, for instance. The defined set S has the order slowest to fastest. A fast component is the one with the smallest predicted integration time step while a slow component is the one with the largest. Thus, the decision which component is fast and which is slow can be made automatically during the integration process. The order of components can be assigned by a simple algorithm like *insertion sort* before each call to the SOLVE_INTERVAL(T,S) function.

    In the problem definition Eq. (1) we have two components $p = 2$, slow s_1 and fast s_2, respectively. Note, that the notation f_i introduced in Eq. (1) denotes the physics of a multiscale system while s_i is used to distinguish between fast and slow components in the algorithm.

(2) For each current component s_r an integration time interval T is provided by the consecutive slower component. For the outermost call of SOLVE_INTERVAL the variable T is equal to the next predicted time step of the first component in the set S, that is the largest predicted time step among all components.

(3) Different numerical approximation methods could be applied to each component to handle individual dynamics in an efficient way. Here, we choose to use one method for all components in the system. We implement the Backward Differential Formulae of the second order (BDF2) due to its good stability properties, high applicability for stiff differential equations and analyzed properties in coupled simulations (referred to as decoupled BDF2 formulae in Skelboe (2000)). The SOLVE function calculates a numerical solution at t_r^{n+1} of the current component s_r using the BDF2 method (see Section 2.2.1 for details).

(4) A set of exchanged variables $\{\tilde{x}_1^{n+1}, \ldots, \tilde{x}_q^{n+1}\}$ of the components $\{s_1, \ldots, s_q\}$ at time t^{n+1} has to be provided to the SOLVE function.

    In the slowest-first strategy the subset $\{\tilde{x}_1^{n+1}, \ldots, \tilde{x}_{r-1}^{n+1}\}$ is obtained by interpolation at t_r^{n+1}; while the subset $\{\tilde{x}_{r+1}^{n+1}, \ldots, \tilde{x}_q^{n+1}\}$ is obtained by extrapolation at t_r^{n+1}.

    In the fastest first strategy with synchronous communication between the components only extrapolation is required to obtain the $\{\tilde{x}_1^{n+1}, \ldots, \tilde{x}_{r-1}^{n+1}\}$ subset since the exchanged variables from the faster components are calculated first. In asynchronous communication between the components an interpolation can be used to calculate the subset $\{\tilde{x}_{r+1}^{n+1}, \ldots, \tilde{x}_q^{n+1}\}$.

    Note that for notational simplicity, we omitted the transformation function g that was introduced in Eq. (1). The exchanged variables usually represent a result of the transformation function g that can be as simple as a subset of the variables of a solution vector x.

(5) An estimation of the local discretization error $|[\epsilon_r]|$ is calculated in the ESTIMATE_ERROR function (line #9). Each component calculates an estimation of the local discretization error using a second order polynomial predictor (Brocke et al. 2016). More details are given below in the Section 2.2.2 section.

(6) The PREDICT function implements a step size controller that predicts an optimal next time step. We use the step size controller implemented in Brocke et al. (2016).

Algorithm 1 Multirate algorithm with the fastest first strategy

```
 1: function SOLVE_INTERVAL($T$, $S = \{s_r \ldots s_q\}, \ldots$)
 2:     if $S = \{\}$ then
 3: % The function was called by the fastest component
            return {}, 0
 4:     end if
 5: % Retrieve the end of the integration time interval
        $t_{end} \leftarrow T(2)$
 6: % While the integration time of the current
    % component $s_r$ does not step over $t_{end}$
        while $t_r^n < t_{end}$ do
 7:         $X, \epsilon_{max_{r+1}} \leftarrow$ SOLVE_INTERVAL($[t_r^n, t_r^{n+1}]$, $S \backslash \{s_r\}, \ldots$)
 8:         $x_r^{n+1} \leftarrow$SOLVE($s_r$, $[t_r^n, t_r^{n+1}]$, $\{\tilde{x}_1^{n+1}, \ldots, \tilde{x}_{r-1}^{n+1}\} \cup X$)
 9:         $|[\epsilon_r]| \leftarrow$ESTIMATE_ERROR
10:         if $|[\epsilon_r]| \leq TOL$ then
11:             $\epsilon_{max_r} \leftarrow \max(\epsilon_{max_r}, \epsilon_{max_{r+1}}, |[\epsilon_r]|)$
12: % Advance current integration time
                $t_r^n \leftarrow t_r^{n+1}$
13:         end if
14:         $t_r^{n+1} \leftarrow$PREDICT($\max(\epsilon_{max_{r+1}}, |[\epsilon_r]|), \ldots$)
15:     end while
16:     return $\{\tilde{x}_r^{n+1}\} \cup X, \epsilon_{max_r}$
17: end function
```

Figure 3 shows an example of the communication pattern between three components s_1, s_2 and s_3 implemented by the Algorithm 1. The algorithm assumes an all-to-all connectivity between the components in the system. An optimization for non all-to-all cases has yet to be considered.

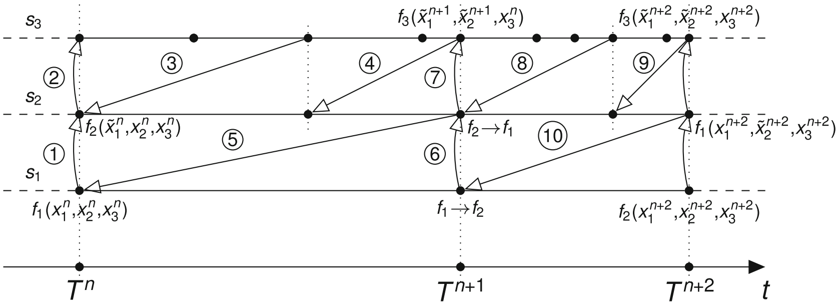

s_3
s_2
s_1
$f_3(\tilde{x}_1^{n+1},\tilde{x}_2^{n+1},x_3^n)$
$f_3(\tilde{x}_1^{n+2},\tilde{x}_2^{n+2},x_3^{n+2})$
$f_2(\tilde{x}_1^n,x_2^n,x_3^n)$
$f_2 \rightarrow f_1$
$f_1(x_1^{n+2},\tilde{x}_2^{n+2},x_3^{n+2})$
$f_1(x_1^n,x_2^n,x_3^n)$
$f_1 \rightarrow f_2$
$f_2(x_1^{n+2},x_2^{n+2},x_3^{n+2})$
1
2
3
4
5
6
7
8
9
10
T^n
T^{n+1}
T^{n+2}
t

Fig. 3 The sequence of communications realized by the fast-first integration strategy with synchronous communication between three components s_1, s_2 and s_3 over two macro time steps $[T^n, T^{n+1}]$ and $[T^{n+1}, T^{n+2}]$. The arrows correspond to the communication of exchanged variables. The order of communications is denoted by the circled numbers. After the first macro time step the second subsystem was recognized as the slowest component. Thus the order of integration has been changed (denoted with $f_1 \rightarrow f_2$ and $f_2 \rightarrow f_1$, respectively)

2.2.1 Backward differentiation formula methods

Here we are addressing a class of problems that is described by a system of ODEs with a given initial value (initial value problem (IVP)). This class of problems is usually solved by finite difference time-stepping methods. We use the Backward differentiation Formula (BDF) methods. These methods were introduced by Curtiss and Hirschfelder in Curtiss and Hirschfelder (1952). The Backward differentiation formulas with an order less than three are often used for stiff systems due to their property of being A-stable. We are interested in the second-order backward differentiation formula (BDF2). On an equidistant grid the BDF2 formula has the form:

$$\frac{2}{3}x_{n+1} - 2x_n + \frac{1}{2}x_{n-1} = hf_{n+1}(t_{n+1}, x_{n+1}) \tag{2}$$

On a non-uniform grid, the BDF2 method can be formulated as:

$$x_{n+1} = \alpha_1 x_n + \alpha_2 x_{n-1} + \beta h_{n+1} f(t_{n+1}, x_{n+1}) \tag{3}$$

where

$$\gamma_{n+1} = h_{n+1}/h_n \tag{4}$$

$$\alpha_1 = 1 - \alpha_2 \tag{5}$$

$$\alpha_2 = -\gamma_{n+1}^2/(2\gamma_{n+1} + 1) \tag{6}$$

$$\beta = (\gamma_{n+1} + 1)/(2\gamma_{n+1} + 1) \tag{7}$$

2.2.2 Error control

To ensure that the exact solution is adequately approximated and the local error is within reasonable bounds we implement an error control mechanism of the coupled system during the simulation. Brocke et al. (2016) present an estimate of the local discretization error based on a component-wise estimation for the solution vector of the whole system at each discretization time step in a singlerate integration approach. The choice was justified by the theoretical background provided in Skelboe (2000). Analyzing the test case developed in Brocke et al. (2016) we noticed that it was the local discretization error of the fastest component that constrained the size of the discretization time step for the larger part of the simulation time. Bearing this in mind we developed an idea how a multirate integration can be achieved when the discretization error of one of the components dominates. We relax the necessity of a singlerate integration by allowing the faster component to have a step size controller independent from the slower component; whereas we constrain the step size controller of the slower component by taking the maximum discretization error between the local error and the maximum discretization error of the faster components calculated over the current step. This strategy is depicted by lines# 11- 14 of Algorithm 1.

Given a set of ordered components $S = \{s_r \ldots s_q\}$ in the fast-first organization strategy, the discretization error at t_r^{n+1} of the current component s_r can be written in the form:

$$\left|\epsilon_r^{n+1}\right| = \max\left\{\max_i \frac{\left|x_{r,i}^{n+1} - \hat{x}_{r,i}^{n+1}\right|}{relTOL_r \cdot \left|x_{r,i}^{n+1}\right| + absTOL_{r,i}},\right.$$
$$\left.\max_{m\in[k,k+j]} \left|\epsilon_{r+1}^m\right|\right\} \tag{8}$$

The first argument in the braces corresponds to a component-wise calculation of the local discretization error of the s_r component. We use a predictor-corrector algorithm where $\hat{x}_r^{n+1}$ is a predicted solution and x_r^{n+1} is an approximated solution at t_r^{n+1}. For the former we used a second order polynomial predictor. Details can be found in Brocke et al. (2016). $relTOL_r$ and $absTOL_r$ are input tolerance parameters. The second argument denotes the maximum discretization error of the faster component s_{r+1} over an integration interval $[t_{r+1}^k, t_{r+1}^{k+j}]$. In synchronous communication between the components the interval is equal to the time step over $[t_r^n, t_r^{n+1}]$ of the current component. The interval overlaps with the latter time step in an asynchronous communication as shown in Fig. 2.

2.3 Test case

In the scope of interest are the models that are usually formulated on the cellular and subcellular levels of neuronal organization where the molecular events closely interact with the electrical ones and give rise to many levels of physiological change like synaptic plasticity and changes in dendritic excitability.

Here we exploit the multiscale test case used in Brocke et al. (2016). The test case was developed with two goals in mind: to keep complexity of the system within reasonable computational bounds, and to implement a moderately realistic behavior that will cover a class of problems among which synaptic and cellular plasticity are of particular interest. The electrical component of the system represented a multi-compartmental neuron with the minimal Hodgkin-Huxley type model for a regular-spiking neuron (Pospischil et al. 2008). During the simulations a pulse current protocol was applied to maintain the regular spiking behavior at approximately 2 Hz and the burst firing at approximately 100 Hz for 5 seconds of simulation time. The biochemical component of the system represented a reduced model of the Mitogen-Activated Protein Kinase (MAPK) signaling pathway previously used in the study of homeostatic regulation of excitability at the scale of a single synapse (Bhalla 2011). The biochemical component models the signaling events triggered by the electrical activity of the neuronal membrane. In particular, it is the calcium signal that feeds into a slowly integrating signaling pathway and initiates the chemical cascades. A schematic representation of the models and their interaction is shown in Fig. 4.

In the implementation of Brocke et al. (2016) the calcium current is transformed into a flux of calcium molecules and then communicated to the biochemical component. In this case the communication signal follows the fast dynamics of the electrical system.

To investigate how the dynamics of the communication signal can influence the efficiency of a coupled integration, we introduce the following modifications to the test case. We solve calcium concentration on the electrical side and then communicate it to the biochemical component. The signal from the biochemical component to the electrical one, that is the fraction of active (non-phosphorylated) potassium channels in the spine, remains unchanged. We will refer to the original test case that has a fast changing communication signal, that is calcium flux, as *TC-fast*, and to the modified version with the slowly changing exchanged variable, that is calcium concentration, as *TC-slow*. In Fig. 5 we plot both communication signals for visual comparison: calcium flux and calcium concentration. It is clear that the dynamics of the calcium concentration is much slower than the dynamics of the flux. Here, microseconds are the characteristic time for the flux, and milliseconds for the concentration.

The morphological and physiological details of the test case *TC-slow* is given in Appendix A.

This modification of the communication signal leads to a *linear* dependence between the components in the problem (1):

$$\begin{aligned}\frac{d}{dt}x_{bioch} &= f_{bioch}(t, x_{bioch}, g_1(x_{elec}))\\ \frac{d}{dt}x_{elec} &= f_{elec}(t, g_2(x_{bioch}), x_{elec}),\end{aligned} \tag{9}$$

where g_1 and g_2 are the output functions from the electrical and the biochemical component respectively.

$$\begin{aligned}g_1(x_{elec}) &= x_{elec,i}\\ g_2(x_{bioch}) &= C_1 x_{bioch,j}\end{aligned} \tag{10}$$

where C_1 is a constant; the indices i, j correspond to positions of the variables in the corresponding solution vector x at time t. In particular, i is the calcium concentration in the spine [M] and j is the concentration of active (non-phosphorylated) potassium channels in the spine [M].

2.4 Evaluation and implementation

In this paper we evaluate the proposed multirate algorithm using the multiscale test cases *TC-fast* and *TC-slow*. In order to gain understanding of the global error propagation in the coupled simulation, we should choose the most representative solutions from both electrical and biochemical components.

The phosphorylated MAPK (*P-MAPK*) and active potassium (*K_A*) molecules are the ones that bring about the bistability property of the biochemical system. We measure the accuracy by calculating the relative error of the solution at $T = 2\ s$ of the simulation time:

$$e_i = \frac{|\check{x}_{T,i} - x_{T,i}|}{\check{x}_{T,i}} \cdot 100 \quad [\%], \tag{11}$$

where $\check{x}_{T,i}$ is the reference solution at time T and $\check{x}_{T,i}$ is the chosen solution at time T simulated with the analyzed method. In our simulations we choose to evaluate the voltage solution in the spine (V), calcium solution (Ca) and either potassium(K_A) or *P-MAPK* solutions in our test cases.

The test case, the algorithms and numerical methods were implemented in the MATLAB® environment to attain a full control over numerical integration. The reference

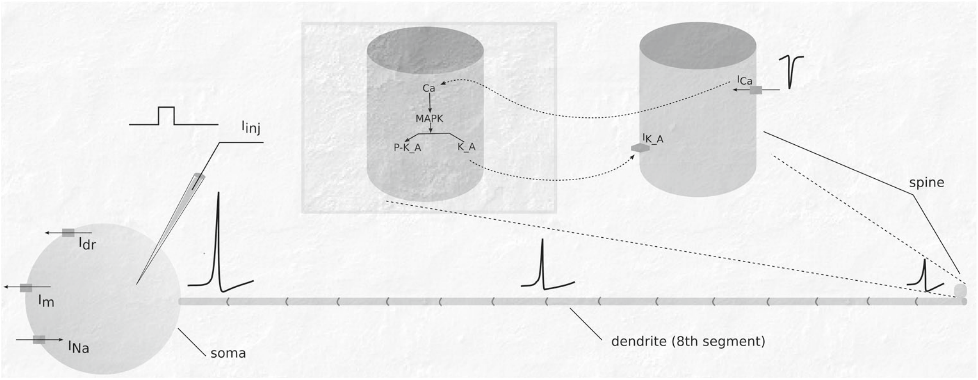


Fig. 4 A schematic representation of the multiscale neuron assembly. The compartmental structure of the neuron is shown in terms of a soma, 15 segments of a dendrite and a spine. The spine segment is zoomed in for visual convenience. The MAPK model (biochemical component) is placed in the spine segment and is marked with a distinct rectangle. Communication signals between electrical and biochemical components are shown by the curved arrows. The response time of the MAPK model is in the range of tens of seconds (Bhalla 2011), while the response of the electrical model is the order of a microsecond

solution for each test case is acquired by solving both components as a single system using the $ode15s$ MATLAB® function with a tight value of the tolerance parameter ($tol = 10^{-12}$).

The sources containing the algorithms and model implementations as well as the datafiles to produce the Figs. 6–8 are available at https://github.com/ebrocke/multiscale.

3 Results

Each figure in this section represents the measurements of the solution accuracy versus the number of ODE calls. We consider the number of ODE calls as a measure of the computational cost for the solver. We plot both first- and second-order declines in dashed lines in order to have visual estimates of the order of the coupling method. The singlerate coupling strategy in Brocke et al. (2016) showed a second order accuracy and is taken as a reference for evaluation of the multirate method in this paper. In addition, by comparing the curves vertically it is possible to estimate the accuracy of the solution for the same amount of

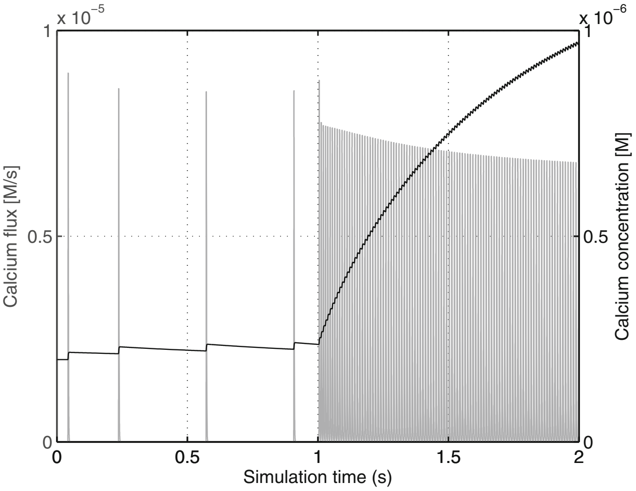


Fig. 5 The communication signal from the electrical to biochemical component. Calcium flux (used in *TC-fast*) is shown in a *grey color* while calcium concentration (used in *TC-slow*) is shown in *black*. Calcium concentration exhibits a much smoother behavior

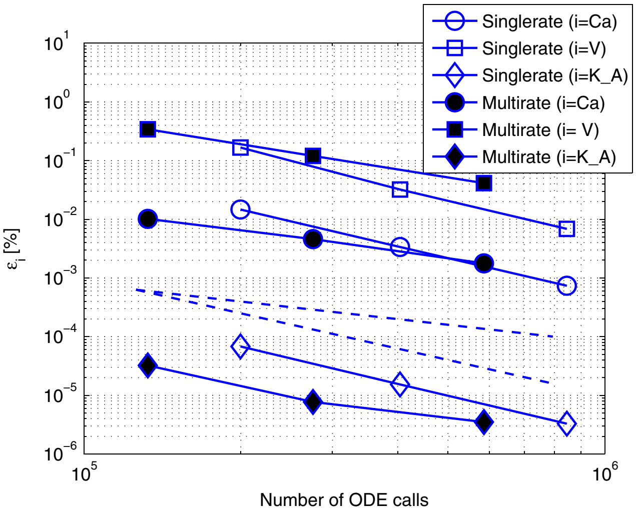


Fig. 6 Efficiency comparison between single- and multirate approaches in the BDF2-BDF2 coupling method with the fast communication signal. The singlerate coupling method corresponds to the results presented in (Brocke et al. 2016). The datapoints in the figure correspond to $relTOL = \{10^{-5}; 10^{-6}; 10^{-7}\}$. The *dashed lines* correspond to first- and second-order declines. The simulations were performed with the *TC-fast* test case that uses calcium flux as a communication signal between the electrical and biochemical component. The accuracy results of the singlerate integration using Gauss-Seidel organization where the electrical component was solved first is represented by hollow markers. The results of a multirate integration with Gauss-Seidel organization and the fast-first strategy are shown with filled markers. Exchanged variables were approximated by a second order polynomial (referred to as *Mode 3* in (Brocke et al. 2016))

computational cost. Looking at the curves horizontally, the computational costs for the same accuracy can be compared. Each curve is an interpolation on a set of datapoints marked with markers. Each datapoint corresponds to the specified tolerance parameter used in the simulation.

3.1 Multirate improves efficiency but accuracy is lost with a fast communication signal

Second order accurate numerical solutions are expected using the BDF2-BDF2 coupling method since this was found with the *TC-fast* test case singlerate simulations in Brocke et al. (2016). To investigate the multirate algorithm's performance we simulated the *TC-fast* test case with different values of the tolerance parameter. We plotted the accuracy measurements of the solutions versus the number of ODE calls in Fig. 6.

In the multirate simulations we observed a loss of order of accuracy in the *TC-fast* test case that is clearly noticeable while comparing the interpolated error behavior with the decline lines plotted in the figure. Presumably the dominant extrapolation error at each micro time step propagates to the slow component through the fast communication signal.

Comparing corresponding datapoints in Fig. 6, a large difference in the number of ODE calls between single- and multirate coupling approaches can be observed. For a detailed numerical overview of this difference we present the synopsis of the simulations in Table 1. The speed ups both with respect to the number of function calls and computation times in the multirate approach were substantial.

3.2 The extrapolation error influence in the fast-first strategy is reduced by a slow changing communication signal

Can we diminish the influence of the expected extrapolation error observed in the previous results? To answer this question we replaced the fast communication signal with the slow one as described in Section 2.3. Here, we simulate the *TC-slow* test case and compare single- and multirate coupling approaches in Fig. 7.

Comparing Figs. 6 and 7 we observed that the extrapolation error in the fast component ($i = V$, the voltage error lines) was not changed. However, the error in the slow component ($i = K\_A$, the potassium error lines) was not only considerably smaller but, also, the second order accuracy was re-established. While the first observation was in accordance with our expectations due to required extrapolation in the fast-first strategy step, the latter was slightly surprising. We compared the order of accuracy of the communication signal ($i = Ca$, the calcium error lines). The results showed that the accuracy of the communication signal, that is calcium concentration solved in the fast component, had been re-established to second order.

3.3 Slow-first strategy restores the accuracy of the coupled multirate simulation

Our previous results showed that the second order of the BDF2-BDF2 coupling using the multirate integration approach was reduced to first order. The simulations were performed using the fast-first strategy where the variables communicated from the slow component had to be extrapolated at each micro time step. There, we suspected that an extrapolation error of the exchanged variables had the main contribution to the numerical approximation error at each discretization time step. After, instead, using a slow communication signal (the *TC-slow* test case) and verifying that the propagation of the approximation error can be diminished we applied the slow-first strategy. In that case an extrapolation of a slow varying signal may have less significant contribution to the discretization error of a slow component. Figure 8 confirms our expectations. Second order accuracy of the multirate algorithm was confirmed using the slow-first strategy and a slow communication signal between the components.

Table 1 Synopsis of the *TC-fast* test case simulations using single- and multirate approaches

relTol	micros	macros	speedup 1	speedup 2	# of switches
1e−5 (singlerate)	90529(4081)	–	–	–	–
1e−5 (multirate)	90543(4266)	23747(16)	1.5	3.4	26
1e−6 (singlerate)	194883(2915)	–	–	–	–
1e−6 (multirate)	193630(1822)	70244(74)	1.47	3.	81
1e−7 (singlerate)	417315(2211)	–	–	–	–
1e−7 (multirate)	416832(1922)	159361(89)	1.44	2.7	133

micros - the number of the micro time steps in case of the multirate or the total number of accepted steps otherwise; macros - the number of the macro time steps; speedup 1 - the speed up measurements based on the number of function calls; speedup 2 - the speed up measurements based on a wallclocktime; # of switches - the number of times when the integration order of the components has been changed. The numbers in parenthesis denote the number of rejected steps during the simulation

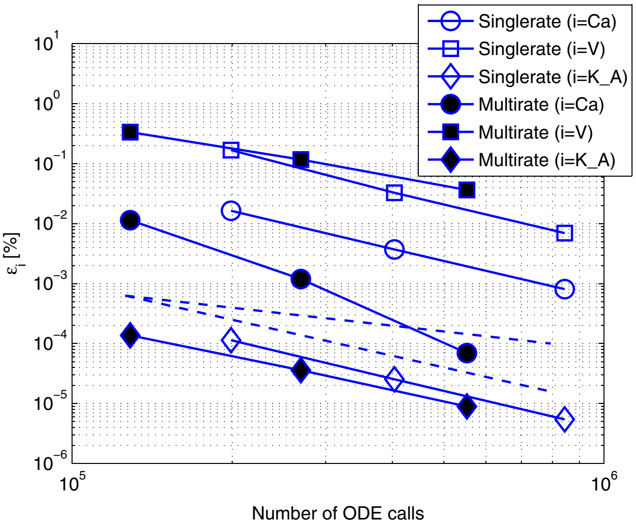


Fig. 7 Efficiency comparison between single- and multirate approaches in the BDF2-BDF2 coupling method with the slow communication signal. The datapoints in the figure correspond to $relTOL = \{10^{-5}; 10^{-6}; 10^{-7}\}$. The simulations were performed with the *TC-slow* test case which has a slow changing communication signal (calcium concentration) between the electrical and biochemical component. Otherwise conditions are as in Fig. 6. Hollow markers: singlerate integration with Gauss-Seidel organization, electrical component solved first. Filled markers: the results of the multirate integration with the fast-first strategy

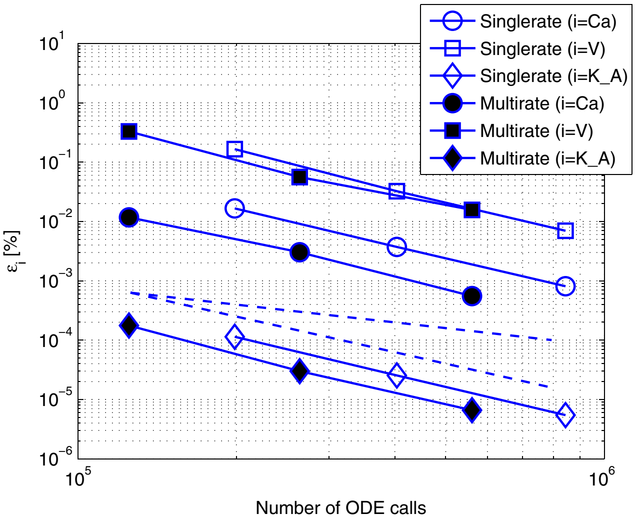


Fig. 8 Efficiency comparison between single- and multirate approaches in the BDF2-BDF2 coupling method using slow-first strategy and a slow communication signal. The datapoints in the figure correspond to $relTOL = \{10^{-5}; 10^{-6}; 10^{-7}\}$. The simulations were performed with the *TC-slow* test case. Gauss-Seidel organization with the biochemical component solved first and slow-first strategy were used in single- and multirate approaches, respectively

4 Discussion

In this paper we presented a new multirate numerical integration algorithm based on the BDF2-BDF2 coupling method described in Brocke et al. (2016). The idea of multirate numerical integration is to let individual components be solved independently for a large integration time in order to gain efficiency by minimizing communication and computation operations. The proposed multirate algorithm showed computational speedups both with respect to the number of function calls and computation times in simulations of a given test case.

We presented and analyzed two organization strategies between the components, fast-first and slow-first. The simulations showed a strong dependence of the numerical accuracy of the coupled simulations on the dynamics of the communication signal between the components. We observed that in multirate simulations the second order of the BDF2-BDF2 coupling method was reduced to the first order in the fast-first strategy. We introduced a few modifications to our multiscale test case. In particular, we replaced the fast communication signal with its analogous slow changing variable. The results showed a strong numerical sensitivity of a slow system to the behavior of the communication signal. As can be seen in Fig. 7, the increased accuracy of the slow communication signal led to the re-establishment of the accuracy order of the slow component. Note, that a coupled method may have properties completely different from those of the individual components (Hanke 2017). It is known that the order of a method may depend on linearity and non-linearity of the system (Hairer et al. 1993). Considering that the slow variable had been defined by a linear ODE in comparison to its highly nonlinear definition in the initial test case, it might be one of the possible explanations for the re-establishment of the order in Fig. 7. In addition, we suspect that the stability domain of the coupled BDF2 method may change similar to what is observed in Hanke (2017). Some methods are known to have a reduced order if applied to very stiff problems (Hairer and Wanner 1996). It might happen that the smoothened variable drives the system away from stiffness thus changing the order. However, these ideas do not yet have any theoretical justification with respect to the coupled BDF2 methods.

The re-established order of the communication signal allowed us to employ a slow-first Gauss-Seidel organization strategy where the extrapolation error of the slow changing variable had a less severe impact on the accuracy of the coupled simulation thus making the slow-first organization strategy more favorable.

We decided to use calcium as a key communication signal between electrical and chemical components in the test case. From the perspective of cellular physiology it is clear that by far the most commonly reported and predominant signal for electrical to chemical coupling is indeed calcium. In our simulations we replaced the fast calcium flux communication signal with a smoother signal (calcium concentration). A few studies have estimated coupling through synaptic transmission via metabotropic G-protein coupled receptors (Bhalla and Iyengar 1999), but here flux calculations are not needed and instead the arrival of an action potential is typically treated as an instantaneous event that raises the activation of the receptor. Downstream events are slow (hundreds of ms) so this component of coupling is not likely to contribute significantly to the error. Neuronal simulation packages such as NEURON, GENESIS, MOOSE offer to solve chemical concentrations along with the fast electrical calculations. Here we represented calcium calculations by a phenomenological model of the calcium activity on the electrical side of the system. The results presented in the paper will remain valid unless a study addresses calcium behavior on a finer compartmentalization level. In this case a more detailed modeling of calcium and its interactions is required where calcium may not be viewed as a slow changing variable anymore.

An important biochemical aspect related to the chemical signal is diffusion. In spatially complex models, the chemical signals may require reaction-diffusion formulations. This introduces spatial discretization requirements for the chemical system and a further numerical integrator to solve the Partial Differential Equations (PDEs) for diffusion. Our current analysis does not address how to couple the chemical reaction calculations both to the chemical diffusion and electrical calculations.

The multirate algorithm developed in this paper is based on the BDF2-BDF2 coupling method. Solver implementations based on the BDF family of methods can be found in multiple scientific open source software packages which can easily be incorporated into existing simulation codes. For example, the NEURON simulation environment supports adaptive integration by means of the SUNDIALS CVODE suite (Serban and Hindmarsh 2005). The latter includes a variable-order variable-stepsize BDF-based method.

For purposes of evaluating the multirate algorithm we simulate the developed test case on a range of low tolerances. Our basic assumption is that the underlying system of ODEs is perfect, that is it depicts the real system with high fidelity. In reality, the knowledge of a system is usually constrained by some unknowns, for example, with regard to parameters. Thus it may turn out that it is not necessary to perform simulations with high levels of numerical precision. Therefore, once the method is designed well, other optimizations may be considered in order to achieve more efficient simulations. One possibility is to take different sources of error into consideration rather than to focus only on the numerical error. Rangan and Cai (2007) present a method that accounts for both *trajectory-wise* and *statistical* accuracies in simulations of large-scale integrate-and-fire neuronal networks. In neuronal network simulations it is usually more important to obtain accurate statistical properties of the network rather than the individual time course for each of the neurons. The emphasis of statistical properties in the work by Rangan and Cai (2007) has a close connection to uncertainty quantification that is often posed in a statistical manner. This analysis extends much beyond the scope of this paper but can be considered for further development of efficient coupling algorithms for electrical-chemical systems.

This paper presents a new multirate numerical integration algorithm developed with a particular interest in co-simulations of electrical-biochemical systems in neuroscience. The ideas and requirements of the implementation presented in the paper can be exploited in the development of the API in communication frameworks such as the MUlti-SImulation Coordinator tool (MUSIC) (Brocke and Djurfeldt 2011; Djurfeldt et al. 2010).

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 604102 (HBP), the European Union's Horizon 2020 research and innovation programme under grant agreement no 720270 (HBP), the Swedish Research Council, NIAAA (grant 2R01AA016022), Swedish e-Science Research Center, and an Erasmus Mundus Joint Doctoral program EuroSPIN.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

Bartel, A., & Günther, M. (2002). A multirate W-method for electrical networks in state–space formulation. *Journal of Computational and Applied Mathematics*, *147*(2), 411–425.

Bhalla, U.S. (2011). Multiscale interactions between chemical and electric signaling in ltp induction, ltp reversal and dendritic excitability. *Neural Networks*, *24*(9), 943–949.

Bhalla, U.S. (2014). Multiscale modeling and synaptic plasticity. *Progress in Molecular Biology and Translational Science*, *123*, 351–386.

Bhalla, U.S., & Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, *283*(5400), 381–387.

Brocke, E., & Djurfeldt, M. (2011). Efficient spike communication in the MUSIC multi-simulation framework. *BMC Neuroscience*, *12*(Suppl 1), P79.

Brocke, E., Bhalla, U.S., Djurfeldt, M., Hellgren Kotaleski, J., & Hanke, M. (2016). Efficient integration of coupled electrical-chemical systems in multiscale neuronal simulations. *Frontiers in Computational Neuroscience*, *10*, 97. doi:10.3389/fncom.2016.00097. http://journal.frontiersin.org/article/10.3389/fncom.2016.00097.

Curtiss, C., & Hirschfelder, J.O. (1952). Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America*, *38*(3), 235.

Delalondre, F., Smith, C., & Shephard, M.S. (2010). Collaborative software infrastructure for adaptive multiple model simulation. *Computer Methods in Applied Mechanics and Engineering*, *199*(21), 1352–1370.

Djurfeldt, M., & Lansner, A. (2007). Workshop report: 1st INCF workshop on large-scale modeling of the nervous system.

Djurfeldt, M., Hjorth, J., Eppler, J.M., Dudani, N., Helias, M., Potjans, T.C., Bhalla, U.S., Diesmann, M., Kotaleski, J.H., & Ekeberg, Ö. (2010). Run-time interoperability between neuronal network simulators based on the MUSIC framework. *Neuroinformatics*, *8*(1), 43–60.

Engstler, C., & Lubich, C. (1997). Multirate extrapolation methods for differential equations with different time scales. *Computing*, *58*(2), 173–185.

Felippa, C.A., Park, K., & Farhat, C. (2001). Partitioned analysis of coupled mechanical systems. *Computer methods in Applied Mechanics and Engineering*, *190*(24), 3247–3270.

Gear, C.W., & Wells, D. (1984). Multirate linear multistep methods. *BIT Numerical Mathematics*, *24*(4), 484–502.

Hairer, E., & Wanner, G. (1996). Stiff and Differential-Algebraic Problems. *Solving Ordinary Differential Equations II*. Berlin: Springer.

Hairer, E., Nørsett, S.P., & Wanner, G. (1993). *Solving ordinary differential equations i (2Nd revised. ed.): Nonstiff problems*. New York: Springer.

Hairer, E., Nørsett, S.P., & Wanner, G. (2010). *Solving ordinary differential equations: v. 1: nonstiff problems. v. 2: stiff and differential-algebraic problems*. Springer.

Hanke, M. (2017). On a variable step size modification of Hines' method in computational neuroscience. arXiv:1702.05917.

Hirakis, S., Boras, B., Votapka, L., Malmstrom, R., McCulloch, A., & Amaro, R. (2015). Bridging scales through multiscale modeling: a case study on protein kinase A. *Frontiers in Physiology*, *6*, 250. doi:10.3389/fphys.2015.00250. http://journal.frontiersin.org/article/10.3389/fphys.2015.00250.

Park, K., & Felippa, C.A. (1983). Partitioned analysis of coupled systems. *Computational Methods for Transient Analysis*, *1*, 157–219.

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., & Destexhe, A. (2008). Minimal Hodgkin–Huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, *99*(4-5), 427–441.

Rangan, A.V., & Cai, D. (2007). Fast numerical methods for simulating large-scale integrate-and-fire neuronal networks. *Journal of Computational Neuroscience*, *22*(1), 81–100.

Sand, J., & Skelboe, S. (1992). Stability of backward Euler multirate methods and convergence of waveform relaxation. *BIT Numerical Mathematics*, *32*(2), 350–366.

Savcenco, V. (2008). *Multirate numerical integration for ordinary differential equations*. PhD thesis, Universiteit van Amsterdam.

Serban, R., & Hindmarsh, A.C. (2005). CVODES: the sensitivity-enabled ode solver in sundials. In *ASME 2005 international design engineering technical conferences and computers and information in engineering conference* (pp. 257–269): American Society of Mechanical Engineers.

Skelboe, S. (2000). Accuracy of decoupled implicit integration formulas. *SIAM Journal on Scientific Computing*, *21*(6), 2206–2224.

Söderlind, G., & Wang, L. (2006). Adaptive time-stepping and computational stability. *Journal of Computational and Applied Mathematics*, *185*(2), 225–243.