Journal of
Computer Science & Systems Biology

Sarmis et al., J Comput Sci Syst Biol 2013, 6:3
DOI: 10.4172/jcsb.1000112

Research Article Open Access

Assessing Numerical Resolution Methods Performance for Kinetic Models of Receptors and Channels

Merdan Sarmis[1,2], Jean-Marie C Bouteiller[1,3#], Nicolas Ambert[1], Arnaud Legendre[1,2], Serge Bischoff[1], Olivier Haeberlé[2] and Michel Baudry[1,4*]

[1]*Rhenovia Pharma SA, Mulhouse, France*
[2]*Laboratoire MIPS EA2332, Université de Haute Alsace, Mulhouse, France*
[3]*Department of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA*
[4]*Western University of Health Sciences, Pomona, CA, USA*

Abstract

In systems biology, systems of kinetic reactions are generally used to model and simulate various biochemical pathways. These reactions are translated into ordinary differential equations, which are computationally resolved by numerical algorithms. Computation performance, defined by how fast the algorithm converges to a numerical solution of the system of ordinary differential equations, critically depends on the choice of the appropriate algorithm. In this paper, we compared several algorithms used to solve ordinary differential equations applied to several kinetic models that describe the dynamic behavior of receptors and ion channels found in chemical synapses of the Central Nervous System; we provide a simplified method to determine the performances of these ordinary differential equation solvers, in order to provide a benchmark for algorithm selection. This method will facilitate the choice of the most efficient algorithm for a given kinetic model with a minimum number of tests. Our results provide a tool for identifying optimal solvers for any biological bilinear kinetic models under various experimental conditions. This comparison also underscored the complexity of biological kinetic models and illustrates how their input dependency could interfere with performance. Despite these challenges, our simplified method helps to select the best solvers for any synaptic receptors kinetic models described, with a bilinear system with minimal a priori information on the solver structure and the model.

Keywords: Algorithm selection; Synaptic receptor; Kinetic models; Ordinary differential equation; Bilinear systems; Java; Numerical resolution method.

Introduction

The main goal of systems biology [1], consists in providing a quantitative and integrative description of living organisms by using simulation of complex and interconnected models of metabolic networks and signaling pathways. These complex processes are described by systems of biochemical reactions and quantitatively analyzed by corresponding sets of mathematical equations that relate reactant concentrations and elementary rate constants to changes in product concentrations with time. In order to analyze the temporal evolution of the systems of reactions, series of ordinary differential equations (ODEs) need to be computed. ODEs are computationally solved with ODE solver algorithms, providing numerical solutions for the temporal changes of the various variables of the biological systems. Over the years, mathematicians have developed numerous ODE algorithms. However, each ODE system or model is specific regarding to the number of states, input, stiffness and linearity, and therefore, each algorithm differs in terms of speed and accuracy. If different ODE solvers produce relatively similar results, their performances are highly variable, in term of speed to reach a stable numerical solution and accuracy of the final solution. It is therefore difficult to choose the best algorithm to solve particular sets of ODE without expertise analysis. In fact, algorithm selection is a complex problem, which depends on many criteria, as described by Rice [2].

According to Ewald [3] (Figure 1), the selection of the most appropriate algorithm to solve an ODE system requires knowing 1) the model and its inputs (problem space), 2) the model size and characteristics (feature space), 3) the ODE solver structure (algorithm space), and 4) user's preferences (criteria space). Users usually have a specific expertise, either in biology, modeling, code development or mathematics, but rarely in all fields. There is, therefore, a real need for a tool guiding any users towards the choice of the best algorithm to solve ODE systems.

Rhenovia is repeatedly faced with this problem throughout the development of a simulation platform for hippocampal glutamatergic synapse [4], and its integration into complex neuronal network. To this purpose, kinetic models were built and implemented. Due to the nature of kinetic models, the ODE systems are bilinear. As this project was initiated several years ago using the Java programming language, the eight solvers compared are all Java-based. We decided to simplify the algorithm selection problem by comparing the eight ODE solver algorithms performances and determine the most appropriate for bilinear synaptic kinetic models, depending on the properties of the system under consideration. This study provides a priori knowledge on solvers performances to help users interested in launching a large number of simulations with a specific model, for example to run an analysis sensibility.

In the literature, bilinear systems are written in the mathematical form represented by equation (1), where $x(t) \in \Re^n$ is the state variable

***Corresponding authors:** Michel Baudry, Western University of Health Sciences, Pomona, CA, USA, Tel: (+33) 389 321 180; Fax: (+33) 389 555 145; E-mail: michel.baudry@rhenovia.com

#Jean-Marie C Bouteiller, Rhenovia Pharma SA, 20C Rue de Chemnitz, F-68200 Mulhouse, France, Tel: +33-389 321 180; Fax: +33-389 555 145; E-mail: jean-marie.bouteiller@rhenovia.com

Received June 17, 2013; **Accepted** July 19, 2013; **Published** July 22, 2013

Citation: Sarmis M, Bouteiller JMC, Ambert N, Legendre A, Bischoff S, et al. (2013) Assessing Numerical Resolution Methods Performance for Kinetic Models of Receptors and Channels. J Comput Sci Syst Biol 6: 150-164. doi:10.4172/jcsb.1000112

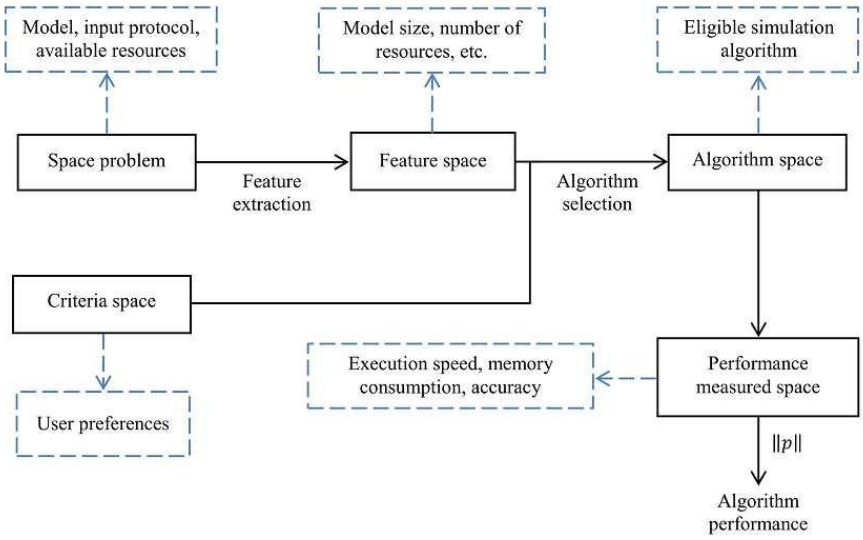

Figure 1: Algorithm selection problem. The algorithm selection problem was defined by Rice in 1976, and completed by Ewald [3]. As presented in this scheme ([3] page 22), users need to know model inputs, characteristics (size) and algorithms structure in order to select the right algorithm for a simulation.

vector and $u(t) \in \Re^m$ is the system's input vector, with $u_i(t)$ the i^{th} command of the system and B_i his associated constant matrix [5,6]. The bilinear systems are a special case of non linear affine systems.

$$\dot{x}(t) = A.x(t) + \sum_{i=1}^{m} u_i(t).B_i.x(t) \quad (1)$$

We compared eight ODE solvers (Table 1 and Table S-1), implemented in the Java language following good coding practice [7], in order to ensure adequate performance and high robustness. The ODE solvers are categorized into three groups: implicit, explicit and hybrid solvers (hybrid solver combines implicit and explicit methods at each integration step). Our kinetic models are usually bilinear systems and can be categorized as stiff or non-stiff according to their temporal evolution under a particular experimental protocol. The notion of stiffness was introduced by Curtiss and Hirschfelder [8], and has been refined several times. Since we do not intend to give a precise definition of stiffness here, we will consider a model to be non-stiff if an explicit solver is more efficient than an implicit one; otherwise we will consider it to be stiff [9]. The stiffness detection methods are commonly based on numerical resolution method stability bound [10-12]. This implies that the stiffness of a model is closely related to the selected numerical resolution method. According to Ekeland et al. [12], the precise definition of the stiffness of a system of equations is not crucial from a practical point of view. Therefore, in order to simplify the algorithm selection problem, we decide to follow Ekeland's idea, and not to determine the stiffness degree of models used in this study.

Explicit	Implicit	Hybrid
RK4 RKF JVODE ADAMS JLSODE	TR-BDF2 JVODE BDF JLSODE	Rosenbrock IMEX

Hybrid solvers combine the use of an explicit method followed by an implicit one for each step. JLSODE solver includes both explicit and implicit schemes and uses the most appropriate one at each integration step.

Table 1: Classification of the 8 ODE solvers used in our study.

All evaluated solvers have variable step-sizes, except for the selected reference solver. The reference solver is a fourth-order explicit Runge-Kutta (RK) algorithm [13] (RK4), and assumed to be the most precise at the chosen (very small) step size. The other solvers were Runge-Kutta Fehlberg, TR-BDF2, Rosenbrock, JVODE ADAMS, JVODE BDF and JLSODE. The Runge-Kutta Fehlberg [14,15] (RKF) is a fourth/fifth-order explicit RK scheme. TR-BDF2 [16,17], an implicit solver, is composed of a trapezoidal rule and a second-order Backward Differential Formula (BDF scheme), and corresponds to the ode23tb solver [18] in the Matlab® software (see details in the supplemental document). The Rosenbrock solver [19] is a hybrid RK4, and IMEX (for IMplicit-EXplicit) [20,21] is another hybrid fourth-order Runge-Kutta. JVODE is a Java version of the CVODE solver [22,23]. JVODE has not been modified, as it was implemented in Java language by the *BioUML* platform developers [24]. JVODE includes two schemes: 1) a variable order (from 1 to 12), explicit Adams-Moulton method for non-stiff systems, and 2) a variable order (from 1 to 5), implicit Backward Differential Formula (BDF scheme) for stiff systems. We will differentiate these two schemes and call them JVODE ADAMS and JVODE BDF, respectively. The last ODE solver is JLSODE (Java version of LSODE [25]), which has the same Adams-Moulton and BDF schemes as JVODE. This last solver however adds stiffness detection, as written by Uteshev and Pennefather [26]. With this stiffness detection, JLSODE starts with the Adams-Moulton scheme and reversibly switches to the BDF scheme, if the system becomes stiff. With stiffness detection and the combination of two methods, one for non-stiff and one for stiff systems, we could assume that this solver would be the best performer and the most stable between all solvers. These eight solvers were implemented in the RHENOMS™ simulation platform [4]. Further details about these 8 ODE solvers are provided in the supplemental document.

Materials and Methods

As a common basis of benchmarking, we used the Rhenovia's biosimulation platform RHENOMS and several models of synaptic receptors/channels. Figure 2 shows an example of a very simplified

biological kinetic model of a generic glutamate receptor called elementary model. The glutamate neurotransmitter is represented by *Glu,* while *R* and *RGlu* represent the receptor in the free and liganded state, respectively. In this kinetic scheme, k_1 and k_2 represent the association and dissociation rate constants for the binding of the neurotransmitter on the receptor. The ordinary differential equations corresponding to this kinetic model consist in the system of equations (2), and the bilinearity is given by the term $k_1 * R * Glu$.

$$\begin{cases} \frac{dRGlu}{dt} = k_1 * R * Glu - k_2 * RGlu \\ \frac{dR}{dt} = k_2 * RGlu - k_1 * R * Glu \end{cases} \quad (2)$$

In this study, we choose four elementary models with different kinetic structures and dynamic characteristics (fast or slow), and corresponding to ligand-gated receptor (activated by a ligand such as glutatmate or GABA) or voltage-dependent channels (activated by a change of the potential), two important elements for synaptic transmission. The first model tested in this study is represented in Figure 3, and could represent either an AMPA or NMDA synaptic receptor according to the set of parameters used [27,28]. To differentiate these two models, we will call them AMPA7 and NMDA7, respectively. Testing the algorithms with two sets of parameters applied to the same model structure allows differentiating the impact of the kinetic scheme and the impact of the parameters on the algorithms performances. Several other models developed by Rhenovia were then tested. The next tested model is a model of NMDA (NR1/NR2A) receptor [29], with 15 states variables (referred later as NMDA15), qualified as slow based on the response of the model (50-250 msec) in the referential of synaptic transmission. This NMDA receptor model gives more details and information on the receptor characteristics than the first NMDA7 model. The fourth model represents the $GABA_A$ synaptic receptor, which is considered as a fast model (20-50 msec). Lastly, a model of an N-type voltage-dependent calcium channel (VDCC) was tested. This

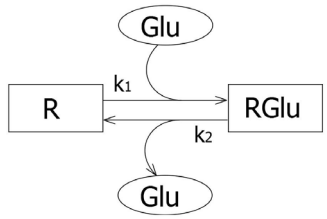

Figure 2: Example of an elementary kinetic model for ligand/receptor binding.

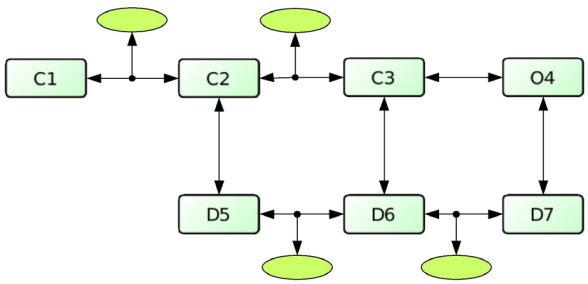

Figure 3: Kinetic scheme of the AMPA7/NMDA7 model depending to the set of parameters. The rectangles represent the different state variables. The green ovals named Glu represent the input of the model and the location with where it interacts in the model.

Model	State size	Number of equations	Number of parameters	Number of inputs	Dynamic rate (msec)
AMPA7	7	7	16	1	20-50
NMDA7	7	7	16	1	50-250
NMDA15	15	15	38	3	50-250
$GABA_A$	8	8	18	1	20-50
N type VDCC	2	6	12	1	0-5
Glutamatergic synapse	144	326	432	1	unknown

For each model, we extracted the same features: state size, number of equations, number of parameters, number of inputs and dynamic rate times.

Table 2: Features of the models used in this study.

VDCC model is relatively fast (0-5 msec for activation), and differs from the models previously described as it is based on the Hodgkin-Huxley formalism [30], instead of kinetic reactions. This VDCC model was parameterized and validated to fit Jaffe and Poirazi results [31,32]. The features of all tested models are summarized on Table 2.

We stimulated each model with two different protocols: 1) Protocol 1 (*P1)* is a single event (Figure 4A and 2) Protocol 2 (*P2)* is the same event repeated 4 times with a 10 msec interval (Figure 4B). These two protocols were selected because they correspond to the type of electrical activity that takes place in the brain under most conditions. In the resting state, neurons communicate at low frequency, with single action potential (the single event protocol). On the other hand, under a number of conditions (learning, exploration), neurons increased their firing frequency and often emit bursts of action potential at frequency between 5 and 100 Hz. As an example, we selected 10 Hz. The repeated events are protocols often used for testing synaptic receptor models [33]. In addition, this low frequency could show if input had an important impact on solver performances. For the ligand-gated receptors, the single event consists in a 1 msec pulse of 1 mM glutamate, while for the voltage-gated channel the single event is a 1 msec depolarization step from -70 mV to 0 mV. These two protocols were tested with two sets of tolerance, which are used to validate a computed step by all the numerical ODE resolution methods: 1) a common set of tolerance with the relative tolerance equal to $1e^{-3}$ (*Rtol*), and the absolute tolerance equal to $1e^{-6}$ (Atol), and 2) a set of restrictive tolerance with Rtol=$1e^{-6}$ and Atol=$1e^{-9}$. These two tolerance sets are respectively called Tol_1 and Tol_2. Although this choice may appear arbitrary, experience leads us to consider these values as relevant for numerical resolution methods [26,34].

Explicit Runge-Kutta methods are stable algorithms, as long as the integration step-size remains small enough. Therefore, we selected the fourth-order explicit Runge-Kutta (RK4) algorithm solver with a constant step-size of 0.5 µsec, as our reference to ensure that the solution of the simulation would remain stable and accurate. To assess performance, we used the execution time (speed) and the number of points (which is proportional to memory consumption) necessary to complete the simulation (i.e. to reach a stable numerical solution), which satisfy the tolerance parameter sets. As the execution time differs a little for each simulation, we repeat it ten times and make an average for measuring the execution times. For each model and protocol, the calculated output of the simulation is the current generated by receptor/channel activation as a function of time. We also evaluated the relative precision of each algorithm by determining the Mean Square Error (MSE), and the Normalized Root Mean Square Deviation (NRMSD) between the results produced by a solver and those generated by the

reference solver (RK4). More details are available in the supplemental document for the measurement of these two accuracy criteria. These 4 performance criteria (execution time, memory consumption, MSE and NRMSD) were selected as indices for rapidly obtaining the most accurate result using the lowest computer resources.

To quantify each algorithm performance, we simplified the algorithm selection problem and used the $||p||$ norm value, as illustrated in equation (3), where p_i represents our selected criteria and w_i the priority that user give for each criteria, respectively. Importantly, in this study, we decided to use the same priority level for all four criteria, and thus, the w vector is $[1, 1, 1, 1]^T$. The norm value provides a rapid comparison of the overall performance of the various algorithms, as it generates a single value for the performance of each algorithm. Algorithms yielding a small norm value will thus be considered as more efficient than others yielding a larger norm value.

$$\|p\| = \sqrt{\sum_{i=1}^{n=4} (p_i . w_i)^2} \quad (3)$$

Furthermore, to verify the performance variation of algorithms, we computed the differences in norm values between the different sets of tolerance and protocols. The performance variations enable us to determine whether an algorithm is sensitive to input protocol, or to tolerance parameter sets. Simulations were performed on a workstation with a LINUX (Ubuntu 10.04) operating system and an Intel® Xeon® CPU at 2.67 GHz frequency equipped with 12 Gbytes of RAM and the version 1.6 of Java installed.

Results and Discussion

In computational neuroscience, most models are stimulus-dependent and bilinear. The major goal of our study was to simplify the algorithm selection problem by benchmarking the solvers performances implemented in RHENOMS. This simplification could make a complex problem accessible to all users, and provide a recommendation for the possible use of biological bilinear kinetic models. Therefore, we will not discuss here the details of the solvers or of the models.

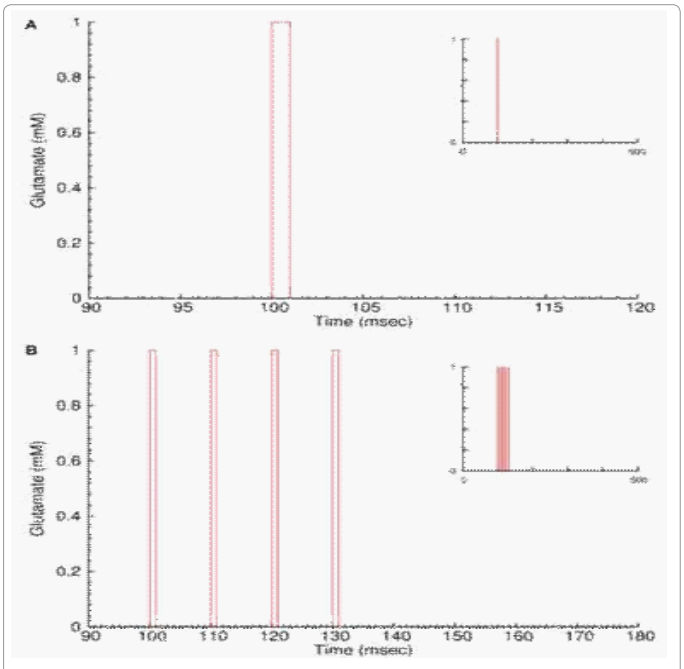

Figure 4: Stimulation protocols. A: Single event protocol P1, B: Repeated event protocol P2. Inserts represent the stimulus embedded in the entire simulation run.

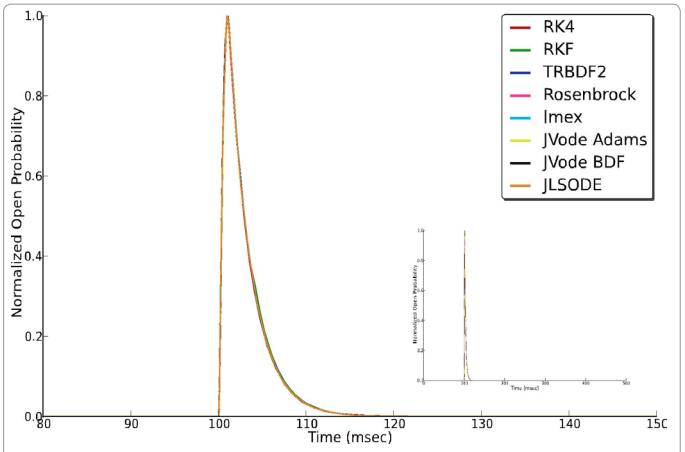

Figure 5: Normalized open state probability for AMPA7 model. Insert represents the results embedded in the entire simulation run.

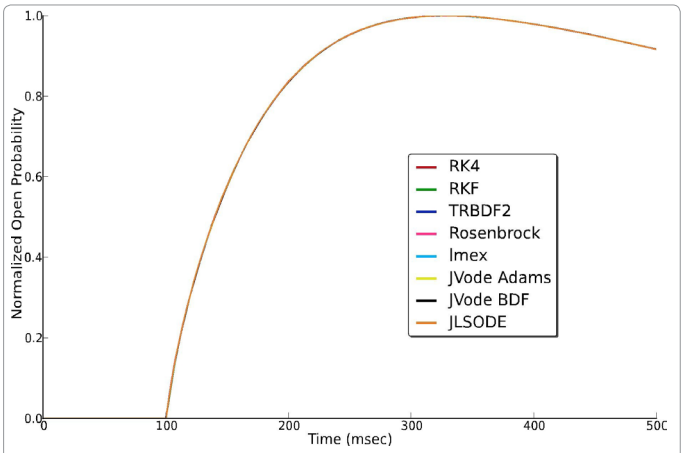

Figure 6: Normalized open state probability for NMDA7 model.

AMPA7/NMDA7 receptor model

This kinetic scheme, presented in Figure 3, comprises 7 states, 7 equations, 16 parameters and 1 input variable. It could model a fast receptor like AMPA or a slower one like NMDA, according to the set of parameters used. For this model, we choose the open state probability represented by the O4 rectangle in Figure 3 as readout. The open probability of the AMPA7 model is depicted on Figure 5, and the same state for the NMDA7 model is shown in Figure 6. As we can see, the two models (AMPA7 and NMDA7) do not give the same dynamics on the open probability state. The raw data of the AMPA7 model are depicted on Table S-2 and S-3 on the additional document for the P1 and P2 protocols. Similar data for the NMDA7 model are given in Table S-4 and S-5. The performance values are presented on Table 3 for the AMPA7 and on Table 4 for the NMDA7 model.

In order to find the best algorithm for this model, we identified the solver(s) with the smallest norm values for the two stimulation protocols and the two sets of tolerance (Figure 7). Based on the first protocol (Figure 7B), the hybrid IMEX solver provides the best performances for the AMPA7 model whereas solvers with a BDF scheme are the

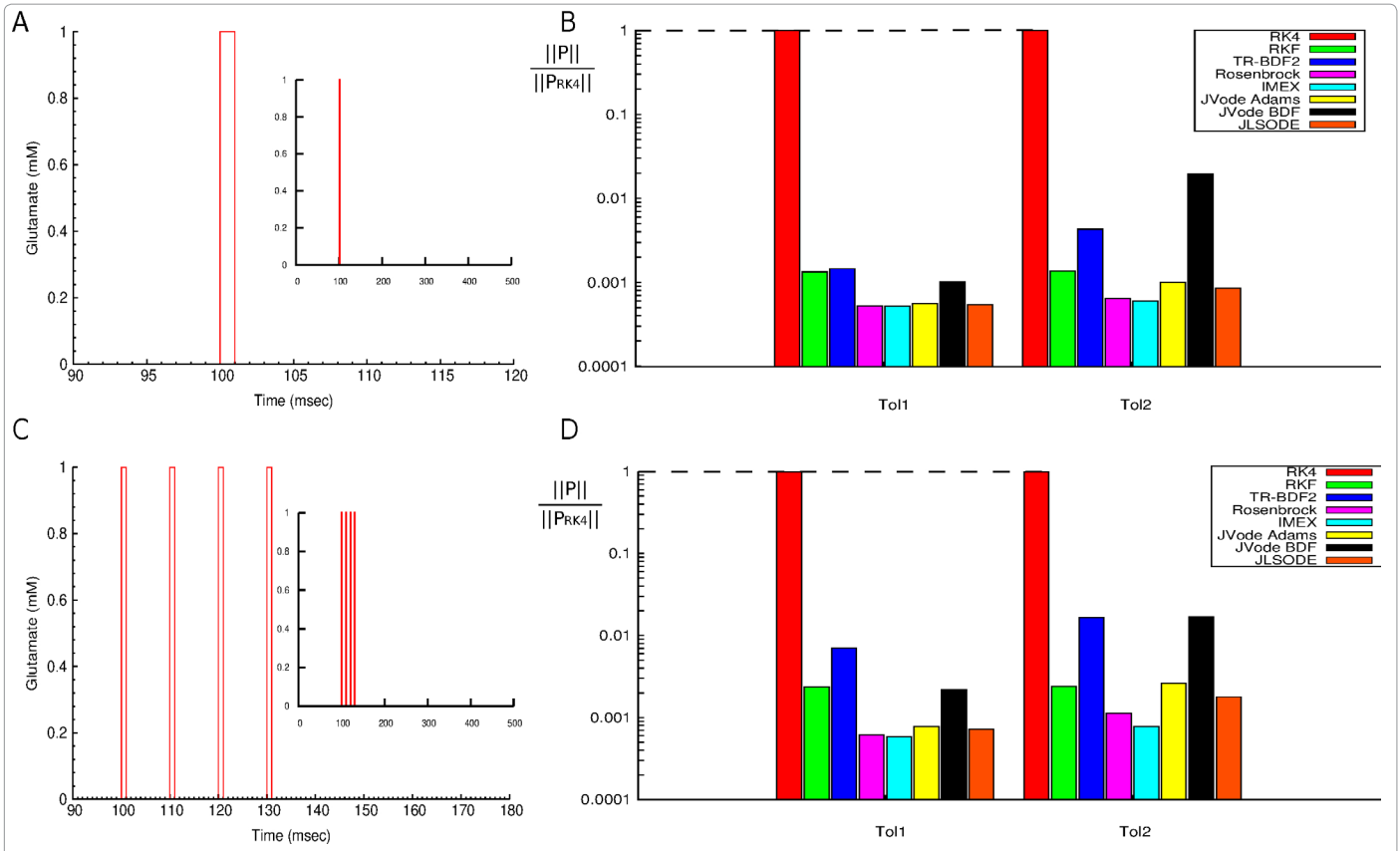

Figure 7: Normalized norm values for the AMPA7 receptor model. A: Review of the protocol P1. B: The normalized norm values for the 8 solver algorithms with P1. C: Review of the protocol P2. D: The normalized norm values for the 8 solver algorithms with P2. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

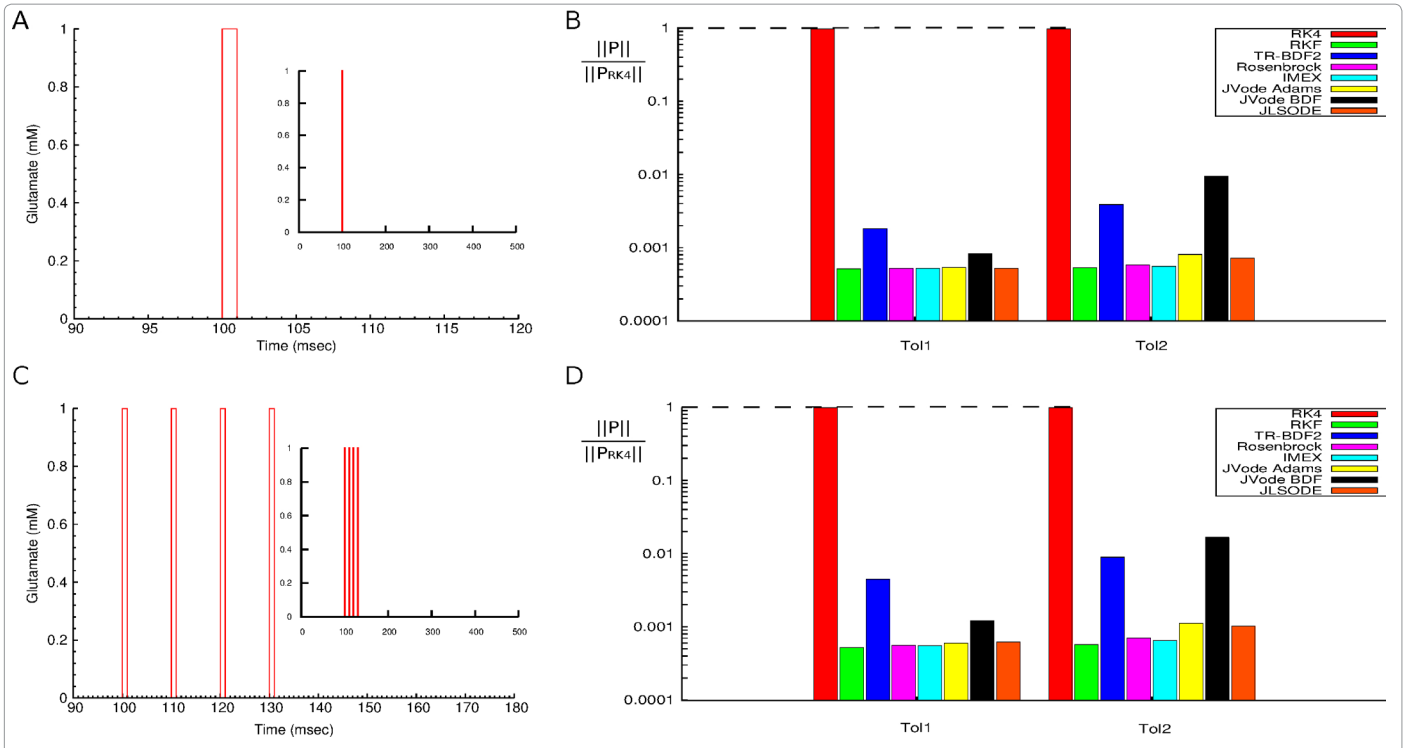

Figure 8: Normalized norm values for the NMDA7 receptor model. A: Review of the protocol P1. B: The normalized norm values for the 8 solver algorithms with P1. C: Review of the protocol P2. D: The normalized norm values for the 8 solver algorithms with P2. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
Protocol 1 (single pulse)								
Tol_1	1.0	$1.34e^{-3}$	$1.46e^{-3}$	$5.27e^{-4}$	$5.23e^{-4}$	$5.59e^{-4}$	$1.02e^{-3}$	$5.49e^{-4}$
Tol_2	1.0	$1.35e^{-3}$	$4.31e^{-2}$	$6.44e^{-4}$	$5.98e^{-4}$	$1.01e^{-3}$	$1.97e^{-2}$	$8.64e^{-4}$
ΔP1 Tol_1/Tol_2	-	$1e^{-5}$	$4.16e^{-2}$	$1.17e^{-4}$	$7.5e^{-5}$	$4.5e^{-4}$	$1.87e^{-2}$	$3.15e^{-4}$
Protocol 2 (repeated pulses)								
Tol_1	1.0	$2.39e^{-3}$	$7.09e^{-3}$	$6.14e^{-4}$	$5.87e^{-4}$	$7.85e^{-4}$	$2.2e^{-3}$	$7.19e^{-4}$
Tol_2	1.0	$2.41e^{-3}$	$1.65e^{-2}$	$1.12e^{-3}$	$7.86e^{-4}$	$2.61e^{-3}$	$1.69e^{-2}$	$1.79e^{-3}$
Δ P2 Tol_1/Tol_2	-	$2.1e^{-5}$	$9.41e^{-3}$	$5.06e^{-4}$	$1.99e^{-4}$	$1.8e^{-3}$	$1.47e^{-2}$	$1.1e^{-3}$
Δ Tol_1 P1/P2	-	$1.1e^{-3}$	$5.6e^{-3}$	$8.7e^{-5}$	$7.5e^{-5}$	$2.26e^{-4}$	$1.2e^{-3}$	$1.7e^{-4}$
Δ Tol_2 P1/P2	-	$1.1e^{-3}$	$2.66e^{-2}$	$4.76e^{-4}$	$7.11e^{-4}$	$1.6e^{-3}$	$2.8e^{-3}$	$9.26e^{-4}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1 Tol_1/ Tol_2 represents the absolute difference between Tol_1 and Tol_2 for P1, whereas Δ P2 Tol_1/ Tol_2 represents the same absolute difference for P2. Δ Tol_1 P1/P2 represents the absolute difference between P1 and P2 for Tol_1 parameters, whereas Δ Tol_2 P1/P2 represents the same absolute difference for Tol_2.

Table 3: Quantification and comparison of normalized solver performances for the AMPA7 model.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
Protocol 1 (single pulse)								
Tol_1	1.0	$5.14e^{-4}$	$1.83e^{-3}$	$5.24e^{-4}$	$5.21e^{-4}$	$5.41e^{-4}$	$8.36e^{-4}$	$5.25e^{-4}$
Tol_2	1.0	$5.36e^{-4}$	$3.93e^{-3}$	$5.84e^{-4}$	$5.63e^{-4}$	$8.14e^{-4}$	$9.54e^{-3}$	$7.2e^{-4}$
ΔP1 Tol_1/Tol_2	-	$2.2e^{-5}$	$2.1e^{-3}$	$6e^{-5}$	$4.2e^{-5}$	$2.73e^{-4}$	$8.7e^{-3}$	$1.95e^{-4}$
Protocol 2 (repeated pulses)								
Tol_1	1.0	$5.29e^{-4}$	$4.55e^{-3}$	$5.64e^{-4}$	$5.57e^{-4}$	$6.08e^{-4}$	$1.23e^{-3}$	$6.28e^{-4}$
Tol_2	1.0	$5.79e^{-4}$	$9.06e^{-3}$	$7.1e^{-4}$	$6.58e^{-4}$	$1.13e^{-3}$	$1.63e^{-2}$	$1.03e^{-3}$
Δ P2 Tol_1/Tol_2	-	$2.1e^{-5}$	$9.41e^{-3}$	$5.06e^{-4}$	$1.99e^{-4}$	$1.8e^{-3}$	$1.47e^{-2}$	$1.1e^{-3}$
Δ Tol_1 P1/P2	-	$1.5e^{-5}$	$2.72e^{-3}$	$4e^{-5}$	$3.6e^{-5}$	$6.7e^{-5}$	$3.9e^{-4}$	$1.03e^{-4}$
Δ Tol_2 P1/P2	-	$4.3e^{-5}$	$5.13e^{-3}$	$1.26e^{-4}$	$9.5e^{-5}$	$8.14e^{-4}$	$6.76e^{-3}$	$3.8e^{-4}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1 Tol_1 / Tol_2 represents the absolute difference between Tol_1 and Tol_2 for P1, whereas Δ P2 Tol_1/ Tol_2 represents the same absolute difference for P2. Δ Tol_1 P1/P2 represents the absolute difference between P1 and P2 for Tol_1 parameters, whereas Δ Tol_2 P1/P2 represents the same absolute difference for Tol_2.

Table 4: Quantification and comparison of normalized solver performances for the NMDA7 model.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
Protocol 1 (single pulse)								
Tol_1	1.0	$8.025e^{-3}$	$3.587e^{-3}$	$5.25e^{-4}$	$5.22e^{-4}$	$5.45e^{-4}$	$8.83e^{-4}$	$5.59e^{-4}$
Tol_2	1.0	$8.051e^{-3}$	1.8478	$6.04e^{-4}$	$5.76e^{-4}$	$8.96e^{-4}$	$1.163e^{-2}$	$8.79e^{-4}$
ΔP1 Tol_1/Tol_2	-	$2.6e^{-5}$	1.8444	$7.9e^{-5}$	$5.4e^{-5}$	$3.51e^{-4}$	$1.075e^{-2}$	$3.29e^{-4}$
Protocol 2 (repeated pulses)								
Tol_1	1.0	$8.034e^{-3}$	$8.004e^{-3}$	$5.74e^{-4}$	$5.63e^{-4}$	$6.76e^{-4}$	$1.51e^{-3}$	$6.6e^{-4}$
Tol_2	1.0	$8.067e^{-3}$	2.7826	$8.04e^{-4}$	$7.18e^{-4}$	$1.554e^{-3}$	$2.18e^{-2}$	$1.35e^{-3}$
Δ P2 Tol_1/Tol_2	-	$3.3e^{-5}$	2.7746	$2.3e^{-4}$	$1.55e^{-4}$	$8.68e^{-4}$	$2.029e^{-2}$	$6.92e^{-4}$
Δ Tol_1 P1/P2	-	$9.1e^{-6}$	$4.417e^{-3}$	$4.9e^{-5}$	$4.1e^{-5}$	$1.31e^{-4}$	$6.27e^{-4}$	$1.01e^{-4}$
Δ Tol_2 P1/P2	-	$1.6e^{-6}$	0.9347	$2.4e^{-4}$	$1.42e^{-4}$	$6.48e^{-4}$	$1.016e^{-2}$	$4.73e^{-4}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1Tol1/Tol2 represents the absolute difference between Tol_1 and Tol_2 for P1, whereas Δ P2 Tol1/Tol2 represents the same absolute difference for P2. Δ Tol_1 P1/P 2 represents the absolute difference between P1 and P2 for Tol_1 parameters, whereas Δ Tol_2 P1/P2 represents the same absolute difference for Tol_2.

Table 5: Quantification and comparison of normalized solver performances for the NMDA15 model.

worst performers. With the second protocol (Figure 7D), we observe the same tendancy. If we analyze the raw data (in the supplemental document), TR-BDF2 and JVODE BDF needed an important number of points to compute the results. Contrary, the IMEX algorithm is the one that had the lowest memory footprint and had the most accurate results. In addition, its execution time is very short.

According to performances obtained on this AMPA7 model, TR-BDF2 and JVODE BDF appear to be the most sensitive to protocol and tolerance sets variations with this model. The RKF solver is the least sensitive to tolerance variations, whereas hybrid solvers were the least sensitive to protocol variations.

In Figure 8, the same graphics are plotted for the parameters, which model the NMDA7 synaptic receptor to visualize the ||*p*|| performances values. With a slow dynamics on this kinetics scheme, the RKF solver became the best choice for all our tested protocols and tolerance sets. As for the AMPA7 model, solvers using BDF scheme are the worst performers. In fact, the raw data (in the supplemental data) show that these solvers needed a large number of points, and a long execution time. Regarding the performance variations on these models, RKF appears to be the least sensitive to protocols and tolerance variations, whereas JVODE BDF and TR-BDF2 are the most sensitive.

This kinetic scheme generates different dynamics with different parameters. With the parameters modeling the fast AMPA7 model, the IMEX solver appears to be the most suitable; and with the parameters modeling the slow NMDA7 model, the RKF solver appears to be the most suitable. With this observation, it appears that the dynamics of the system is more important than the structure of the kinetic scheme.

NMDA15 receptor model

The NMDA15 (NR1/NR2A) receptor is a relatively slow (50-250 msec) glutamate receptor. This model was previously calibrated and validated to fit a variety of experimental data [26]. The tested NMDA15 model had 3 inputs: glutamate concentration, glycine concentration and depolarization. For this study, we choose the glutamate as the protocol input. Glycine and depolarization are parameterized to be constant. The current generated by receptor activation using the single pulse protocol *P1,* and calculated with all the solvers is depicted on Figure 9. All algorithms provided similar results, and Figure 10 summarizes these performance values. The criteria of the NMDA15 model are quantified on Table S-6 and S-7 of the supplemental document. Algorithm performance values for the NMDA15 receptor model are quantified in Table 5.

To find the best algorithm for this model, we identified the solver(s) with the smallest norm values for the two stimulation protocols and the two sets of tolerance (Figure 10B and 10D). Based on Figure 10B, for *P1* protocol, the algorithm performances did not differ much between Tol_1 and Tol_2 tolerances except for the BDF scheme (TR-BDF2 and JVODE BDF solvers), for which a more than 10 times difference on ||*p*|| performance value is observed. For *P2* protocol, the performance differences between Tol_1 and Tol_2 parameters were more pronounced with the last three algorithms. As for *P1* protocol, BDF scheme performances deteriorated significantly with Tol_2. For both stimulation protocols, the best solver was IMEX for both tolerance parameters, while the least performing algorithm was RKF for Tol_1 and TR-BDF2 for Tol_2 for both protocols. Considering the algorithm performance variations presented in Table 5, we concluded that TR-BDF2 solver was the most sensitive to protocol and tolerance parameters for the NMDA15 receptor model, whereas RKF solver has the overall most stable performance.

$GABA_A$ receptor model

The $GABA_A$ synaptic receptor is a fast model with approximately the same dynamics as the AMPA7 model. It is sensitive to the GABA neurotransmitter. Our $GABA_A$ model was developed using parameters is presented [35]. The tested $GABA_A$ model had 8 states, 8 equations, 18 parameters and 1 input. Figure 11 shows the $GABA_A$ model current generated with protocol P1. This current was our readout to determine performance values. Our results indicate that all solvers provide similar results. This model's quantification criteria are depicted in details in the supplemental document on Table S-8 and S-9 (respectively for *P1* and *P2* protocols). The algorithm performance values for the tested $GABA_A$ receptor model are quantified in Table 6.

The graphs presenting the ||*p*|| performance value in Figure 12 quantitatively helps to determine the most appropriate algorithm for the $GABA_A$ receptor model. A brief examination of Figure 12B and 12D suggests that TR-BDF2 and JVODE BDF are the worst solvers for this model, as they yield large ||*p*|| values. JLSODE appears as the best solver for this model with the first protocol (*P1*), and the first tolerance parameters (*Tol1*). For all others simulations, IMEX yields the best performances. Observing the performance variations on Table 6, the worst solvers (JVODE BDF and TR-BDF2) for this model are the most sensitive to input and tolerance variations. With the GABAA model and the tested situations, IMEX and RKF yield the most consistent performances.

In fact, from Table S-8 and S-9 in the supplemental file, we could observe that RKF is the faster solver. However, the IMEX was most accurate with the $GABA_A$ model with all tested simulation. The IMEX was clearly not competing with other solvers for the execution time on the second protocol, as it needs longer time to compute the results. But, its low memory consumption combined with its final accuracy background made it the best solver for this particular model.

N-Type VDCC model

In general, it has proven difficult to model voltage-dependent calcium channels (VDCC), due to the existence of a tail current at the end of the plateau current. For our comparative studies, we decided to simulate the N-Type VDCC, a high voltage-activated channel. This channel is very fast (0-5 msec activation), as shown in Figure 13. For this channel, we increased the depolarization duration to 10 msec. The interval between pulses was adjusted in protocol *P2* to maintain the 10 Hz stimulation condition. All the solvers were able to accurately fit the references results. The various performance values parameters are summarized in Figure 14.

No significant differences were observed between the eight considered algorithms, except for TR-BDF2, which generated the highest ||*p*|| norm performance values for both protocols and tolerance sets. Comparing the overall performance values of the solvers (Table 7), JLSODE was the algorithm that provided the best performance for both stimulation protocols and tolerance sets. However, RKF performance values were very close to that of JLSODE ($1e^{-6}$ difference).

When we analyzed the criteria quantification table (Table S-10 and S-11 in supplemental file), all algorithms required approximately the same number of points, except for TR-BDF2, which needed more points with Tol_2 parameters (10 times more for *P1* and 64 times more for *P2*). In addition, all solvers generated almost the same NRMSD,

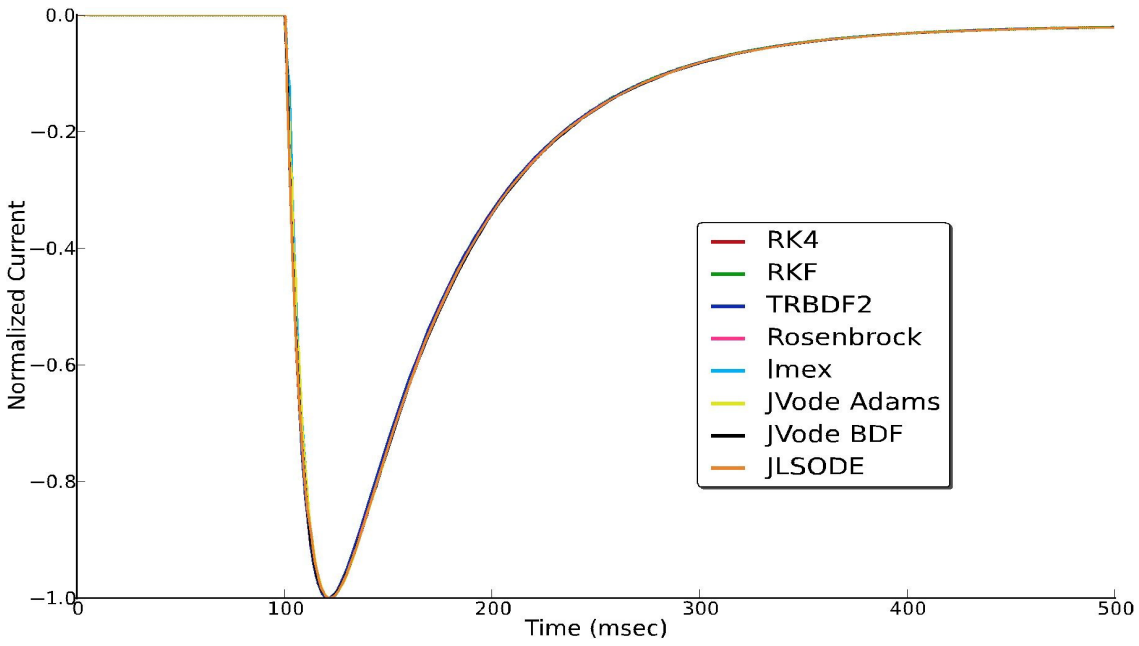

Figure 9: Normalized NMDA15 receptor current. NMDA15 receptor's current resulting from a single pulse protocol generated by all solver algorithms.

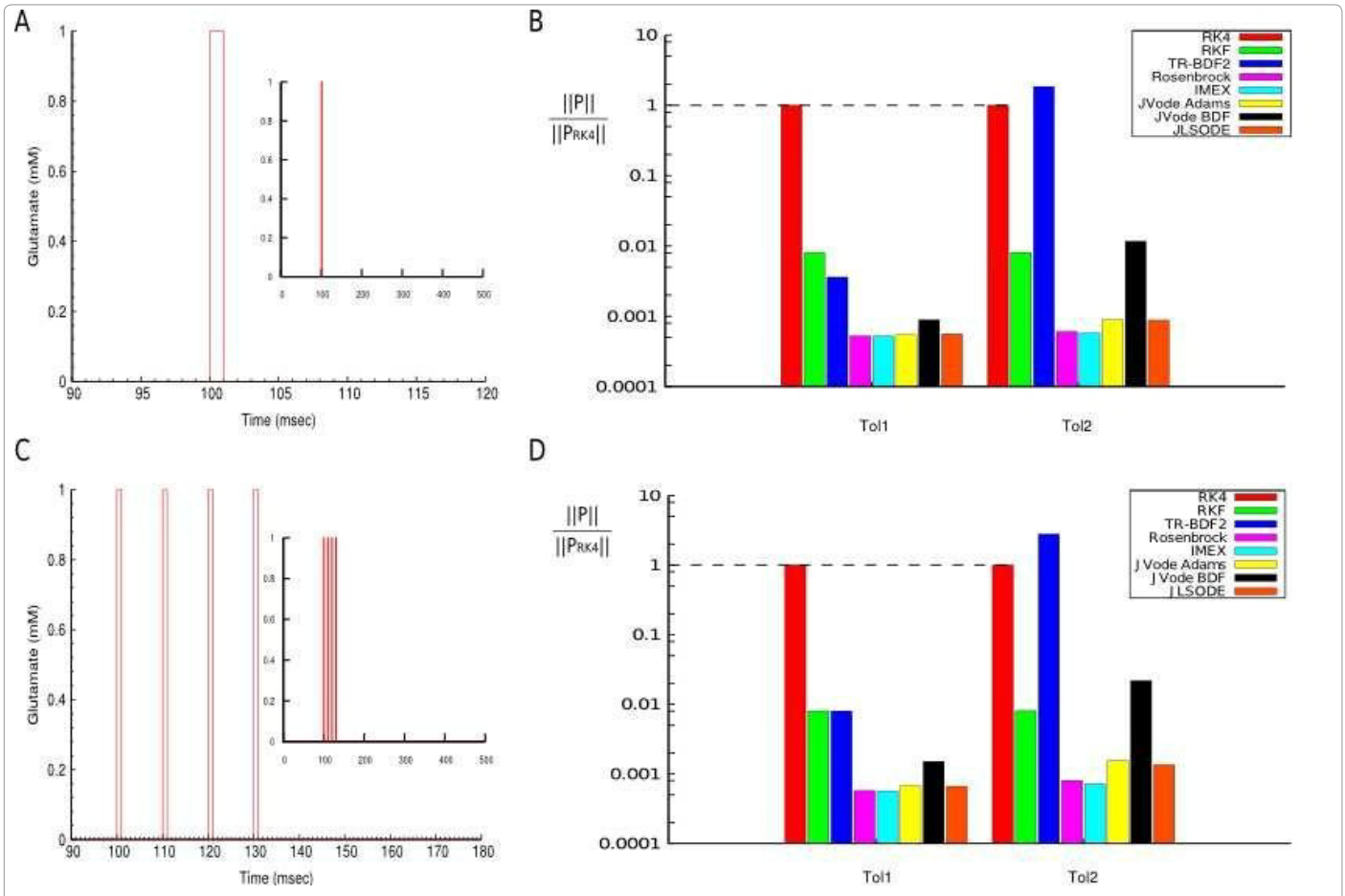

Figure 10: Normalized norm values for the NMDA15 receptor model. A: Review of the protocol P1. B: The normalized norm values for the 8 solver algorithms with P1. C: Review of the protocol P2. D: The normalized norm values for the 8 solver algorithms with P2. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

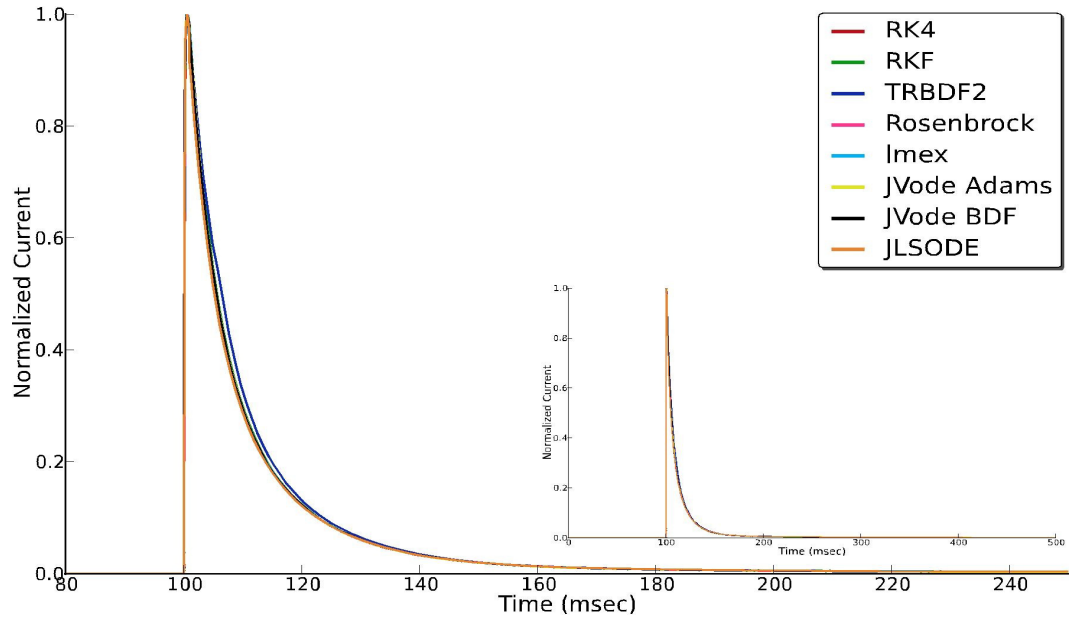

Figure 11: Normalized $GABA_A$ receptor model current. $GABA_A$ receptor's current resulting from the single pulse protocol generated by all solvers. Insert represents the results embedded in the entire simulation run.

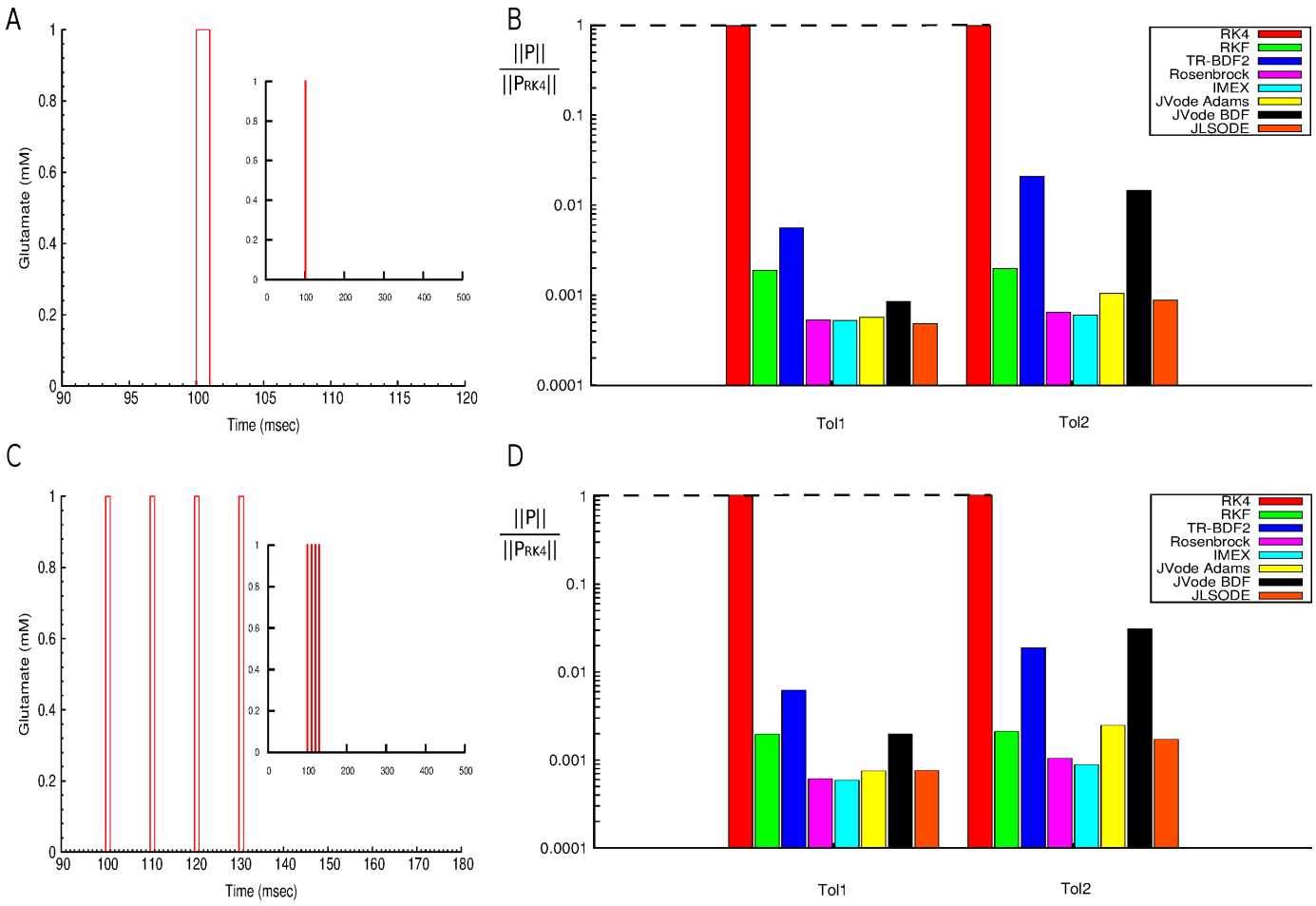

Figure 12: Normalized norm values for the $GABA_A$ receptor model. A: review of the protocol P1. B: The normalized norm values for the 8 solver algorithms with P1. C: review of the protocol P2. D: The normalized norm values for the 8 solver algorithms with P2. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
Protocol 1 (single pulse)								
Tol_1	1.0	$1.19e^{-3}$	$5.66e^{-3}$	$5.32e^{-4}$	$5.27e^{-4}$	$5.74e^{-4}$	$8.62e^{-4}$	$4.88e^{-4}$
Tol_2	1.0	$1.98e^{-3}$	$2.11e^{-2}$	$6.49e^{-4}$	$6.06e^{-4}$	$1.05e^{-3}$	$1.47e^{-2}$	$8.88e^{-4}$
$\Delta P1\ Tol_1/Tol_2$	-	$7.9e^{-4}$	$1.54e^{-2}$	$1.17e^{-4}$	$7.9e^{-5}$	$4.76e^{-4}$	$1.38e^{-2}$	$4e^{-4}$
Protocol 2 (repeated pulses)								
Tol_1	1.0	$1.95e^{-3}$	$6.12e^{-3}$	$6.07e^{-4}$	$5.87e^{-4}$	$7.48e^{-4}$	$1.95e^{-3}$	$7.53e^{-4}$
Tol_2	1.0	$2.09e^{-3}$	$1.86e^{-2}$	$1.03e^{-3}$	$8.8e^{-4}$	$2.46e^{-3}$	$3.07e^{-2}$	$1.7e^{-3}$
Δ P2 Tol_1/Tol_2	-	$1.4e^{-4}$	$1.25e^{-2}$	$4.23e^{-4}$	$2.93e^{-4}$	$1.71e^{-3}$	$1.12e^{-2}$	$9.47e^{-3}$
$\Delta\ Tol_1$ P1/P2	-	$7.6e^{-4}$	$4.6e^{-4}$	$7.5e^{-5}$	$6e^{-5}$	$1.74e^{-4}$	$1.08e^{-3}$	$2.65e^{-4}$
$\Delta\ Tol_2$ P1/P2	-	$11e^{-4}$	$2.5e^{-3}$	$3.81e^{-4}$	$2.74e^{-4}$	$6.6e^{-4}$	$1.6e^{-2}$	$8.12e^{-4}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1 Tol_1/Tol_2 represents the absolute difference between Tol_1 and Tol_2 for P1, whereas Δ P2 Tol_1 / Tol_2 represents the same absolute difference for P2. $\Delta\ Tol_1$ P1/P 2 represents the absolute difference between P1 and P2 for Tol_1 parameters, whereas $\Delta\ Tol_2$ P1/P2 represents the same absolute difference for Tol_2.

Table 6: Quantification and comparison of normalized solver performances for the $GABA_A$ model.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
Protocol 1 (single pulse)								
Tol_1	1.0	$5.004e^{-2}$	$5.154e^{-2}$	$5.006e^{-2}$	$5.007e^{-2}$	$5.008e^{-2}$	$5.008e^{-2}$	$5.003e^{-2}$
Tol_2	1.0	$5.004e^{-2}$	0.683	$5.006e^{-2}$	$5.007e^{-2}$	$5.008e^{-2}$	$5.008e^{-2}$	$5.003e^{-2}$
Δ P1 Tol_1/Tol_2	-	$2.5e^{-8}$	0.6314	$9.84e^{-7}$	$9.4e^{-8}$	$1.056e^{-6}$	$1e^{-6}$	$1.2e^{-8}$
Protocol 2 (repeated pulses)								
Tol_1	1.0	$5.001e^{-2}$	$5.647e^{-2}$	$5.018e^{-2}$	$5.019e^{-2}$	$5.037e^{-2}$	$5.037e^{-2}$	$5e^{-2}$
Tol_2	1.0	$5.001e^{-2}$	3.215	$5.018e^{-2}$	$5.019e^{-2}$	$5.037e^{-2}$	$5.037e^{-2}$	$5e^{-2}$
Δ P2 Tol_1/Tol_2	-	$2.7e^{-10}$	3.1587	$2.8e^{-8}$	$1.08e^{-7}$	$6.4e^{-8}$	$2e^{-9}$	$1e^{-9}$
$\Delta\ Tol_1$ P1/P2	-	$5.976e^{-6}$	$4.933e^{-3}$	$1.198e^{-5}$	$1.192e^{-5}$	$2.988e^{-5}$	$2.899e^{-5}$	$5.99e^{-6}$
$\Delta\ Tol_2$ P1/P2	-	$6e^{-6}$	2.5322	$1.102e^{-5}$	$1.213e^{-5}$	$3.006e^{-5}$	$3e^{-5}$	$6.01e^{-6}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1 Tol_1/Tol_2 represents the absolute difference between Tol_1 and Tol_2 for P1, whereas Δ P2 Tol_1/Tol_2 represents the same absolute difference for P2. $\Delta\ Tol_1$ P1/P2 represents the absolute difference between P1 and P2 for Tol_1 parameters, whereas $\Delta\ Tol_2$ P1/P2 represents the same absolute difference for Tol_2.

Table 7: Quantification and comparison of normalized solver performances for the N-type VDCC model.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BDF	JLSODE
P_1	1.0	0.3156	0.4989	0.1334	0.4327	0.1425	0.1456	0.1537
P_2	1.0	0.3273	0.261	0.1663	0.4705	0.1884	0.1831	0.1926
$\Delta\ Tol_{1\ P1/P2}$	-	$1.16e^{-2}$	0.2379	$3.291e^{-2}$	$3.773e^{-2}$	$4.585e^{-2}$	$3.753e^{-2}$	$3.886e^{-2}$

Tol_1 and Tol_2 represent the ||p|| norm value for each algorithm with P1 and P2 protocols. $\Delta Tol_{1\ P1/P2}$ represents the absolute difference between P1 and P2 for Tol_1 parameters.

Table 8: Quantification and comparison of normalized solver performances for the glutamate synapse model.

except for RKF (for both stimulation protocols) and TR-BDF2 (for Tol_2 tolerance set).

As illustrated in the supplemental file (Figure S-1), when switching from Tol_1 to Tol_2, the TR-BDF2 solver yielded smaller error values (both MSE and NRMSD), although execution time and memory consumption (i.e. number of points) increased and exceeded RK4 values. In addition, a small relative tolerance value ($Rtol<1e^{-6}$) was not required to obtain accurate results with TR-BDF2 solver. These results indicate that the JLSODE solver is the best choice for this VDCC model, as it provides the best performance. However, RKF results were very close. In addition, when observing performance variations, RKF performances were less sensitive to input protocols or tolerance sets.

Application to RHENOMS model

In order to complete our study, we then analyzed the performance of the different ODE solvers with a model of glutamatergic synapse that integrates a large number of synaptic elementary models, such as receptors, channels, transporters, enzymes and signaling pathway

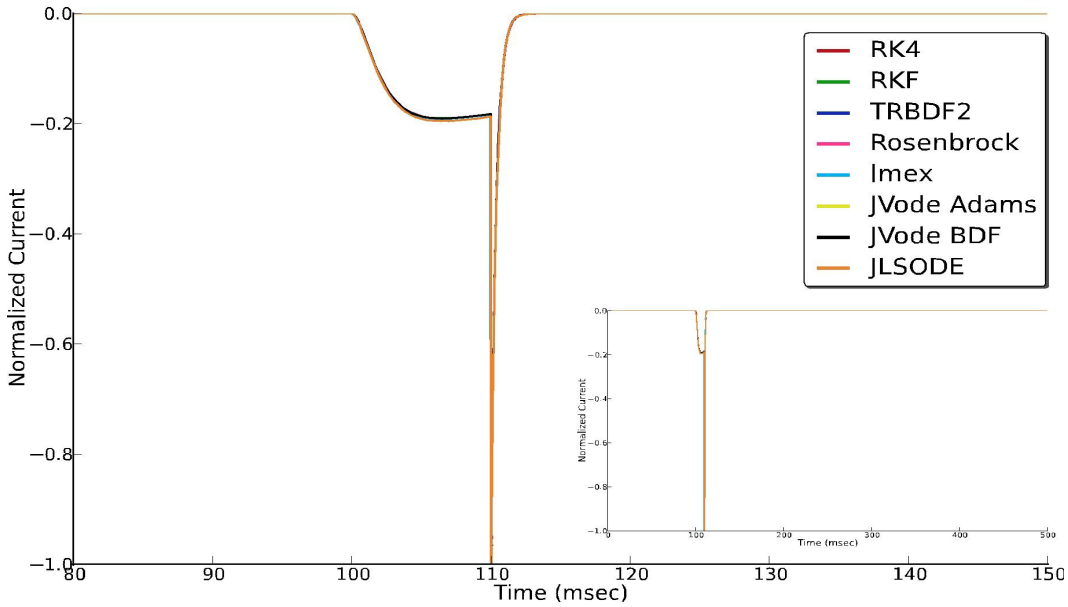

Figure 13: N-type VDCC current. N-type VDCC's current resulting from a single event protocol generated by all ODE solvers Insert represents the results embedded in the entire simulation run.

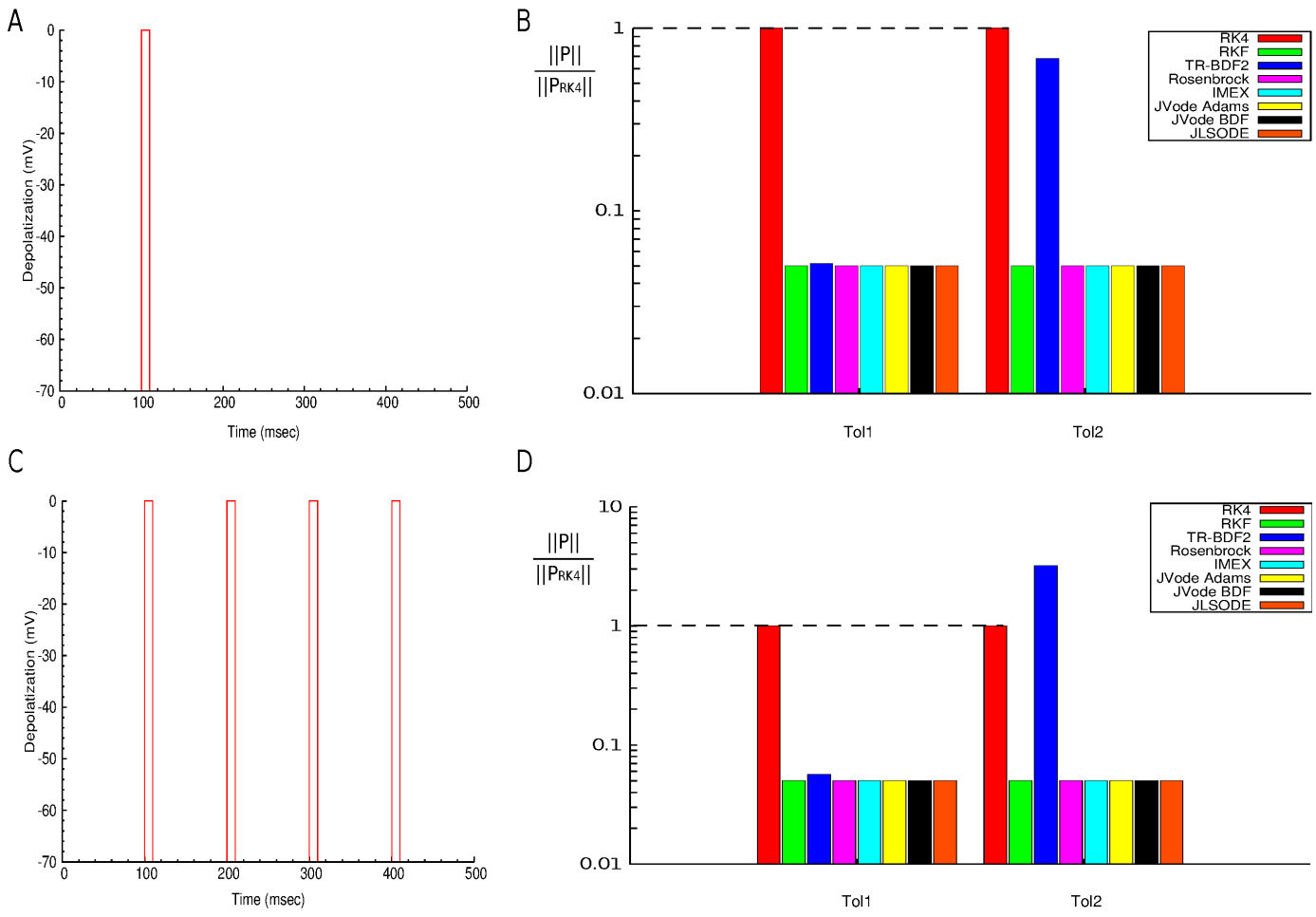

Figure 14: Normalized norm values for the N type VDCC model. A: review of the protocol P1. B: The normalized norm values for the 8 solver algorithms with P1. C: review of the protocol P2. D: The normalized norm values for the 8 solver algorithms with P2. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

models. Overall, the glutamatergic synapse platform integrates more than 300 equations to be solved at each integration step, 144 variable states for 21 bilinear synaptic elementary kinetic models. A simplified representation of the glutamatergic model is illustrated in Figure 15, which specifically highlights the previously tested elementary models. In addition to the large number of equations, the neurotransmitter release model uses the Systems Biology Markup Language (SBML) events [36], which results in more non-linearity, stiffness and difficulties to solve equations during simulations.

Despite the power and available memory in the computer we used, computing the glutamatergic synapse model with TR-BDF2 algorithm with the second tolerance set (Tol_2) was beyond the available capacity, and we, therefore, cannot present these results. Indeed such calculation would require a computer cluster with much more memory as the number of points computed for all elementary models becomes prohibitive. The second set of tolerance placed TR-BDF2 in its worst configuration, and this algorithm did not need a restrictive tolerance to provide accurate enough results. In addition, users do not often change the default tolerance parameters, which are commonly set close to our first set (Tol_1). For these reasons, we present algorithm performances with the first tolerance set only (Tol_1) with the two stimulation protocols (*P1* and *P2*). Quantified criteria are presented in the supplemental file by Table S-12.

Considering the number of synaptic receptors integrated in this model, we selected the postsynaptic current as the main readout, as it corresponds to the sum of currents generated by activation of all receptors, and is a common readout in biological experiments. In addition, we replaced the number of points (used to evaluate the performance of previous tested models) by the memory consumption for the simulation, which seems more appropriate for this complex model. A two-second simulation is performed, with a 100 msec delay before the start of stimulation was applied, and the quantified performances values and variations for the glutamatergic synapse model are presented in Table 8.

As shown in Figure 16, the Rosenbrock solver provided the smallest norm values for both protocols. In contrast, IMEX, which is a hybrid solver too, with a fourth-order RK-scheme, generated a large norm value. Surprisingly, TR-BDF2 algorithm improved its performance between *P1* and *P2* protocol. This could be due to its step-size adaptation t_{stop} (see supplemental file), with which TR-BDF2 readapts the integration step-size around an abrupt variation (i.e. a stimulation input). In theory, the more stimulation the model receives, the more efficient the algorithm becomes [16]. It was the sole algorithm with a better performance with *P2* as compared to *P1*. TR-BDF2 was also the most sensitive to input protocol with the glutamatergic synapse model, whereas RKF was the least sensitive.

Conclusion

Computational neuroscience encompasses a wide range of kinetic models with very different characteristics. It is very difficult to select the appropriate algorithm to solve the ODE systems representing biological bilinear kinetic models. Indeed, in order to select the right algorithm, as suggested by Rice, users need to know the model input, model characteristics (size), and ODE solver structure. A benchmark comparing various ODE solver algorithms or a recommendation could help users to select the most appropriate algorithm for a given simulation. We simplified the algorithm selection problem and benchmarked eight ODE solvers performances using several types of kinetic models with two different stimulation protocols. Although this benchmark was done in Java programming language, it is important to

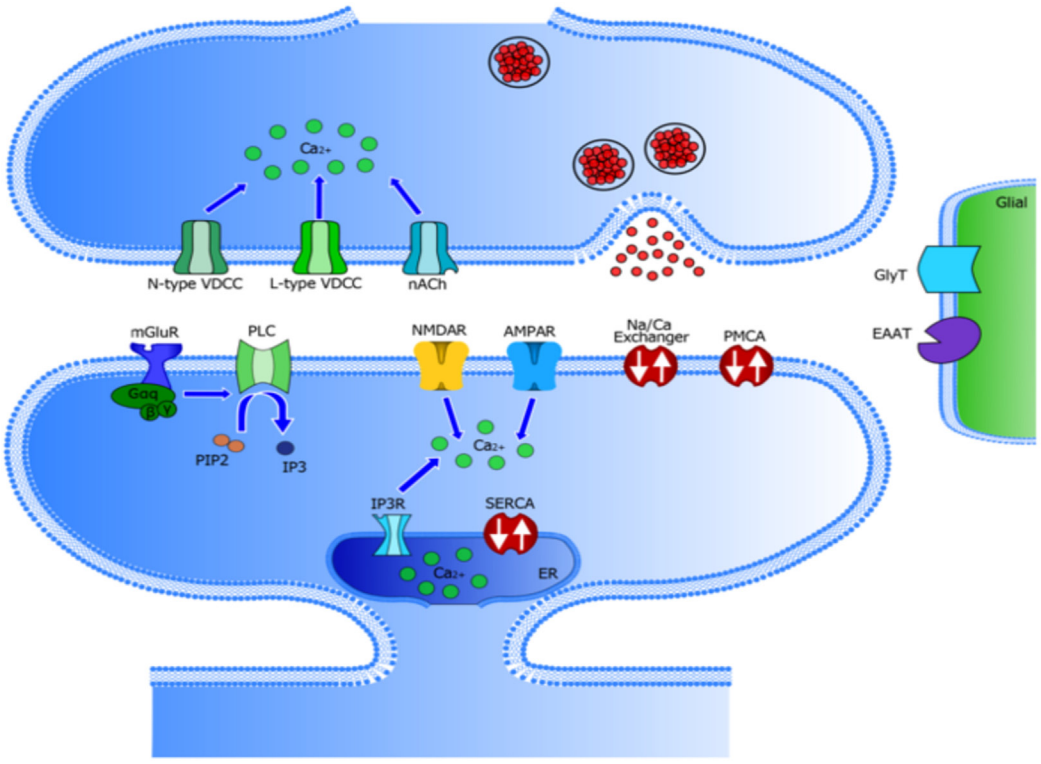

Figure 15: Simplified schematic representation of the used glutamatergic synapse model.

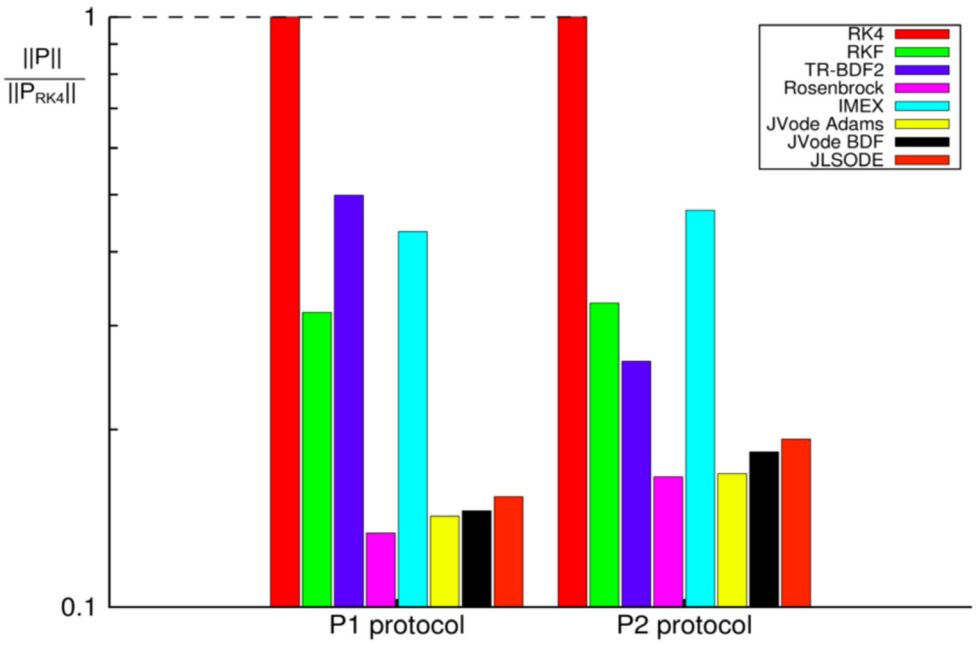

Figure 16: Normalized norm values for glutamatergic model. Normalized norm values for the 8 solver algorithms for the glutamate synapse model stimulated with P1 and P2 protocols. All ||p|| norms are normalized to the reference RK4 $||p_{RK4}||$ norm.

Model	Protocol	Tolerance	Best ||*p*||	Worst ||*p*||
Fast model:	P1	Tol_1	IMEX	TR-BDF2
AMAP7		Tol_2	IMEX	JVODE BDF
	P2	Tol_1	IMEX	TR-BDF2
		Tol_2	IMEX	JVODE BDF
Slow model:	P1	Tol_1	RKF	TR-BDF2
NMDA7		Tol_2	RKF	TR-BDF2
	P2	Tol_1	RKF	TR-BDF2
		Tol_2	RKF	JVODE BDF
Slow model:	P1	Tol_1	IMEX	RKF
NMDA15		Tol_2	IMEX	TR-BDF2
	P2	Tol_1	IMEX	RKF
		Tol_2	IMEX	TR-BDF2
Fast model:	P1	Tol_1	IMEX	TR-BDF2
$GABA_A$		Tol_2	IMEX	TR-BDF2
	P2	Tol_1	IMEX	JVODE BDF
		Tol_2	IMEX	JVODE BDF
Fast model:	P1	Tol_1	JLSODE	TR-BDF2
N-type VDCC		Tol_2	JLSODE	TR-BDF2
	P2	Tol_1	JLSODE	TR-BDF2
		Tol_2	JLSODE	TR-BDF2
Glutamatergic	P1	Tol_1	Rosenbrock	TR-BDF2
synapse	P2	Tol_1	Rosenbrock	IMEX

Summary of the best and worst solvers for the two types of models with two different stimulation protocols and tolerance values, and the applicative simulation with the glutamatergic synapse. P1 stands for the single event protocol and P2 stands for the repeated event protocol.

Table 9: Summary of the best and worst solvers.

note that the conclusions would not change with another programming language, as long as good coding practices are respected. Our overall conclusions are summarized in Table 9.

According to Table 5, TR-BDF2 was the least appropriate algorithm for all the models used in this study, with the selected stimulation protocols and tolerance parameter sets. TR-BDF2 is an ODE solver combining a trapezoidal scheme, followed by a second-order Backward Differential Formula (BDF2). As shown in our results, a relative tolerance set below or equal to $1e^{-6}$ reduced this algorithm's performance, and produced the highest overall $||p||$ norm values. This algorithm could produce better performances and accurate enough results with a larger tolerance, but a more economical (in terms of performance) solver could then give accurate enough results as well.

The IMEX algorithm was the overall best choice for the tested AMPA7, NMDA15 and GABAA models. RKF solver was appropriate for the NMDA7. JLSODE was the best choice for the fast N type Calcium Channel model and Rosenbrock solver was the most appropriate for the glutamatergic synapse model, which was considered as a complex model. However, algorithm performance variations indicated that the RKF algorithm was the most stable one, when comparing stimulation protocols and tolerance sets. In fact, RKF solver appeared to be more stable to input and tolerance sets variation compared to JLSODE, which possesses stiffness detection with our *w* preference criteria. This study clearly showed that for a kinetic model, changing the model dynamics will affect the solver performances. In fact, for our model, the dynamic appears to be more important than the underlying kinetic structure.

Synaptic elementary kinetic models are composed in most cases of time-varying continuous time control system with input ($\dot{x}=f(t,x,u)$), also called affine in control or bilinear. Due to the dependency of ODEs with respect to inputs, it is important to note that model stiffness could differ if changing the stimulus protocol. Additionally, algorithm performance could change depending on the selected performance criteria or the criteria priority level (or weight), as described by Rice [2]. The presented method gives a priori information on solvers performance using an equal priority level for all four selected criteria. Users who need to run a large number of simulations with a model (for example to perform a sensitivity analysis or to test a drug concentration effect) could use this method to optimize the computational effort.

As a comparison with other simulation platforms or software, *NEURON* [37] uses the CVODE algorithm with a manual selection of either Adams-Moulton or BDF scheme. The *Copasi* [38] tool (biochemical network simulator) uses a fourth-order hybrid Runge-Kutta algorithm, in addition to the LSODE algorithm, which integrates Petzolds stiffness detection. *BioUML* [24] developers made the same choice as *Copasi* ones, using a fourth-order hybrid Runge-Kutta algorithm, in addition to the CVODE algorithm implemented in Java language. The *Matlab* software uses the same approach as ours, which consists in integrating several resolution algorithms, and allowing the users to choose their algorithm based on the model they intend on simulating.

PYTHIA [39] is software, which provides a strict selection algorithm and tries to select a pair of machine/algorithm, in order to optimize the computational effort with a given problem. However, to use this tool, users need some expertise due to the complexity of the algorithm selection problem. As synaptic receptor models are commonly designed using SBML [36], a possible extension of this work would be to generate an automatic algorithm selection, which uses the strength of SBML. The idea would be to use SBML standard for automatically extracting the model features, and make the process completely transparent to the user.

Acknowledgements

We would like to thank Lhassane Idoumghar at the University of Haute Alsace (France) and Tameem Albash at the University of Southern California (USA, CA), for constructive discussions; the operational team of Rhenovia Pharma, which helped for the elaboration of this study. Rhenovia obtained financing by the French National Agency of Research and Technology (ANRT), with a CIFRE scholarship (432/2010) for Merdan Sarmis. The authors would like thank both anonymous referees for their valuable comments and suggestions.

References

1. Kitano H (2002) System biology: A brief overview. Science 295: 1662-1664.
2. Rice JR (1976) The algorithm selection problem. Advances in Computers 15: 65-118.
3. Ewald R (2010) Automatic algorithm selection for complex simulation problems. Rostock University Thesis, Germany.
4. Bouteiller JM, Baudry M, Allam SL, Greget RJ, Bischoff S, et al. (2008) Modeling glutamatergic synapses: Insights into mechanisms regulating synaptic ecacy. J Integr Neurosci 7: 185-197.
5. Mohler R (1973) Bilinearcontrol processes. Academic Press, Oval Road, London, UK.
6. Elliot DL (2000) Bilinearcontrol systems. Springer, New York, USA.
7. Eckel B (2005) Thinking in Java. (4th Edn), Prentice Hall PTR, Upper Saddle River, New Jersey, USA.
8. Curtiss CF, Hirschfelder JO (1952) Integration of stiff equations. Proc Natl Acad Sci U S A PMC1063538 .
9. Hairer E, Wanner G (1996) Solving ordinary differential equations II. (2nd Edn), Springer, New York, USA.
10. Shampine LF (1991) Diagnosing stiffness for runge-kutta methods. SIAM Journal on Scientific and Statistical Computing 12: 260-272.
11. Sofroniou M (2004) Construction of explicit runge-kutta pairs with stiffness detection. Mathematical and Computer Modelling 40: 1157-1169.
12. Ekeland K, Owren B, Øines E (1998) Stiffness detection and estimation of dominant spectra with explicit runge-kutta methods. ACM Transaction on Mathematical Software 24: 368-382.
13. Hairer E, Norsett SP, Wanner G (1996) Runge-Kutta and Extrapolation Methods. Solving ordinary differential equations I. (2nd Edn), Chapter 2, Springer, New York, USA.
14. Bogacki P, Shampine LF (1996) An effcient Runge-Kutta (4,5) pair. Comput Math Appl 32:15-28.
15. Shampine LF (1973) Local extrapolation in the solution of ordinary differential equations. Math Comput 27: 91-97.
16. Bank RE, Cougghram WM, Fichtner W, Grosse EH, Rose D (1985) Transient simulation of silicon devices andcircuits. IEEE Trans Electron Devices 32: 1992-2007.
17. Shampine LF, Hosea ME (1996) Analysis and implementation of TR-BDF2. Appl Numer Math 20: 21-37.
18. Dharmaraja S, Wang Y, Strang G (2009) Optimal stability for trapezoidal–backward difference split-steps. IMA Journal of Numerical Analysis 30: 141-148.
19. Shampine LF (1982) Implementation of rosenbrock methods. ACM Trans Math Softw 8: 93-113.
20. Ascher UM, Ruuth SJ, Spiteri RJ (1997) Implicit-explicit Runge-Kutta methods for time-dependent partialdifferential equations. Appl Numer Math 25: 151-167.
21. Toshiyuki K (2008) IMEX Runge–Kutta schemes for reaction–diffusion equations. Journal of Computationaland J Comput Appl Math 215: 182-195.
22. Brown PN, Hindmarsh AC (1989) Reduced storage matrix methods in stiff ODE systems. Applied Mathematics and Computation 32: 40-91.
23. Cohen SD, Hindmarsh AC (1996) CVODE, a stiff/nonstiff ODE solver in C.

Computers in Physics 10: 138-143.

24. Kolpakov FA (2004) BioUML–Open source extensible workbench for systems biology. Proceedings of BGRS, Russia.
25. Hindmarsh AC (1982) ODEPACK, A systematized collection of ODE solvers. IMACS Transactions on Scientific Computation 1: 55-64.
26. Uteshev V, Pennefather PS (1997) Analytical description of the activation of multi-state receptors by continuous neurotransmitter signals at brain synapses. Biophys J 72: 1127-1134.
27. Savtchenko LP (2007) Bilateral processing in chemical synapses with electrical 'ephaptic' feedback: A theoretical model. Math Biosci 207: 113-137.
28. Petzold LR (1983) Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. SIAM J Sci Comput 4: 136-148.
29. Schorge S, Elenes S, Colquhoun D (2005) Maximum likelihood fitting of single channel NMDA activity witha mechanism composed of independent dimers of subunits. J Physiol 562: 395-418.
30. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application toconduction and excitation in nerve. J Physiol 117: 500-544.
31. Jaffe DB, Ross WN, Lisman JE, Lasser-Ross N, Miyakawa H, et al. (1994) A model for dendritic Ca^{2+}accumulation in hippocampal pyramidal neurons based on fluorescence imaging measurements. J Neurophysiol 71: 1065-1077.
32. Poirazi P, Brannon T, Mel BW (2003) Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell. Neuron 37: 977-987.
33. Greget R, Pernot F, Bouteiller JMC, Ghaderi V, Allam S, et al. (2011) Simulation of postsynaptic glutamate receptors reveals critical features of glutamatergic transmission. PLoS ONE 6: e28380.
34. Hall G, Suleiman MB (1985) Asingle code for the solution of stiff and nonstiff ODE's. SIAM J Sci and Stat Comput 6: 684-697.
35. Pugh JR, Raman IM (2005) GABAA receptor kinetics in the cerebellar nuclei: Evidence for detection of transmitter from distant release sites. Biophys J 88:1740-1754.
36. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, et al. (2003) The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics 19: 524-531.
37. Hines ML, Carneval NT (1997) The NEURON simulation environment. Neural Comput 9: 1179-1209.
38. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, et al. (2006) COPASI–A COplex Pathway Simulator. Bioinformatics 22: 3067-3074.
39. Weerawarana S, Houstis EN, Rice JR, Joshi A, Houstis CE (1996) PYTHIA: A knowledge based system to select scientific algorithms. ACM Trans Math Softw 22: 447-468.