



HAL
open science

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

Merdan Sarmis, Jean-Marie C. Bouteiller, Nicolas Ambert, Arnaud Legendre,
Serge Bischoff, Olivier Haeberlé, Michel Baudry

► **To cite this version:**

Merdan Sarmis, Jean-Marie C. Bouteiller, Nicolas Ambert, Arnaud Legendre, Serge Bischoff, et al.. Assessing Numerical Resolution Methods Performance for Kinetic Models of Receptors and Channels. *Journal of Computer Science & Systems Biology*, 2013, 6 (3), pp.150-164 (Article). 10.4172/jcsb.1000112 . hal-00984346

HAL Id: hal-00984346

<https://hal.science/hal-00984346>

Submitted on 28 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assessing numerical resolution methods performance for kinetic models of receptors and channels

Merdan Sarmis^{1,2}, Jean-Marie C. Bouteiller^{1,3}, Nicolas Ambert¹, Arnaud Legendre^{1,2}, Serge Bischoff¹,
Olivier Haerberlé² and Michel Baudry^{1,4}

¹ Rhenovia Pharma SA, Mulhouse, France;

² Laboratoire MIPS EA2332, Université de Haute Alsace, Mulhouse, France;

³ Dep. of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA

⁴ Western University of Health Sciences, Pomona, CA

Send correspondence to:

Michel Baudry
Rhenovia, Pharma
20C Rue de Chemnitz
F-68200 MULHOUSE
Phone: (+33) 389 321 180
Fax: (+33) 389 555 145
Email: michel.baudry@rhenovia.com

Or

Jean-Marie C. Bouteiller
Rhenovia, Pharma
20C Rue de Chemnitz
F-68200 MULHOUSE
Phone: (+33) 389 321 180
Fax: (+33) 389 555 145
Email: jean-marie.bouteiller@rhenovia.com

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.

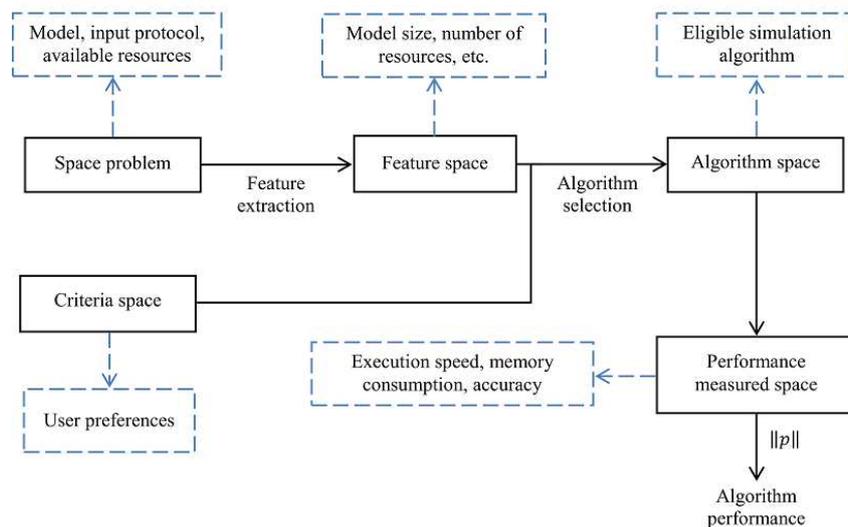


Figure 1: Algorithm selection problem. The algorithm selection problem was defined by Rice in 1976, and completed by Ewald [3]. As presented in this scheme ([3] page 22), users need to know model inputs, characteristics (size) and algorithms structure in order to select the right algorithm for a simulation.

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 $x(t) \in \mathfrak{R}^n$ is the state variable vector and $u(t) \in \mathfrak{R}^m$ 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.

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

We compared eight ODE solvers (Table 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 Curtis 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, 11, 12]. This implies that the stiffness of a model is closely related to the selected numerical resolution method. According to Ekeland, the precise definition of the stiffness of a system of equations is not crucial from a practical point of view [12]. 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	TR-BDF2	Rosenbrock
RKF	JVODE BDF	IMEX
JVODE ADAMS	JLSODE	
JLSODE		

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

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* [24] platform developers. 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 Linda R. Petzold [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.

Material 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 $k_1 \times R \times Glu$.

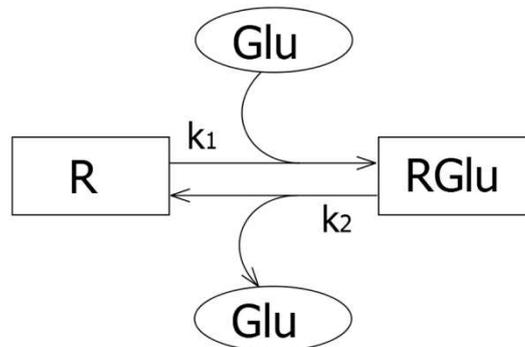


Figure 2: Example of an elementary kinetic model for ligand/receptor binding.

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

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 glutamate 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 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.

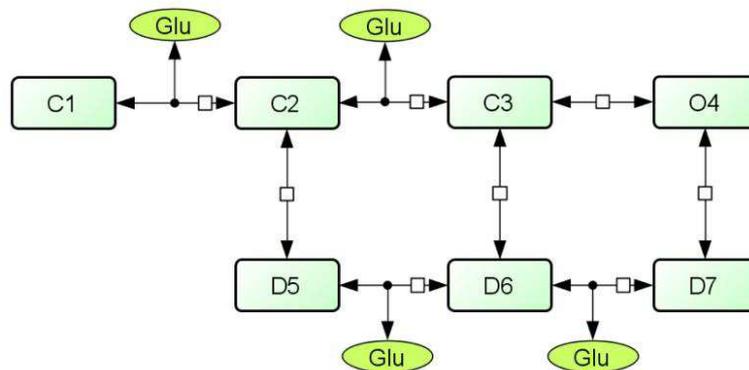


Figure 3: Kinetic scheme of the AMPA7 / NMDA7 model depending to the set of parameters. The rectangles represent the different state variables. The green ovals named *Glu* represent the input of the model and the location with where it interacts in the model.

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

Table 2: Used models features for this study. For each used model we extract the same features: state size, number of equations, number of parameters, number of inputs and dynamic rate times.

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].

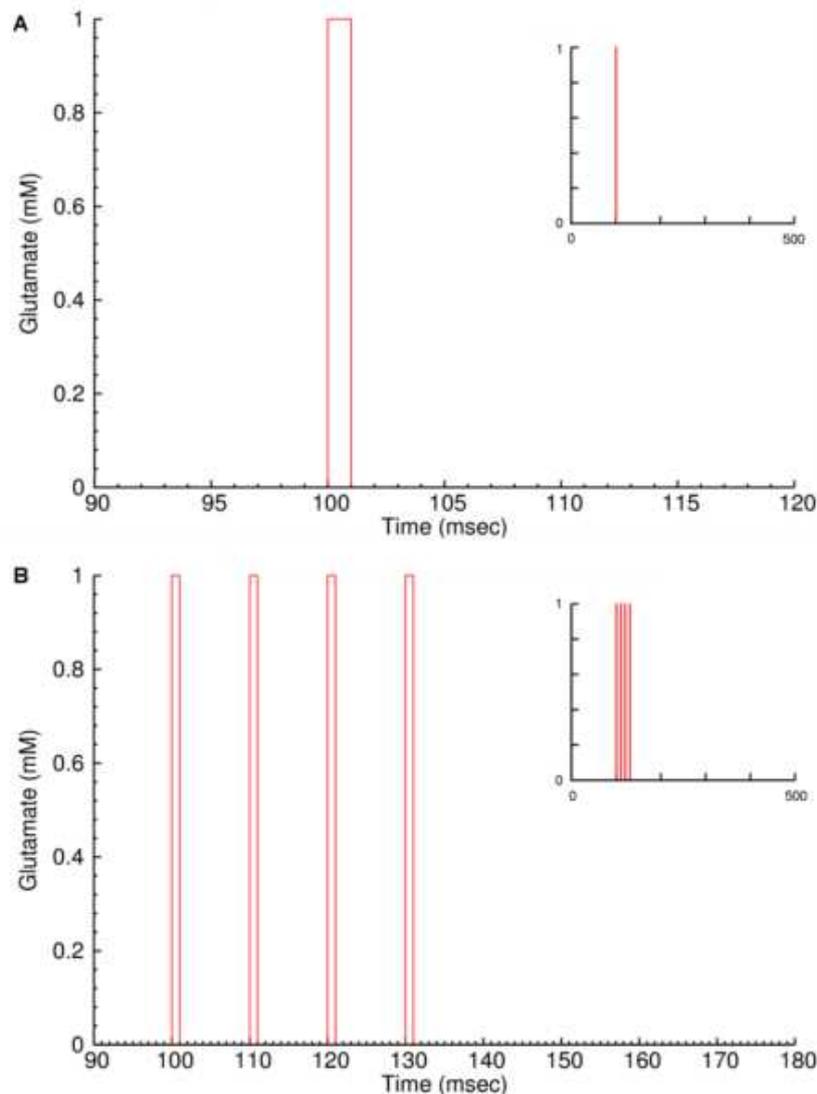


Figure 4: Stimulation protocols. A: Single event protocol $P1$, B: Repeated event protocol $P2$. Inserts represent the stimulus embedded in the entire simulation run.

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 $0.5 \mu\text{sec}$ 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 satisfying 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.

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 Table 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 Table S–5. The performance values are presented on Table 3 for the AMPA7 and on Table 4 for the NMDA7 model.

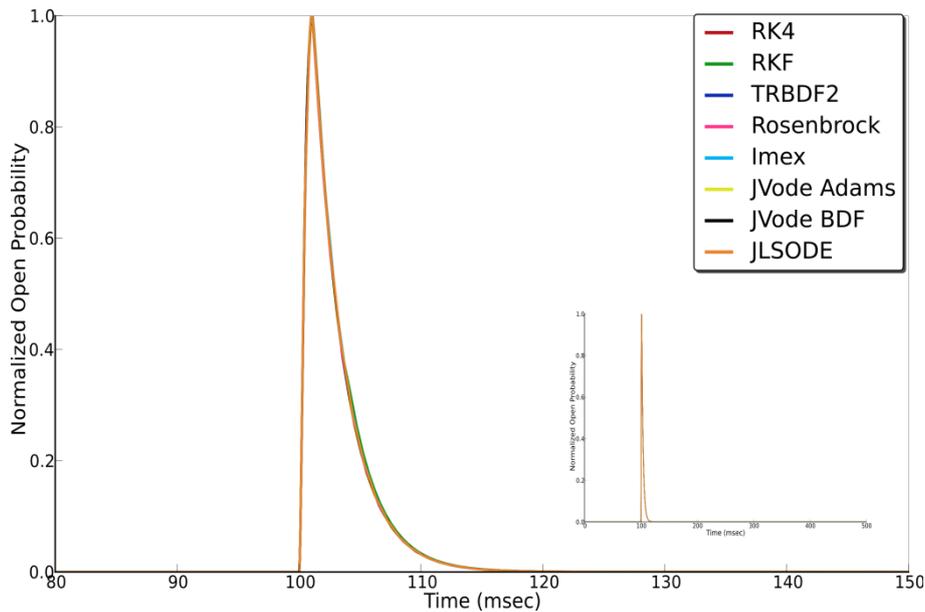


Figure 5: Normalized open state probability for AMPA7 model. Insert represents the results embedded in the entire simulation run.

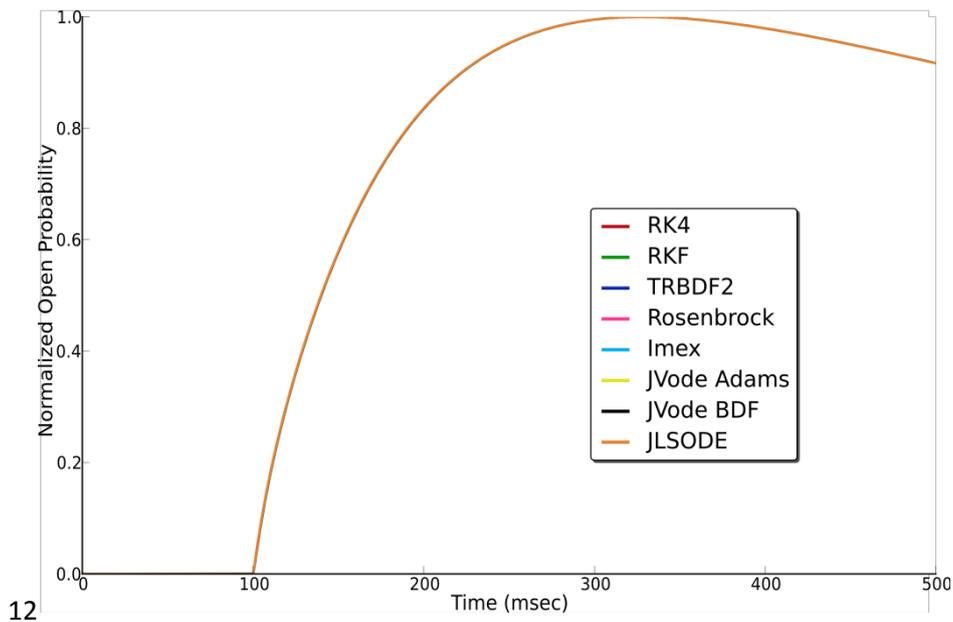


Figure 6: Normalized open state probability for NMDA7 model.

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 worst performers. With the second protocol (Figure 7D), we observe the same tendency. 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.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE DAMS	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
Δ P1 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 pulse)								
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

Table 3: Quantification and comparison of normalized solver performances for the AMPA7 receptor model. Tol₁ and Tol₂ represent the $||p||$ norm value for each algorithm with P1 and P2 protocols. Δ P1Tol₁/Tol₂ represents the absolute difference between Tol₁ and Tol₂ for P1, whereas Δ P2 Tol₁/Tol₂ represents the same absolute difference for P2. Δ Tol₁ P1/P2 represents the absolute difference between P1 and P2 for Tol₁ parameters, whereas Δ Tol₂ P1/P2 represents the same absolute difference for Tol₂.

	RK4	RKF	TR-BDF2	Rosen-brock	IMEX	JVODE ADAMS	JVODE BD	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
Δ P1 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 pulse)								
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

Table 4: Quantification and comparison of normalized solver performances for the NMDA7 receptor model. Tol₁ and Tol₂ represent the $||p||$ norm value for each algorithm with P1 and P2 protocols. Δ P1Tol₁/Tol₂ represents the absolute difference between Tol₁ and Tol₂ for P1, whereas Δ P2 Tol₁/Tol₂ represents the same absolute difference for P2. Δ Tol₁ P1/P2 represents the absolute difference between P1 and P2 for Tol₁ parameters, whereas Δ Tol₂ P1/P2 represents the same absolute difference for Tol₂.

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 models 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 JNODE BDF and TR-BDF2 are the most sensitive.

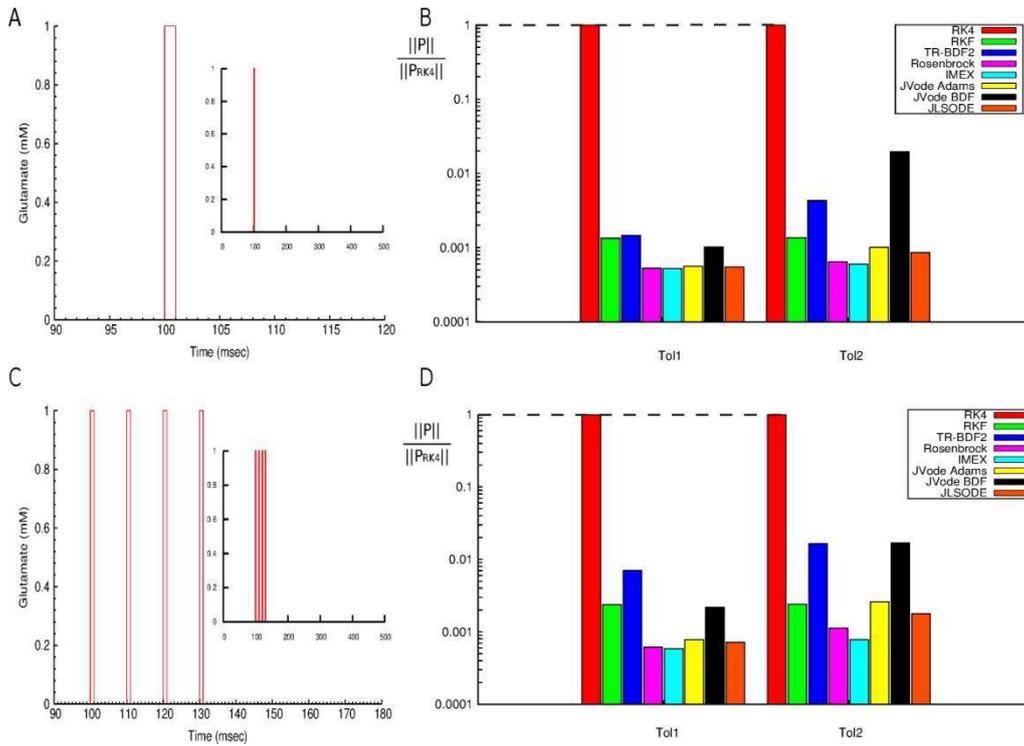


Figure 7: Normalized norm values for the AMPA7 receptor model. A: Review of the protocol $P1$. B: The normalized norm values for the 8 solver algorithms with $P1$. C: Review of the protocol $P2$. D: The normalized norm values for the 8 solver algorithms with $P2$. All $||\rho||$ norms are normalized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

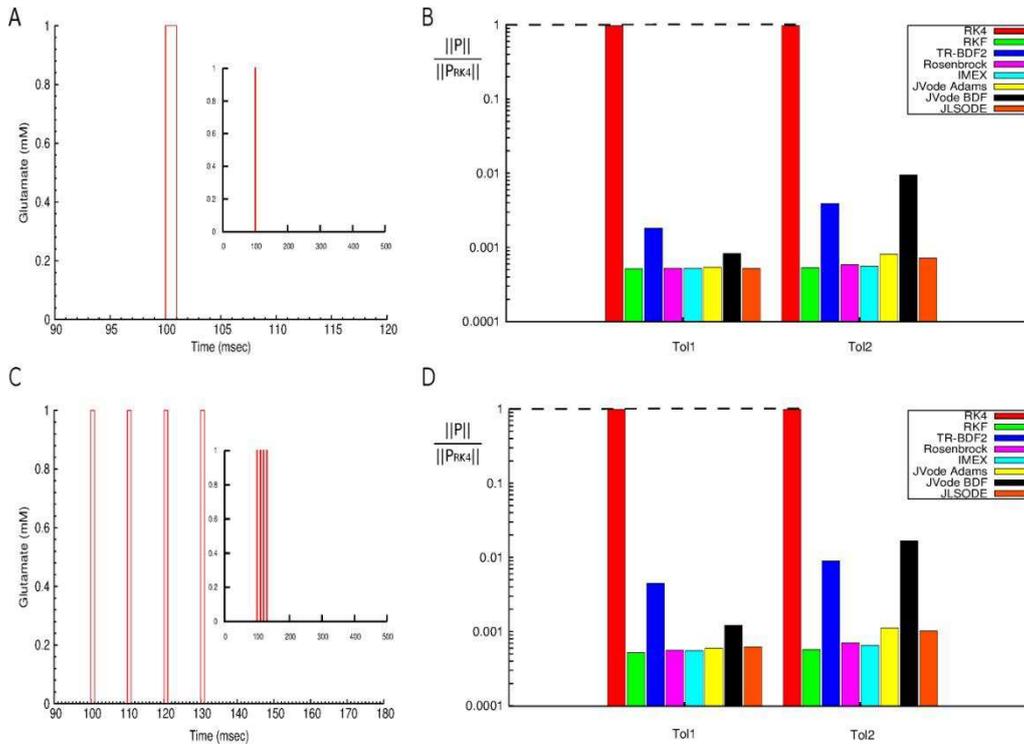


Figure 8: Normalized norm values for the NMDA7 receptor model. A: Review of the protocol $P1$. B: The normalized norm values for the 8 solver algorithms with $P1$. C: Review of the protocol $P2$. D: The normalized norm values for the 8 solver algorithms with $P2$. All $||\rho||$ norms are normalized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

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 Tables S-6 and S-7 of the supplemental document. Algorithm performance values for the NMDA15 receptor model are quantified in Table 5.

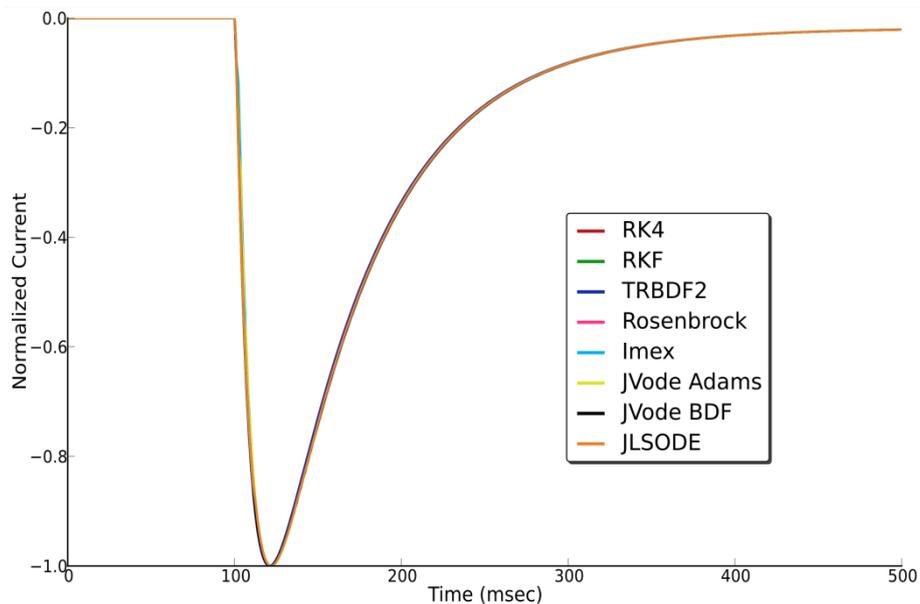


Figure 9: Normalized NMDA15 receptor current. NMDA15 receptor's current resulting from a single pulse protocol generated by all solver algorithms.

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 (Figures 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.

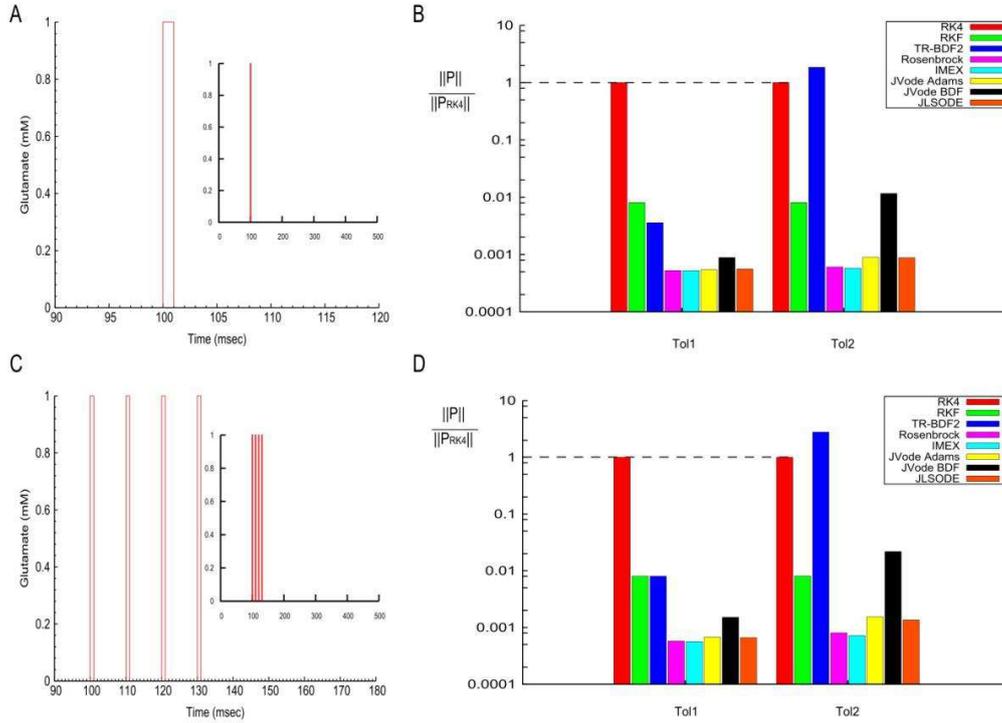


Figure 10: Normalized norm values for the NMDA15 receptor model. A: Review of the protocol $P1$. B: The normalized norm values for the 8 solver algorithms with $P1$. C: Review of the protocol $P2$. D: The normalized norm values for the 8 solver algorithms with $P2$. All $\|p\|$ norms are normalized to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

	RK4	RKF	TR-BDF2	Rosenbrock	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
$\Delta P1$ 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
$\Delta P2$ 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

Table 5: Quantification and comparison of normalized solver performances for the NMDA15 model. Tol₁ and Tol₂ represent the $\|p\|$ norm value for each algorithm with P1 and P2 protocols. $\Delta P1$ Tol₁/Tol₂ represents the absolute difference between Tol₁ and Tol₂ for P1, whereas $\Delta P2$ Tol₁/Tol₂ represents the same absolute difference for P2. Δ Tol₁ P1/P2 represents the absolute difference between P1 and P2 for Tol₁ parameters, whereas Δ Tol₂ P1/P2 represents the same absolute difference for Tol₂.

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 presented in [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 Tables 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.

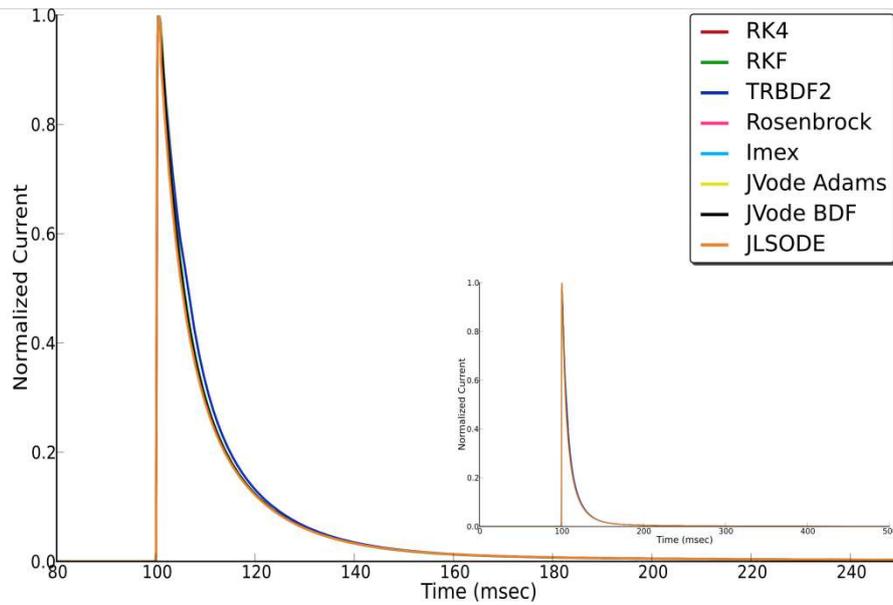


Figure 11: Normalized GABA_A receptor model current. GABA_A receptor's current resulting from the single pulse protocol generated by all solvers. Insert represents the results embedded in the entire simulation run.

	RK4	RKF	TR-BDF2	Rosenbrock	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
ΔP1 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
Δ Tol ₁ P1/P2	-	7.6e ⁻⁴	4.6e ⁻⁴	7.5e ⁻⁵	6e ⁻⁵	1.74e ⁻⁴	1.08e ⁻³	2.65e ⁻⁴
Δ Tol ₂ P1/P2	-	11e ⁻⁴	2.5e ⁻³	3.81e ⁻⁴	2.74e ⁻⁴	6.6e ⁻⁴	1.6e ⁻²	8.12e ⁻⁴

Table 6: Quantification and comparison of normalized solver performances for the GABA_A model. Tol₁ and Tol₂ represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1Tol₁/Tol₂ represents the absolute difference between Tol₁ and Tol₂ for P1, whereas Δ P2 Tol₁/Tol₂ represents the same absolute difference for P2. Δ Tol₁ P1/P2 represents the absolute difference between P1 and P2 for Tol₁ parameters, whereas Δ Tol₂ P1/P2 represents the same absolute difference for Tol₂.

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 JNODE 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 (JNODE BDF and TR-BDF2) for this model are the most sensitive to input and tolerance variations. With the GABA_A model and the tested situations, IMEX and RKF yield the most consistent performances.

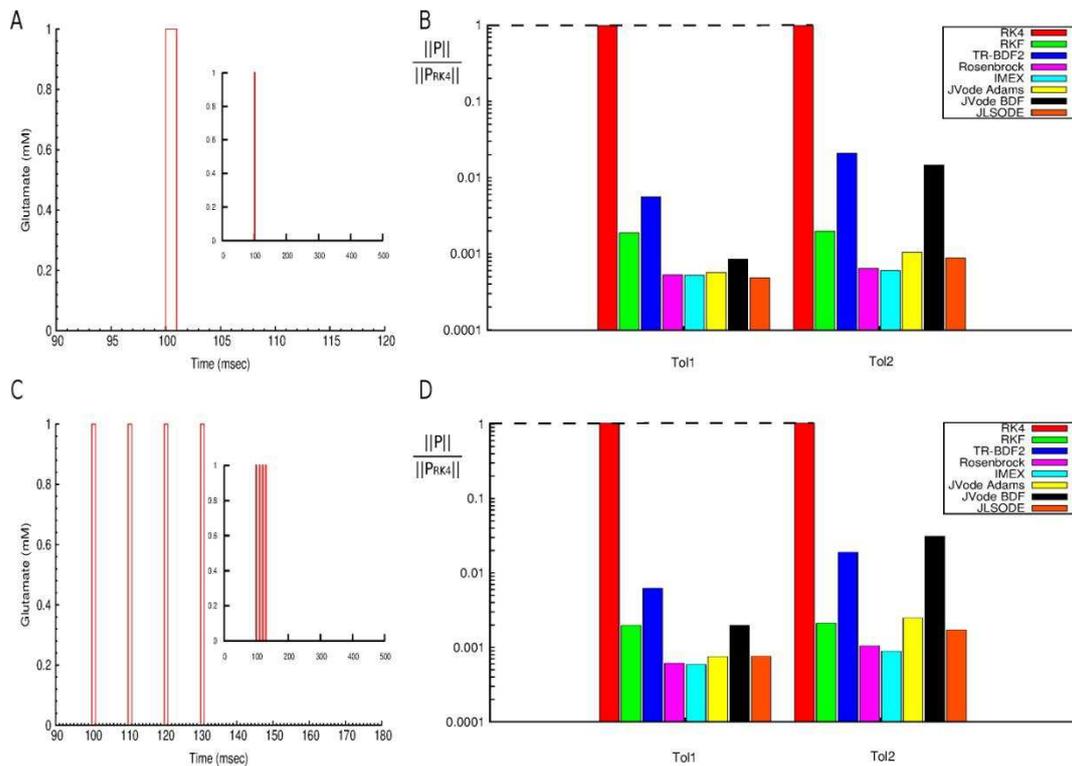


Figure 12: Normalized norm values for the GABA_A receptor model. A: review of the protocol *P1*. B: The normalized norm values for the 8 solver algorithms with *P1*. C: review of the protocol *P2*. D: The normalized norm values for the 8 solver algorithms with *P2*. All $||p||$ norms are normalized to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

In fact, from Table S–8 an 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.

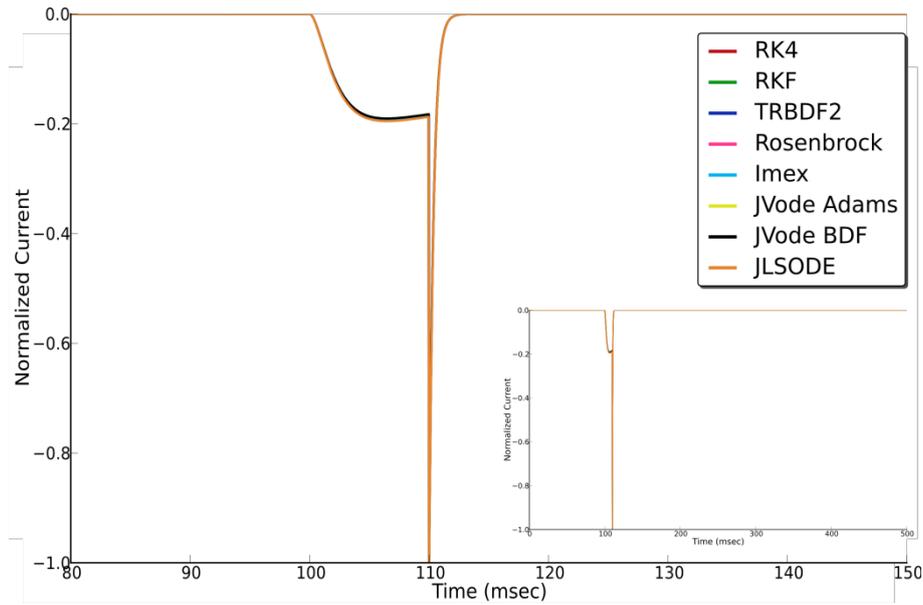


Figure 13: N-type VDCC current. N-type VDCC's current resulting from a single event protocol generated by all ODE solvers. Insert represents the results embedded in the entire simulation run.

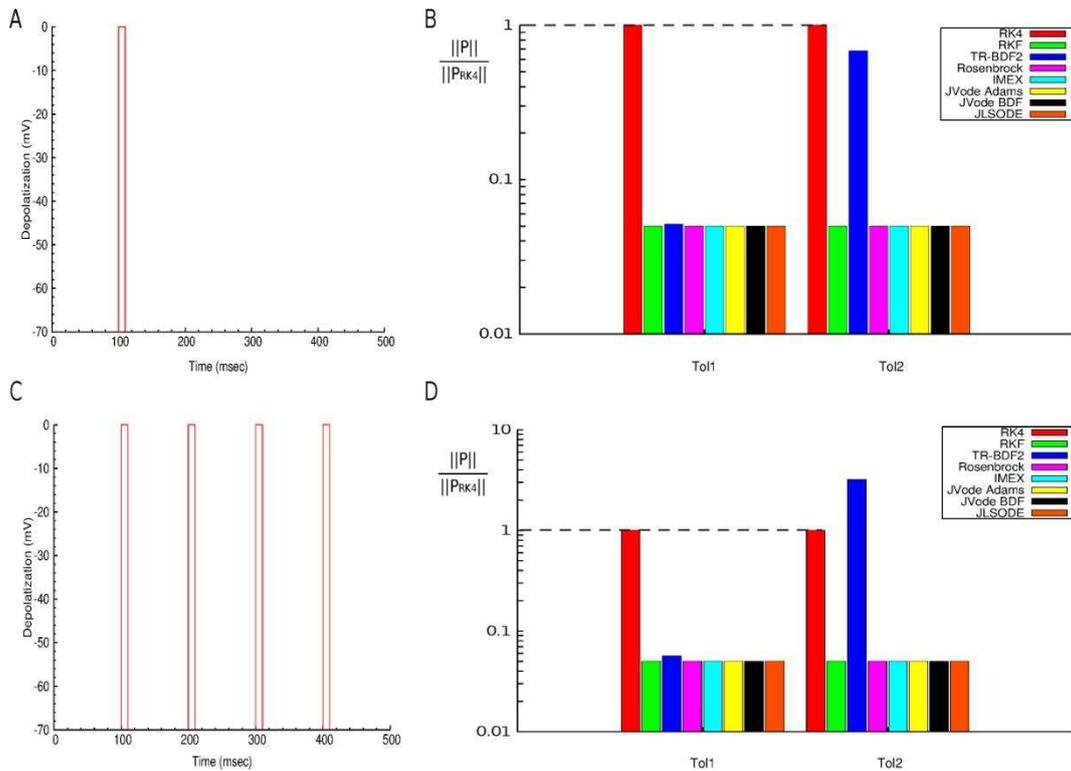


Figure 14: Normalized norm values for the N type VDCC model. A: review of the protocol $P1$. B: The normalized norm values for the 8 solver algorithms with $P1$. C: review of the protocol $P2$. D: The normalized norm values for the 8 solver algorithms with $P2$. All $||p||$ norms are normalized to the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

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).

	RK4	RKF	TR-BDF2	Rosenbrock	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
Δ 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
Δ Tol ₂ P1/P2	-	6e ⁻⁶	2.5322	1.102e ⁻⁵	1.213e ⁻⁵	3.006e ⁻⁵	3e ⁻⁵	6.01e ⁻⁶

Table 7: Quantification and comparison of normalized solver performances for the N-type VDCC model. Tol₁ and Tol₂ represent the ||p|| norm value for each algorithm with P1 and P2 protocols. Δ P1Tol₁/Tol₂ represents the absolute difference between Tol₁ and Tol₂ for P1, whereas Δ P2 Tol₁/Tol₂ represents the same absolute difference for P2. Δ Tol₁ P1/P2 represents the absolute difference between P1 and P2 for Tol₁ parameters, whereas Δ Tol₂ P1/P2 represents the same absolute difference for Tol₂.

When we analyzed the criteria quantification table (see 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 Tol₂ parameters (10 times more for P1 and 64 times more for P2). In addition, all solvers generated almost the same NRMSD, except for RKF (for both stimulation protocols) and TR-BDF2 (for Tol₂ tolerance set).

As illustrated in the supplemental file (Figure S–1), when switching from Tol₁ to Tol₂, 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 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 (Tol₂) 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 (P_1 and P_2). Quantified criteria are presented in the supplemental file by Table S–12.

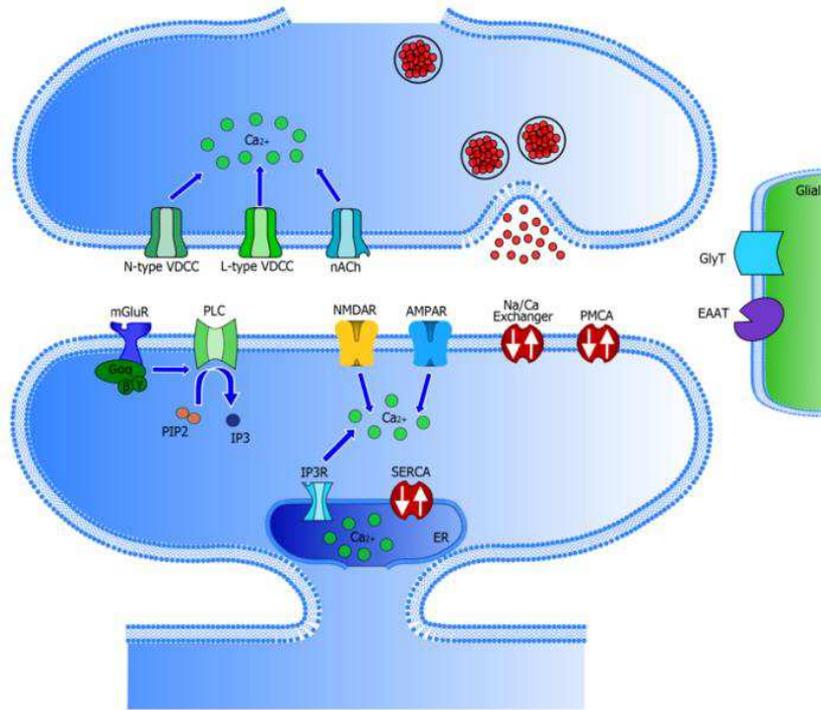


Figure 15: Simplified schematic representation of the used glutamatergic synapse model.

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.

	RK4	RKF	TR-BDF2	Rosenbrock	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
$\Delta Tol_1_{P_1/P_2}$	-	$1.16e^{-2}$	0.2379	$3.291e^{-2}$	$3.773e^{-2}$	$4.585e^{-2}$	$3.753e^{-2}$	$3.886e^{-2}$

Table 8: Quantification and comparison of normalized solver performances for the glutamate synapse model. Tol_1 and Tol_2 represent the $\|p\|$ norm value for each algorithm with P_1 and P_2 protocols. $\Delta Tol_1_{P_1/P_2}$ represents the absolute difference between P_1 and P_2 for Tol_1 parameters.

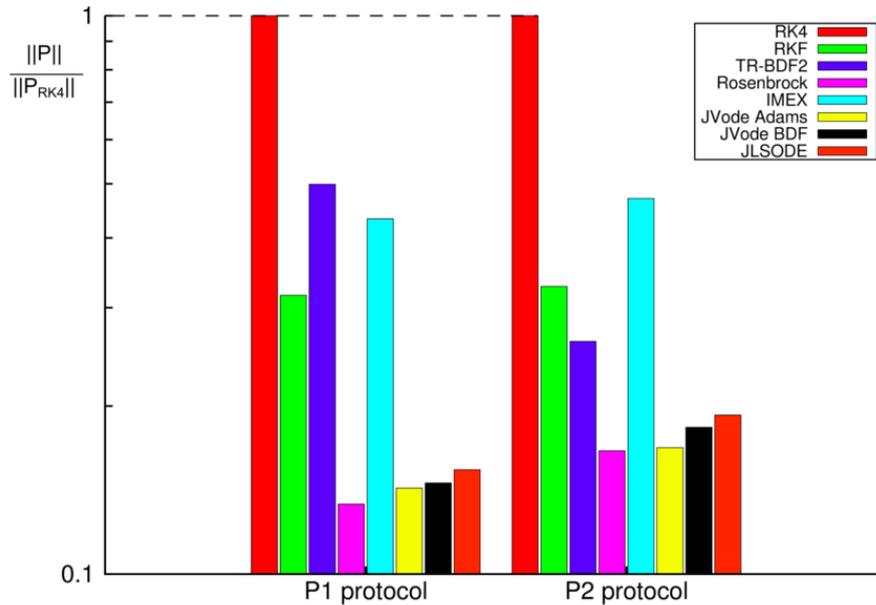


Figure 16: Normalized norm values for glutamatergic model. Normalized norm values for the 8 solver algorithms for the glutamate synapse model stimulated with *P1* and *P2* protocols. All $\|p\|$ norms are normalized to the reference RK4 $\|p_{RK4}\|$ norm.

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 stimulations 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 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 9, 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.

Model	Protocol	Tolerance	Best $ p $	Worst $ p $
Fast model: AMAP7	P1	Tol ₁	IMEX	TR-BDF2
		Tol ₂	IMEX	JVODE BDF
	P2	Tol ₁	IMEX	TR-BDF2
		Tol ₂	IMEX	JVODE BDF
Slow model: NMDA7	P1	Tol ₁	RKF	TR-BDF2
		Tol ₂	RKF	TR-BDF2
	P2	Tol ₁	RKF	TR-BDF2
		Tol ₂	RKF	JVODE BDF
Slow model: NMDA15	P1	Tol ₁	IMEX	RKF
		Tol ₂	IMEX	TR-BDF2
	P2	Tol ₁	IMEX	RKF
		Tol ₂	IMEX	TR-BDF2
Fast model: GABA _A	P1	Tol ₁	IMEX	TR-BDF2
		Tol ₂	IMEX	TR-BDF2
	P2	Tol ₁	IMEX	JVODE BDF
		Tol ₂	IMEX	JVODE BDF
Fast model: N-type VDCC	P1	Tol ₁	JLSODE	TR-BDF2
		Tol ₂	JLSODE	TR-BDF2
	P2	Tol ₁	JLSODE	TR-BDF2
		Tol ₂	JLSODE	TR-BDF2
Glutamatergic synapse	P1	Tol ₁	Rosenbrock	TR-BDF2
	P2	Tol ₁	Rosenbrock	IMEX

Table 9: Summary of the best and worst solvers. 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.

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 a 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.

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.
4. Bouteiller JM, Baudry M, Allam SL, Greget RJ, Bischoff S et al. (2008) Modeling Glutamatergic Synapses: Insights Into Mechanisms Regulating Synaptic Efficacy. *Journal of integrative neuroscience* 7: 185-197.
5. Mohler R (1973) *Bilinear Control Processes*, Academic Press, Oval Road, London.
6. Elliot DL (2000) *Bilinear Control Systems*, Springer, Dordrecht, Heidelberg, London, New York.
7. Eckel B (2006) *Thinking in Java. (4th Edn)*, Prentice Hall PTR, Upper Saddle River, New Jersey.
8. Curtiss CF, Hirschfelder JO (1952) Integration of Stiff Equations. *Proceedings of the National Academy of Sciences of the United States of America*.

9. Hairer E, Wanner G (1996) Solving Ordinary Differential Equations II. (2nd Edn) Springer, Heidelberg, Dordrecht, London, New York.
10. Shampine LF (1991) Diagnosing Stiffness for Runge-Kutta Methods. *SIAM Journal on Scientific and Statistical Computing* 12: 260-272.
11. Sofroniou M (2004) Constructing of Explicit Runge-Kutta Pairs with Stiffness Detection. *Mathematical and Computer Modelling* 40: 1157-1169.
12. Ekeland K, Owren B and Øines E (1998) Stiffness Detection and Estimation of Dominant Spectra with Explicite Runge-Kutta Methods. *ACM Transaction on Mathematical Software* 24: 368-382.
13. Hairer E, Norsett SP, Wanner G (1996) Solving Ordinary Differential Equations I. Chapter 2: Runge-Kutta and Extrapolation Methods. (2nd Edn) Springer, Heidelberg, Dordrecht, London, New York.
14. Bogacki P, Shampine LF (1996) An efficient Runge-Kutta (4,5) pair. *Computers & Mathematics with Applications* 32:15-28.
15. Shampine LF (1973) Local Extrapolation in the Solution of Ordinary Differential Equations. *Mathematics of Computation* 27: 91-97.
16. Bank RE, Coughram WM, Fichtner W, Grosse EH, Rose D (1985) Transient simulation of silicon devices and circuits. *IEEE Transactions on electron devices* 32: 1992-2007.
17. Shampine LF, Hosea ME (1996) Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics* 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 Transactions on Mathematical Software* 8: 93-113.
20. Ascher UM, Ruuth SJ, Spiteri RJ (1997) Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics* 25: 151-167.
21. Toshiyuki K (2008) IMEX Runge–Kutta schemes for reaction–diffusion equations. *Journal of Computational and Applied Mathematics* 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*.

25. Hindmarsh AC (1982) ODEPACK, A Systematized Collection Of ODE Solvers. *IMACS Transactions on Scientific Computation* 1: 55-64.
26. Petzold LR (1983) Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations. *Society for Industrial and Applied Mathematics* 4: 136-148.
27. Uteshev V and Pennefather PS (1997) Analytical Description of the Activation of Multi-State Receptors by Continuous Neurotransmitter Signals at Brain Synapses. *Biophysical Journal* 72: 1127-1134.
28. Savtchenko LP (2007) Bilateral processing in chemical synapses with electrical 'ephaptic' feedback: a theoretical model. *Mathematical Biosciences* 207: 113-137.
29. Schorge S, Elenes S, Colquhoun D (2005) Maximum likelihood fitting of single channel NMDA activity with a mechanism composed of independent dimers of subunits. *The Journal of Physiology* 562: 395-418.
30. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117: 500-544.
31. Jaffe DB, Ross WN, Lisman JE, Lasser-Ross N, Miyakawa H, et al. (1994) A model for dendritic Ca²⁺ accumulation in hippocampal pyramidal neurons based on fluorescence imaging measurements. *Journal of Neurophysiology* 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) A Single Code for the Solution of Stiff and Nonstiff ODE's. *SIAM Journal on Scientific and Statistical Computing* 6: 684-697.
35. Pugh JR and Raman IM (2005) GABAA Receptor Kinetics in the Cerebellar Nuclei: Evidence for Detection of Transmitter from Distant Release Sites. *Biophysical Journal* 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 Computation* 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 and Houstis CE (1996) PYTHIA: A Knowledge Based System to Select Scientific Algorithms. *ACM* 22: 447-468.

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.

Supplemental Files

Assessing numerical resolution method performance for kinetic models of receptors and channels: Supplemental file. Additional information and details are presented on this additional file about the simulation, protocols and numerical resolution methods to ensure reproducibility of our results. All quantified results are depicted and detailed.

Assessing numerical resolution methods performance for kinetic models of receptors and channels: Supplemental file

Merdan Sarmis^{*1,2}, Jean-Marie C. Bouteiller^{1,3}, Nicolas Ambert¹, Arnaud Legendre^{1,2},
Serge Bischoff¹, Olivier Haeberlé² and Michel Baudry^{1,4}

¹ Rhenovia Pharma SA, 20c rue de chemnitz 68200 Mulhouse, France

² MIPS Laboratory EA2332, University of Haute-Alsace, 12 rue des Frères Lumière 68100 Mulhouse , France

³ Dep. of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA

⁴ Western University of Health Science, Pomona, CA, USA

Email:

Merdan Sarmis* - merdan.sarmis@rhenovia.com;

Jean-Marie C. Bouteiller - jean-marie.bouteiller@rhenovia.com;

Nicolas Ambert - nicolas.ambert@rhenovia.com;

Arnaud Legendre - arnaud.legendre@rhenovia.com;

Serge Bischoff - serge.bischoff@rhenovia.com;

Olivier Haeberlé - olivier.haeberle@uha.fr;

Michel Baudry - michel.baudry@rhenovia.com.

* Corresponding author

This document supplements the manuscript entitled “*Assessing numerical resolution methods performance for kinetic models of receptors and channels*” and provides detailed explanation regarding the reproducibility of the results.

The Numerical Resolution Methods

We implemented the fourth-order Runge-Kutta solver (RK4), as indicated in the book edited by Hairer, Norsett and Wanner (referenced in the main article). We did not modify the parameters of this algorithm, which is very easy to implement.

The Runge-Kutta Fehlberg solver (RKF) is a fourth/fifth-order explicite RK-scheme. We implemented it as presented by Bogacki and Shampine (referenced in the main article). The c parameters for RKF solver at different stages are presented in Table S-1, and the stage equations are:

$$k_1 = f(t, y_n) \tag{S-1}$$

$$k_2 = f(t + c_{20}.h, y_n + c_{21}.h.k_1) \tag{S-2}$$

$$k_3 = f(t + c_{30}.h, y_n + c_{31}.h.k_1 + c_{32}.h.k_2) \tag{S-3}$$

$$k_4 = f(t + c_{40}.h, y_n + c_{41}.h.k_1 + c_{42}.h.k_2 + c_{43}.h.k_3) \tag{S-4}$$

$$k_5 = f(t + c_{50}.h, y_n + c_{51}.h.k_1 + c_{52}.h.k_2 + c_{53}.h.k_3) + c_{54}.h.k_4 \tag{S-5}$$

$$k_6 = f(t + c_{60}.h, y_n + c_{61}.h.k_1 + c_{62}.h.k_2 + c_{63}.h.k_3) + c_{64}.h.k_4 + c_{65}.h.k_5 \tag{S-6}$$

$c_{20} = \frac{1}{4}$	$c_{21} = \frac{1}{4}$				
$c_{30} = \frac{3}{8}$	$c_{31} = \frac{3}{32}$	$c_{32} = \frac{9}{32}$			
$c_{40} = \frac{12}{13}$	$c_{41} = \frac{1932}{2197}$	$c_{42} = \frac{-7200}{2197}$	$c_{43} = \frac{7296}{2197}$		
$c_{50} = 1$	$c_{51} = \frac{439}{216}$	$c_{52} = -8$	$c_{53} = \frac{3680}{513}$	$c_{54} = \frac{-845}{4104}$	
$c_{60} = \frac{1}{2}$	$c_{61} = \frac{-8}{27}$	$c_{62} = 2$	$c_{63} = \frac{-3544}{4104}$	$c_{64} = \frac{1859}{4104}$	$c_{65} = \frac{-11}{40}$

Table S-1: RKF ODE solver c parameters values used for this study.

The TR-BDF2 algorithm corresponds to the *ode23tb* function in the Matlab software with the γ parameter equal to $(2 - \sqrt{2})$, as explained by S. Dharmaraja who is referenced in the main article. In the original version of this algorithm, the authors describe a t_{stop} parameter, which readjusts its value for the next calculated step-size before a large variation. Since in our case, we cannot forecast future variations, we transformed this equation into equation (S-7), which better fits our simulations.

$$h_{n+1} = \frac{t - t_{stop}}{\lceil \frac{t - t_{stop}}{h_{n+1}} + \epsilon \rceil} \tag{S-7}$$

The IMEX and Rosenbrock solvers are both fourth-order, RK-scheme-based hybrid solvers. For each computation step, hybrid solvers resolve the system twice, first with an explicit scheme, then with an implicit one. The two results are compared and if they are similar, the step is accepted. If not, the step-size is reduced and the computation step restarts. These two solvers are implemented as described in their respective references (see main article).

JVODE is a Java version of the CVODE solver. This is the only solver that we did not implement, as it is provided as open source files by the BioUML developers. This solver is in fact composed of two solvers: 1) an explicit Adams-Moulton method with variable order (from 1 to 12) for non-stiff systems and 2) an implicit Backward Differential Formula (BDF scheme) with variable order (from 1 to 5) for stiff systems. These two solvers are called JVODE ADAMS and JVODE BDF, respectively.

The last algorithm we consider is JLSODE, a Java version of the LSODE solver, which has the same Adams-Moulton and BDF schemes as JVODE. This last solver adds stiffness detection, as written by Linda R. Petzold (referenced in the main article). With this stiffness detection, JLSODE starts with the Adams-Moulton scheme and switches to the BDF scheme if the system becomes stiff, and vice-versa. In this case, we used the Adams-Moulton and BDF solvers integrated in JVODE and added stiffness detection, as described by Linda R Petzold (referenced in the main article).

Material and Methods

To simplify the study and its reproducibility, we used sequential computations, even if the platform works on parallel computing with MPI library. The two tolerances, $Rtol$ and $Atol$, are used to validate a computed step for the numerical ODE resolution methods. All simulations are run ten times and the presented quantitative results are the averages of the ten simulation results. All solvers are variable step-size except for the reference solver RK4, which uses a $5 \mu s$ constant step-size. To compute the accuracy criteria, we did an interpolation of all solver's results for the reference times.

In addition, in order to determine the sensitivity to stimulation protocol and tolerance parameters, we computed the following performance differences: 1) the differences between Tol_1 and Tol_2 for the first and second protocol, respectively called $\Delta P1_{Tol_1/Tol_2}$ and $\Delta P2_{Tol_1/Tol_2}$ and 2) the differences between the first and second protocol for Tol_1 and Tol_2 parameters, respectively called $\Delta Tol_{1_{P1/P2}}$ and $\Delta Tol_{2_{P1/P2}}$.

AMPA7/NMDA7 receptor model

The kinetic model presented in Figure 3 of the main document could model either an AMPA or a NMDA receptor, depending on the set of parameters. The quantified criteria of the AMPA7 model are shown in Table S-2 for the $P1$ protocol and Table S-3 for the $P2$ protocol. TR-BDF2 and JVODE BDF required a large number of points to achieve an accurate result. In contrast, the hybrid IMEX solver generated more accurate results with the minimal number of points.

For NMDA7 model, the quantified criteria are in Table S-4 for the $P1$ protocol and Table S-5 for the $P2$ protocol. As for AMPA7 model, TR-BDF2 and JVODE BDF required a large number of points to achieve accurate results. The RKF solver consumed less memory but the hybrid IMEX solver was more accurate, according to MSE and NRMSD columns. With our selected criteria priority, RKF takes the advantage on IMEX solver due to its speed.

NMDA15 Receptor Model

Based on MSE and NRMSD values (Table S-6 and Table S-7, respectively) all the algorithms are accurate enough for this particular type of model. The Rosenbrock and IMEX solvers required the smallest number of points, which means they used the least amount of memory for the simulations for both stimulation protocols and both tolerance sets.

Analyzing the two tables, TR-BDF2 and RKF solvers appear to be the worst solvers for NMDA15 model for both protocols and tolerance sets. In addition, the performance of JVODE BDF solver seems to be worst with the second tolerance parameter Tol_2 . Thus, BDF-scheme solvers perform poorly with a restrictive tolerance set.

GABA_A Receptor Model

Table S-8 and S-9 present the raw data for the criteria for protocols $P1$ and $P2$, respectively. As for AMPA7 model, the more accurate solver is IMEX and the faster is RKF. Despite RKF's rapidity, IMEX is more efficient due to its lower memory usage.

N-type VDCC Model

As shown in Figure S-1.B, all solvers did not provide the same current peak. All solvers have specific step-size adaptation algorithms and criteria, which result in different computing time. The last three solvers (JVODE ADAMS, JVODE BDF and JLSODE) include a step-size adaptation, but also an algorithm to adapt the order of the ODE solver, in order to provide accurate results without decreasing step-size. This variable step-size, together with the order adapt is likely to account for the different results obtained for the peak current. However, since variations in MSE and NRMSD are small (see Table S-10 and Table S-11), we can neglect these differences and consider that all solvers provide similar accurate results.

For the N-type VDCC model, all solvers required approximately the same number of points, except for TR-BDF2, which needed more points with Tol_2 parameters. In addition, all solvers generated almost the same NRMSD, except for RKF (for both stimulation protocols) and TR-BDF2 (for Tol_2 set tolerance). Considering the MSE error criterion, hybrid solvers are the worst solvers for this model. Switching from Tol_1 to Tol_2 , TR-BDF2 solver decreased the error parameters (both MSE and NRMSD), although execution time and memory consumption (i.e. number of points) increased and exceeded RK4 values.

Application to glutamatergic synapse Model

The postsynaptic current generated by the simulations is presented in Figure S-2 and Table S-12 presents the quantified results for the glutamatergic synapse model.

As was the case for the N-type VDCC model, the applicative case output currents (postsynaptic current) observed on Figure S-2 are different, depending on the choice of ODE solver (step-size and order variation). However, the small variations in MSE and NRMSD shown in Table S-12 allow us to neglect these differences and to consider that all solvers were accurate and provided the same postsynaptic current.

Hybrid solvers generated the largest MSE and NRMSD, which appears to be their weak features for solving biological kinetic models. But, as indicated in the main article, IMEX is the best solver for the NMDA receptor model (slow model) and Rosenbrock for RHENOMS, which models a glutamatergic synapse.

From Table S-12, MSE and NRMSD values indicated that all solvers are accurate; however, they also indicated that choosing the right solver could save execution time and memory consumption. Without evaluating solver performance, we might have selected a solver that would require a long simulation time. In fact, for a 2-second simulation of the glutamatergic synapse model, the execution time varies approximately between 150 and 900 seconds on our workstation depending on the selected algorithm. This difference would obviously become more important with longer simulations, underlying the importance of the choice of the numerical solver and its underlying parameters to obtain accurate results in minimal time.

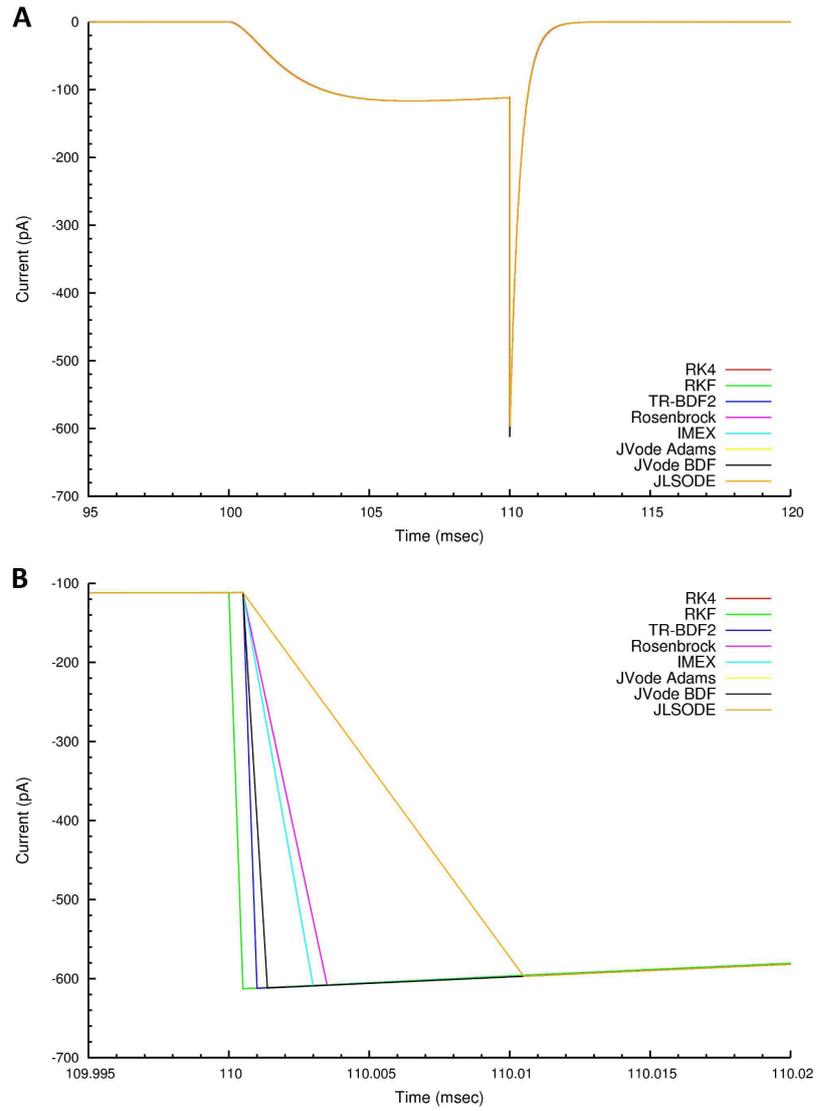


Figure S-1: A. VDCC current resulting from a single event protocol *P1* generated by all ODE solvers. B. Zoom on the rise of the peak current at the end of the plateau current, illustrating the variability between ODE solvers.

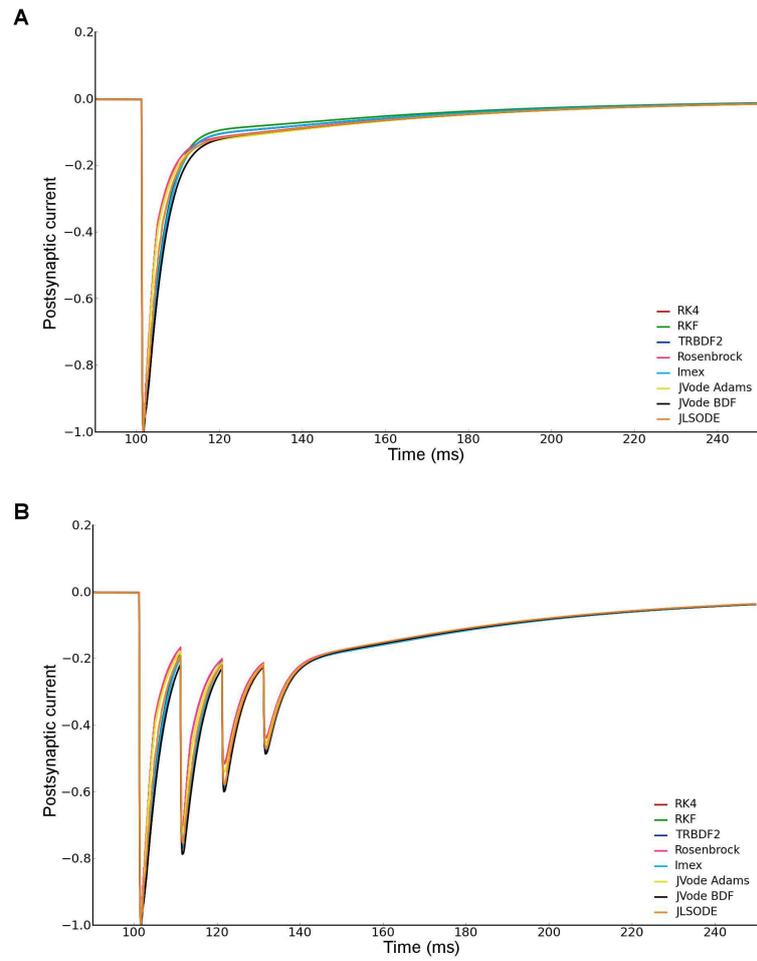


Figure S-2: Normalized postsynaptic current generated by all solver algorithms. A. Single event $P1$ protocol. B. Repeated events $P2$ protocol.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	164.2	1 000 000	–	–
RKF	2.5	1 340	1.67e ⁻⁶	0.23
TR-BDF2	5.4	1 462	2.17e ⁻⁷	0.08
Rosenbrock	2.4	527	2.98e ⁻⁶	0.29
IMEX	4.3	522	4.53e ⁻⁹	0.01
JVODE ADAMS	2.6	559	1.5e ⁻⁶	0.21
JVODE BDF	3.0	1 022	1.26e ⁻⁷	0.06
JLSODE	3.1	549	2.2e ⁻⁸	0.02
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	172.6	1 000 000	–	–
RKF	2.7	1 359	2.72e ⁻⁷	0.09
TR-BDF2	10.1	4 311	2.26e ⁻⁷	0.08
Rosenbrock	2.9	644	5.86e ⁻⁷	0.13
IMEX	5.4	598	7.02e ⁻⁹	0.01
JVODE ADAMS	3.6	1 013	6.07e ⁻⁸	0.04
JVODE BDF	16.2	19 792	1.52e ⁻⁸	0.02
JLSODE	5.4	864	2.03e ⁻⁸	0.02

Table S-2: Detailed quantitative results for the single event protocol *P1* with the two sets of tolerance for the AMPA7 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	167.6	1 000 000	–	–
RKF	3.2	2 391	2.23e ⁻⁶	0.25
TR-BDF2	16.2	7 090	7.49e ⁻⁸	0.05
Rosenbrock	3.4	616	5.96e ⁻⁶	0.41
IMEX	5.4	587	5.37e ⁻⁹	0.01
JVODE ADAMS	3.6	785	3.59e ⁻⁶	0.32
JVODE BDF	4.7	2 206	1.04e ⁻⁷	0.05
JLSODE	5.5	719	9.22e ⁻⁷	0.16
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	173.2	1 000 000	–	–
RKF	3.3	2 412	1.68e ⁻⁶	0.22
TR-BDF2	952.2	16 571	6.14e ⁻⁸	0.04
Rosenbrock	3.6	1 126	1.12e ⁻⁶	0.18
IMEX	6.3	786	1.31e ⁻⁵	0.61
JVODE ADAMS	5.2	2 619	1.85e ⁻⁷	0.07
JVODE BDF	16.9	16 953	1.26e ⁻⁸	0.02
JLSODE	5.8	1 794	1.54e ⁻⁷	0.07

Table S-3: Detailed quantitative results for the single event protocol *P2* with the two sets of tolerance for the AMPA7 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	175.6	1 000 000	–	–
RKF	2.8	514	8.96e ⁻⁶	0.84
TR-BDF2	5.5	1 832	5.43e ⁻⁹	0.02
Rosenbrock	3.6	524	6.56e ⁻⁶	0.72
IMEX	5.0	521	6.26e ⁻¹⁰	0.007
JVODE ADAMS	6.4	541	2.84e ⁻⁷	0.15
JVODE BDF	7.3	836	1.06e ⁻⁸	0.03
JLSODE	10.9	525	3.65e ⁻⁶	0.54
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	180.3	1 000 000	–	–
RKF	3.6	536	2.99e ⁻⁶	0.48
TR-BDF2	10.1	3 932	2.99e ⁻¹⁰	0.003
Rosenbrock	4.2	584	2.73e ⁻⁷	0.14
IMEX	5.4	563	1.52e ⁻¹¹	0.001
JVODE ADAMS	7.1	814	1.07e ⁻⁸	0.03
JVODE BDF	10.6	9 548	5.82e ⁻¹¹	0.002
JLSODE	12.2	725	1.43e ⁻⁷	0.11

Table S-4: Detailed quantitative results for the single event protocol *P1* with the two sets of tolerance for the NMDA7 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	179.8	1 000 000	–	–
RKF	4.1	529	6.5e ⁻⁵	0.92
TR-BDF2	9.1	4 551	1.68e ⁻⁹	0.01
Rosenbrock	3.9	564	6.52e ⁻⁵	0.93
IMEX	5.6	557	1.76e ⁻¹¹	0.001
JVODE ADAMS	4.3	608	9.11e ⁻⁶	0.72
JVODE BDF	10.1	1 234	1.0e ⁻⁶	0.24
JLSODE	11.2	628	3.19e ⁻⁵	0.93
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	182.5	1 000 000	–	–
RKF	4.9	576	8.31e ⁻⁶	0.69
TR-BDF2	28.7	9 066	5.75e ⁻¹¹	0.002
Rosenbrock	5.6	710	4.31e ⁻⁵	0.95
IMEX	9.4	658	3.09e ⁻¹²	4.2e ⁻⁴
JVODE ADAMS	10.1	1 131	6.29e ⁻⁶	0.6
JVODE BDF	28.9	16 834	8.05e ⁻¹⁰	0.006
JLSODE	12.3	1 037	1.03e ⁻⁶	0.24

Table S-5: Detailed quantitative results for the single event protocol *P2* with the two sets of tolerance for the NMDA7 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	270.2	1 000 000	–	–
RKF	19.8	8 025	6.703e ⁻⁸	0.39
TR-BDF2	40.5	3 587	2.696e ⁻⁸	0.24
Rosenbrock	9.1	525	4.538e ⁻⁷	0.1
IMEX	29.1	522	6.393e ⁻⁷	0.12
JVODE ADAMS	10.1	545	5.146e ⁻⁷	0.1
JVODE BDF	10.3	883	7.816e ⁻⁸	0.42
JLSODE	29.1	559	1.669e ⁻⁷	0.61
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	271.2	1 000 000	–	–
RKF	29.5	8 051	4.284e ⁻⁸	0.31
TR-BDF2	13 792	1 847 827	2.419e ⁻¹²	2.35e ⁻⁵
Rosenbrock	14.4	604	2.787e ⁻⁷	0.8
IMEX	39.1	576	3.908e ⁻⁷	0.94
JVODE ADAMS	14.3	896	1.312e ⁻⁷	0.54
JVODE BDF	53.5	11 637	2.641e ⁻¹⁰	0.024
JLSODE	38.8	879	1.479e ⁻⁷	0.58

Table S-6: Detailed quantitative results for the single event protocol *P1* with the two sets of tolerance for the NMDA15 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	356.2	1 000 000	–	–
RKF	18.4	8 034	9.614e ⁻⁸	0.27
TR-BDF2	83.7	8 004	1.07e ⁻⁷	0.29
Rosenbrock	11.7	574	1.23e ⁻⁶	0.98
IMEX	26.6	563	9.823e ⁻⁷	0.87
JVODE ADAMS	14.2	676	8.03e ⁻⁷	0.79
JVODE BDF	16.4	1 510	9.03e ⁻⁸	0.26
JLSODE	30.8	660	1.26e ⁻⁶	0.99
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	356.5	1 000 000	–	–
RKF	22.9	8 067	1.15e ⁻⁷	0.3
TR-BDF2	19 939	2 782 546	9.47e ⁻¹²	2.72e ⁻⁵
Rosenbrock	20.4	804	1.29e ⁻⁶	1.01
IMEX	31.7	718	5.36e ⁻⁷	0.64
JVODE ADAMS	23.1	1 544	1.31e ⁻⁷	0.32
JVODE BDF	80.6	21 805	3.69e ⁻¹⁰	0.016
JLSODE	43.9	1 352	6.77e ⁻⁷	0.72

Table S-7: Detailed quantitative results for the single event protocol *P2* with the two sets of tolerance for the NMDA15 receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = $1e^{-3}$ and <i>Atol</i> = $1e^{-6}$)				
RK4	432.7	1 000 000	–	–
RKF	4.7	1 906	$1.31e^{-7}$	0.16
TR-BDF2	9.1	5 664	$9.96e^{-7}$	0.43
Rosenbrock	5.2	532	$3.16e^{-7}$	0.25
IMEX	8.7	527	$2.12e^{-9}$	0.02
JVODE ADAMS	5.5	574	$1.72e^{-8}$	0.06
JVODE BDF	6.6	862	$8.23e^{-8}$	0.12
JLSODE	5.8	488	$8.04e^{-6}$	0.56
<i>Tol</i> ₂ parameters (<i>Rtol</i> = $1e^{-6}$ and <i>Atol</i> = $1e^{-9}$)				
RK4	434.4	1 000 000	–	–
RKF	6.2	1 948	$3.64e^{-8}$	0.08
TR-BDF2	29.2	21 101	$3.41e^{-7}$	0.26
Rosenbrock	6.4	649	$5.01e^{-8}$	0.09
IMEX	7.2	606	$1.11e^{-8}$	0.04
JVODE ADAMS	9.9	1 052	$4.52e^{-8}$	0.09
JVODE BDF	20.5	14 778	$2.38e^{-8}$	0.06
JLSODE	7.1	888	$2.3e^{-7}$	0.21

Table S-8: Detailed quantitative results for the single event protocol *P1* with the two sets of tolerance for the GABA_A receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = $1e^{-3}$ and <i>Atol</i> = $1e^{-6}$)				
RK4	432.6	1 000 000	–	–
RKF	5.2	1 950	$3.27e^{-8}$	0.25
TR-BDF2	18.4	6 129	$5.09e^{-7}$	0.31
Rosenbrock	5.6	607	$2.96e^{-6}$	0.75
IMEX	14.5	585	$2.47e^{-9}$	0.02
JVODE ADAMS	6.0	748	$1.92e^{-7}$	0.19
JVODE BDF	7.5	1 950	$9.77e^{-8}$	0.13
JLSODE	7.9	753	$1.12e^{-5}$	0.83
<i>Tol</i> ₂ parameters (<i>Rtol</i> = $1e^{-6}$ and <i>Atol</i> = $1e^{-9}$)				
RK4	435.2	1 000 000	–	–
RKF	5.3	2 094	$1.36e^{-7}$	0.16
TR-BDF2	112.7	18 656	$1.12e^{-7}$	0.14
Rosenbrock	6.6	1 037	$3.2e^{-6}$	0.78
IMEX	14.7	880	$9.81e^{-9}$	0.04
JVODE ADAMS	7.9	2 461	$1.41e^{-6}$	0.52
JVODE BDF	40.8	30 728	$1.48e^{-8}$	0.05
JLSODE	9.7	1 703	$1.23e^{-6}$	0.49

Table S-9: Detailed quantitative results for the single event protocol *P2* with the two sets of tolerance for the GABA_A receptor model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	251.4	1 000 000	–	–
RKF	54.5	50 004	3.17e ⁻³	0.01
TR-BDF2	66.7	50 541	4.87e ⁻³	0.37
Rosenbrock	52.7	50 006	5.021e ⁻²	0.36
IMEX	181.1	50 007	5.02e ⁻²	0.36
JVODE ADAMS	85.6	50 008	5.021e ⁻³	0.36
JVODE BDF	49.5	50 008	5.021e ⁻³	0.36
JLSODE	66.9	50 003	5.021e ⁻³	0.36
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	250.2	1 000 000	–	–
RKF	20.8	50 004	3.41e ⁻³	9.53e ⁻⁵
TR-BDF2	612.5	683 015	6.75e ⁻⁵	8.38e ⁻⁵
Rosenbrock	33.8	50 007	5.021e ⁻²	0.36
IMEX	151.6	50 007	5.02e ⁻²	0.36
JVODE ADAMS	44.3	50 007	5.021e ⁻³	0.36
JVODE BDF	45.1	50 007	5.021e ⁻³	0.36
JLSODE	57.2	50 003	5.021e ⁻³	0.36

Table S-10: Detailed quantitative results for the single event protocol *P1* with the two sets of tolerance for the N-type VDCC model.

Algorithm	Simulation time (ms)	Number of points	MSE	NRMSD (%)
<i>Tol</i> ₁ parameters (<i>Rtol</i> = 1e ⁻³ and <i>Atol</i> = 1e ⁻⁶)				
RK4	249.8	1 000 000	–	–
RKF	24.2	50 010	8.15e ⁻³	0.01
TR-BDF2	65.4	56 475	6.14e ⁻⁴	0.12
Rosenbrock	37.3	50 018	1.28e ⁻²	0.57
IMEX	159.8	50 019	1.27e ⁻²	0.57
JVODE ADAMS	46.4	50 037	1.27e ⁻³	0.57
JVODE BDF	47.5	50 037	1.27e ⁻³	0.57
JLSODE	58.6	50 009	1.28 ⁻³	0.57
<i>Tol</i> ₂ parameters (<i>Rtol</i> = 1e ⁻⁶ and <i>Atol</i> = 1e ⁻⁹)				
RK4	250.6	1 000 000	–	–
RKF	24.8	50 010	6.7e ⁻³	0.01
TR-BDF2	3 370.4	3 215 250	2.5e ⁻⁵	0.01
Rosenbrock	63.4	50 018	1.28e ⁻²	0.57
IMEX	190.6	50 019	1.27e ⁻²	0.57
JVODE ADAMS	92.4	50 037	1.27e ⁻³	0.57
JVODE BDF	45.5	50 037	1.27e ⁻³	0.57
JLSODE	66.5	50 009	1.28e ⁻³	0.57

Table S-11: Detailed quantitative results for the single event protocol *P2* with the two sets of tolerance for the N-type VDCC model.

Algorithm	Simulation time (ms)	Used memory (Mbytes)	MSE	NRMSD (%)
Protocol $P1$ with Tol_1 parameters ($Rtol = 1e^{-3}$ and $Atol = 1e^{-6}$)				
RK4	982	4 302	–	–
RKF	310	3 505	$1.58e^{-3}$	0.087
TR-BDF2	490	2 538	$1.14e^{-4}$	0.023
Rosenbrock	131	1 250	$5.78e^{-2}$	0.532
IMEX	425	947	$3.36e^{-4}$	0.055
JVODE ADAMS	140	973	$4.79e^{-3}$	0.048
JVODE BDF	143	1 106	$3.66e^{-3}$	0.042
JLSODE	151	1 127	1^{-3}	0.022
Protocol $P2$ with Tol_1 parameters ($Rtol = 1e^{-3}$ and $Atol = 1e^{-6}$)				
RK4	950	4 750	–	–
RKF	311	3 616	$1.44e^{-3}$	0.084
TR-BDF2	248	2 807	$1.74e^{-4}$	0.029
Rosenbrock	158	992	$4.82e^{-2}$	0.485
IMEX	447	1 265	$5.02e^{-4}$	0.049
JVODE ADAMS	160	1 068	$3.04e^{-3}$	0.048
JVODE BDF	174	1 824	$2.73e^{-3}$	0.046
JLSODE	183	1 586	$5.37e^{-4}$	0.026

Table S-12: Detailed quantitative results for the glutamatergic synapse with single event $P1$ and repeated events $P2$ protocols for Tol_1 parameters.