HAL
open science

Efficient high order schemes for stiff ODEs in cardiac electrophysiology

Charlie Douanla Lontsi, Yves Coudière, Charles Pierre

To cite this version:

Charlie Douanla Lontsi, Yves Coudière, Charles Pierre. Efficient high order schemes for stiff ODEs in cardiac electrophysiology. 2017. hal-01406683v2

HAL Id: hal-01406683

https://inria.hal.science/hal-01406683v2

Preprint submitted on 19 Dec 2017 (v2), last revised 24 Apr 2018 (v3)

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.

Efficient high order schemes for stiff ODEs in cardiac electrophysiology

Charlie Douanla-Lontsi *[1], Yves Coudière †[1,2] and Charles Pierre ‡[3]

[1] INRIA Bordeaux Sud Ouest, Université de Bordeaux, France.
[2] Institut de Mathématiques de Bordeaux, UMR CNRS 5241.
[3] Laboratoire de Mathématiques et de leurs Applications, UMR CNRS 5142, Université de Pau et des Pays de l'Adour, France.

13 October, 2017

Abstract

In this work we analyze the resort to high order exponential solvers for stiff ODEs in the context of cardiac electrophysiology modeling. The exponential Adams-Bashforth and the Rush-Larsen schemes will be considered up to order 4. These methods are explicit multistep schemes.The accuracy and the cost of these methods are numerically analyzed in this paper and benchmarked with several classical explicit and implicit schemes at various orders. This analysis has been led considering data of high particular interest in cardiac electrophysiology : the activation time (t_a), the recovery time (t_r) and the action potential duration (APD). The Beeler Reuter ionic model, especially designed for cardiac ventricular cells, has been used for this study. It is shown that, in spite of the stiffness of the considered model, exponential solvers allow computation at large time steps, as large as for implicit methods. Moreover, in terms of cost for a given accuracy, a significant gain is achieved with exponential solvers. We conclude that accurate computations at large time step are possible with explicit high order methods. This is a quite important feature when considering stiff non linear ODEs.

Keywords: Exponential schemes, stiff ordinary differential equations, high order schemes, cardiac electrophysiology

Acknowledgments. This study received financial support from the french government as part of the "Investissement d'Avenir" program managed by the "Agence Nationale de la Recherche" (ANR), grant reference ANR-10-IAHU-04. It also received fundings of the ANR project HR-CEM n0. 13-MONU-0004-04.

*charlie.douanla-lontsi@inria.fr
†yves.coudiere@inria.fr
‡charles.pierre@univ-pau.fr

1 Introduction

The numerical resolution of stiff ordinary differential equations (ODEs) is an issue encountered in many fields of applied sciences. In cardiac electrophysiology, the electrical activity of the heart is described by a system of parabolic partial differential equations coupled with a system of ODEs called *ionic models*. The stiffness and the nonlinearity of the ionic models (see [10] for the stiffness analysis) make their numerical resolution very challenging. The classical schemes have serious drawbacks to solve such ODEs. On the one hand, the classical stable methods are implicit and lead to high computational cost (because of the nonlinear solvers) for large time-steps, on the other hand explicit solvers require very small time steps also leading to high computational costs. Meanwhile current solvers in cardiac electrophysiology are usually based on order 1 or 2 schemes (see [8, 11, 9, 4]). In this paper we investigate the resort to a class of both explicit and stable schemes referred as «exponential methods » of high order as an alternative to solve cardiac electrophysiological problems. Namely we will consider the exponential Adams-Bashforth (EAB) and the Rush-Larsen (RL) techniques.
Let us consider the general initial value problem,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = F(t,y) \quad t \in (0,T], \quad y(0) = y_0 \in \mathbb{R}^N. \tag{1}$$

EAB schemes [6] and RL schemes [3] take advantage of a splitting of the model function F into some linear part a and a nonlinear part b, such that (1) becomes,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a(t,y)y + b(t,y), \quad y(0) = y_0 \in \mathbb{R}^N. \tag{2}$$

Notice that in (2), a is not the exact linear part of F (its differential) but, an approximation or a guess thereof. The EAB and RL are built from a transformation of (2) on each time discretization interval $[t_n, t_{n+1}]$ in the following form,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \alpha_n y + c_n(t,y), \quad y(0) = y_0 \in \mathbb{R}^N. \tag{3}$$

Where $\alpha_n \in \mathbb{R}^N$ is a stabilizer set at every time step and $c_n(t,y) = (a(t,y) - a_n)y + b(t,y)$. With the formulation (3), the exact solution satisfies the variation of the constant formula,

$$y(t_{n+1}) = \mathrm{e}^{\alpha_n h}\left(y(t_n) + \int_{t_n}^{t_{n+1}} \mathrm{e}^{-\alpha_n(\tau - t_n)}\, c_n(\tau, y(\tau))\mathrm{d}\tau\right). \tag{4}$$

EAB and RL schemes are based on this formula and are obtained by replacing c_n by an approximation. Their precise definitions will be given in Section 2. The aim of this paper is to study the efficiency of EAB and RL methods of order 1 up to 4, and to show for a given scheme of order k, how to compute without degrading the accuracy, the activation time (t_a), the recovery time (t_r) and the action potential duration (APD) which are informative values of a particular interest in cardiac electrophysiology. The efficiency of

the schemes is analyzed both in terms of the accuracy and of the cost. The comparison is made using a realistic test case and is completed by including a benchmark with several classical methods either of implicit or explicit type: The Crank-Nicolson (CN), the Runge Kutta (RK_4), the Adams-Bashforth (AB_k), and the backward differentiation (BDF_k) (see [4, 5] for the statement of these schemes). One interesting result from this test case is the possibility with EAB and RL, to use large time steps as in the case of implicit schemes but with the same cost as for the explicit schemes. The paper is organized as follows. In Section 2 are presented the stabilized schemes. A brief description of the transmembrane action potential, t_a, t_r, APD and ionic model is given in Section 3. The methodology used to compare the numerical schemes and to compute the values t_a, t_r and APD predicted by a numerical solution are developed in Section 4. The comparison of the methods follows in Section 5. In Section 6, a brief conclusion is given.

2 EAB_k and RL_k scheme statements

When the function $c_n(t, y)$ in (3) is a polynomial $\sum_{j=0}^{k-1} p_j(t - t_n)^j$ of degree $k - 1$, the relation (4) becomes,

$$y_{n+1} = \mathrm{e}^{\alpha_n h}\, y_n + h \sum_{j=0}^{k-1} p_j j! h^j \varphi_{j+1}(\alpha_n h),$$

where the functions φ_j are defined recursively by,

$$\varphi_{j+1}(z) = \frac{\varphi_j(z) - \varphi_j(0)}{z}, \quad \varphi_0(z) = \mathrm{e}^z \quad \text{and} \quad \varphi_j(0) = \frac{1}{j!} \quad \forall j \geq 0.$$

The schemes introduced in the sequel are multi-steps. We will use the following notation $a_n = a(t_n, y_n)$, $b_n = b(t_n, y_n)$.

Table 1: Coefficients γ_{nj} for the EAB_k schemes.

k	1	2	3	4
γ_{n0}	c_n^n	c_n^n	c_n^n	c_n^n
γ_{n1}	–	$c_n^n - c_n^{n-1}$	$\frac{3}{2}c_n^n - 2c_n^{n-1} + \frac{1}{2}c_n^{n-2}$	$\frac{11}{6}c_n^n - 3c_n^{n-1} + \frac{3}{2}c_n^{n-2} - \frac{1}{3}c_n^{n-3}$
γ_{n2}	–	–	$c_n^n - 2c_n^{n-1} + c_n^{n-2}$	$2c_n^n - 5c_n^{n-1} + 4c_n^{n-2} - c_n^{n-3}$
γ_{n3}	–	–	–	$c_n^n - 3c_n^{n-1} + 3c_n^{n-2} - c_n^{n-3}$

- EAB_k: On the one hand we set $\alpha_n = a_n$, on the other hand the function c_n in (3) is approximated by its Lagrange interpolation polynomial $\tilde{c}_n$ of degree $k - 1$ at the time instants $t_n, \ldots, t_{n-k+1}$. This polynomial satisfies $\tilde{c}_n(t_{n-j}) = c_n(t_{n-j}, y_{n-j})$ for $j = 0, \ldots, k-1$. The values $c_n(t_{n-j}, y_{n-j})$ are given by $c_n^{n-j} = b_{n-j} + (a_{n-j} - a_n)y_{n-j}$

for $j = 0, \ldots, k-1$. If we write $\tilde{c}_n(t) = \sum_{j=0}^{k-1} \frac{\gamma_{nj}}{j!} \left(\frac{t-t_n}{h}\right)^j$, the definition of the EAB_k scheme is deduced from the formula (4) by

$$y_{n+1} = \mathrm{e}^{a_n h}\, y_n + h \sum_{j=0}^{k-1} \gamma_{nj} \varphi_{j+1}(a_n h), \tag{5}$$

where the coefficients γ_{nj} are given in Table 1.

- RL_k: In the case the function $c_n(t, y)$ in (4) is a constant $c_n = \beta_n \in \mathbb{R}$ then we have the following simple scheme definition,

$$y_{n+1} = y_n + h\varphi_1(\alpha_n h)(\alpha_n y_n + \beta_n), \tag{6}$$

that we refer as Rush-Larsen schemes as in the continuity of [8]. The following choices for defining α_n and β_n ensure the convergence at order k of the scheme (6) and thus are named Rush-Larsen schemes of order k (RL_k).

  - $k = 1: \quad \alpha_n = a_n, \quad \beta_n = b_n.$
  - $k = 2: \quad \alpha_n = \frac{3}{2}a_n - \frac{1}{2}a_{n-1}, \quad \beta_n = \frac{3}{2}b_n - \frac{1}{2}b_{n-1}.$
  - $k = 3: \quad \alpha_n = \frac{1}{12}(23a_n - 16a_{n-1} + 5a_{n-2}),$
    $\beta_n = \frac{1}{12}(23b_n - 16b_{n-1} + 5b_{n-2}) + \frac{h}{12}(a_n b_{n-1} - a_{n-1} b_n).$
  - $k = 4: \quad \alpha_n = \frac{1}{24}(55a_n - 59a_{n-1} + 37a_{n-2} - 9a_{n-3}),$
    $\beta_n = \frac{1}{24}(55b_n - 59b_{n-1} + 37b_{n-2} - 9b_{n-3}) + \frac{h}{12}(a_n(3b_{n-1} - b_{n-2}) - (3a_{n-1} - a_{n-2})b_n).$

Notice that the EAB_1 scheme is the same with RL_1 scheme and also the exponential Euler scheme.

The previous description of the EAB_k scheme has been given very briefly but, more details can be found in [6, 7] (for general ODEs) and in [2] for cardiac electrophysiology application.

3 Modeling in cellular cardiac electrophysiology

3.1 The action potential

The phenomenon studied here is the so called *cellular action potential*, that we briefly present here. A potential difference is observed between the inside and outside of the cell, said membrane potential and denoted V. This potential caused by the differences in ionic concentrations between the inside and outside of the cells is dynamic in time, as well as these ionic concentrations. The potential V can abruptly switch from a *resting* state (during which $V = V_r \simeq -100mV$) to an *excited* state (where V is in the range

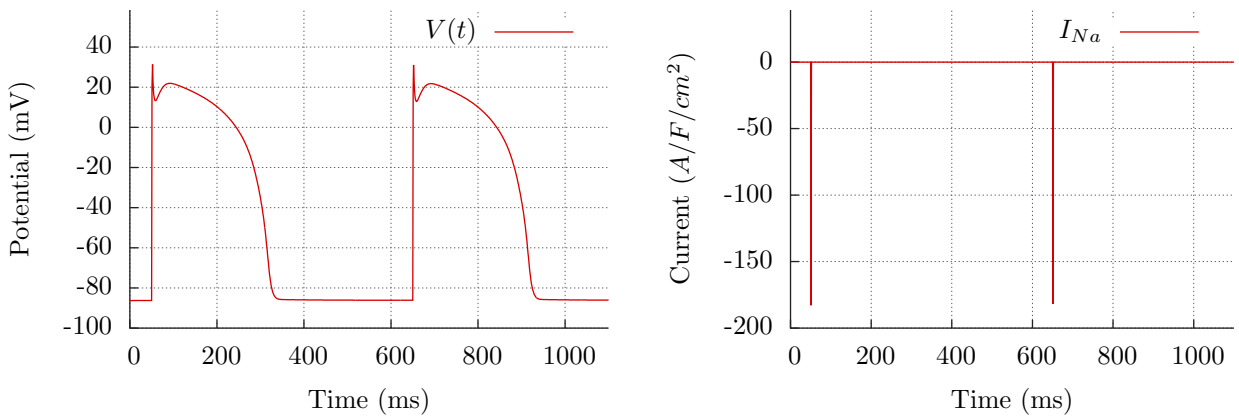

Figure 1: TNNP model [12] illustration. Left, two cellular action potentials : starting at a negative resting value, the transmembrane voltage $V(t)$ has a stiff depolarization followed by a plateau and repolarizing to the resting value. Right : each depolarization is induced by an ionic sodium current $I_{Na}(t)$

of 10 mV) in which it is maintained during a few tenth of seconds before returning to its resting state (see Figure 1). It is this cycle,

$$\text{resting state} \xrightarrow{\text{excitation}} \text{excited state} \xrightarrow{\text{recovery}} \text{resting state},$$

that one designates as action potential. The resting potential V_r is associated to a pic potential V_p corresponding to the maximum of the potential V at the end of the excitation and a threshold potential V_{th} such that $V_r < V_{th} < V_p$.

We adopt here the following definitions : V_{th} is the potential corresponding to 20% of depolarization, the activation time (t_a) and the recovery time (t_r) are the time instants where the potential reaches the value V_{th} the first and the second time respectively and the action potential duration (APD) is the amount of time in which the voltage remains elevated above V_{th}. More precisely,

$$V_{th} = 0.8V_r + 0.2V_p, \quad V(t_a) = V_{th} = V(t_r),\ t_a < t_r \quad \text{and} \quad \text{APD} = t_r - t_a. \tag{7}$$

3.2 Ionic Models

The variations of the ionic concentrations are described by ionic models and are systems of ODE. Experimental models (such as BR [1] and TNNP [12] models designed for cardiac human cells) involve a variable $y \in \mathbb{R}^N$ composed of the following entries:

- **The membrane potential:** V in mV. The equation on the potential is written,

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -I_{ion}(y(t)) + I_{st}(t), \tag{8}$$

  where I_{ion} (reaction term) is the total ionic current crossing the membrane cell and I_{st} is the stimulation current, it is a source term.

- **The gating variables:** they are parameters between 0 and 1 expressing the variability and the permeability of the membrane cell for the specific ionic species. One denote by $W \in \mathbb{R}^P$ the vector of gating variables. The equations on W are, for $i = 1 \dots P$,
$$\frac{\mathrm{d}W_i}{\mathrm{d}t} = \frac{W_{\infty,i}(y) - W_i}{\tau_i(y)}, \tag{9}$$
where $W_{\infty,i}(y) \in \mathbb{R}$, $\tau_i(y) \in \mathbb{R}$ are scalar functions given by the model. In these equations the linear and nonlinear parts are encoded in the model and are equal to $-1/\tau_i(y)$ and $W_{\infty,i}(y)/\tau_i(y)$ respectively.

- **Ionic concentrations:** One denote by $C \in \mathbb{R}^{N-P-1}$ the vector of concentrations.

All these entries are collected in the vector y as follows ,

$$y = \begin{bmatrix} W \\ X \end{bmatrix}, \quad X = \begin{bmatrix} C \\ V \end{bmatrix}, \quad W \in \mathbb{R}^P, \quad C \in \mathbb{R}^{N-P-1}, \quad V = y_N \in \mathbb{R},$$

The sub-vectors W corresponds to the lines of (1) including stabilization with the linear part $a(t, y) = -1/\tau(y)$ and the non linear part $b(t, y) = -W_\infty(y)/\tau(y)$. The sub-vector X corresponds to the lines of (1) with no stabilization ($a(t, y) = 0$). The associated ODE written in the form (2) is then defined by,

$$a(t, y) = \begin{bmatrix} A_1(t, y) & 0 \\ 0 & 0 \end{bmatrix}, \quad b(t, y) = \begin{bmatrix} B_1(t, y) \\ B_2(t, y) \end{bmatrix},$$

where the matrix $A_1(t, y) \in \mathbb{R}^P \times \mathbb{R}^P$ is diagonal, $A_1(t, y) = \mathrm{Diag}(-1/\tau_i(y))$, and $B_1(t, y) = \{W_{\infty,i}(y)/\tau_i(y), i = 1 \dots P\} \in \mathbb{R}^P$.

4 Numerical Study

4.1 Scheme analysis methods

- **Test case :** The evaluation and comparisons between different ODE solvers is done with a test case. Specifically, the Beeler Reuter [1] model is considered and written in the form (2) as described in Section 3.2. We denote by $y(t)$ the solution of the associated ODE (2) in $(0, T]$ with $T = 396\ ms$. this solution is uniquely defined once the initial condition y_0 and the stimulation current I_{st} in (8) are fixed. y_0 is the resting state as described by the model. The function $I_{st}(t)$ is positive, null outside the interval $(t_s - 1, t_s + 1)$, t_s=20 ms and with integral $\int_0^T I_{st}(t)dt = I_{stim}$, a typical current of stimulation fixed by the models, equal to 50 mA. We also impose to I_{st} a C^4 regularity in order to observe the convergence orders of schemes up to 4.

- **Numerical solution:** Let $m \geq 1$ be an integer for which one associates the time-step $h = T/m$ and the regular mesh $T_m = \{t_n = jh,\ j = 0 \dots m\}$ of

the interval $(0, T]$. The numerical solution (y^n) is the element of the space E_m, $E_m = \{(y^n)_{0\leq n\leq m},\ y^n \in \mathbb{R}^N\}$. The space E_m of the numerical solutions is simply $(\mathbb{R}^N)^m$ but to $(y^n) \in E_m$ is implicitly associated a time-step h and a mesh T_m, such that each value y^n, $0 \leq n \leq m$ of $(y^n) \in E_m$ is supposed to be an approximation of $y(t_n)$.

- **Reference solution:** For a given test case, we cannot access to the exact solution $y(t)$ of the associated ODE. So for a numerical solution $(y^n) \in E_m$, we set $m' = 2^r m$ with $r \geq 0$ an integer and define the reference solution associated to (y^n) (or m) as the numerical solution $y_{ref} \in E_{m'}$ for the problem (1), computed by the RK_4 scheme with the time-step $h_{ref} = T/m' = h/2^r$. The reference solution y_{ref} is then not unique and depend on r. In practice r is chosen *large enough* such that the error between the exact solution y and y_{ref} is negligible compared to the error between the numerical solution (y^n) and y_{ref}.

- **Interpolation of the solution:** To compare the numerical solution with the reference solution and to be able to compute the numerical error in terms of function norm, we define an interpolator $\pi_{m,i} : E_m \longrightarrow C^0(0, T]$, transforming the component i of the numerical solution $(y^n) \in E_m$ in $C^0(0, T]$, the set of the continuous functions on $(0, T]$. We impose to the interpolant $\pi_{m,i}y^n$ to be a polynomial piecewise function of degree 3, this constraint is necessary to observe the convergence order up to 4. We assume that m is a multiple of 3 and fix $(y^n) \in E_m$. We decompose the interval $[0, T]$ in a sequence of 3 intervals packages $P_s = [t_{3s}, t_{3s+1}] \cup [t_{3s+1}, t_{3s+2}] \cup [t_{3s+2}, t_{3(s+1)}]$, for $s = 0 \ldots m/3$. The interpolated $f := \pi_{m,i}y^n$ is the unique polynomial of degree 3 on each P_s, continuous on $[0, T]$, such that $f(t_n) = y_i^n$ for all $n = 0 \ldots m$. This interpolator is not Canonical: an H^3-Hermite interpolation on each interval (t_n, t_{n+1}) is an alternative. The emphasis will be on the membrane potential $V(t) = y_N(t)$ and for more simplicity we note $\pi_m = \pi_{m,N}$ and $\pi = \pi_{m,N}$ in confusion absence.

- **Accuracy:** Let (y^n) be a numerical solution and y_{ref} a reference solution. We denote $\widehat{V}_{ref}$ and $\pi y^n = \widehat{V}$ the reference membrane potential and the membrane potential interpolating associated to (y^n). The accuracy of each method is evaluated through a relative error between the reference solution and the numerical solution. We define the errors in norm L^∞ by :

$$e_\infty = \frac{\max |\widehat{V} - \widehat{V}_{ref}|}{\max |\widehat{V}_{ref}|}. \tag{10}$$

  Notice that the choice of the membrane potential V is arbitrary and that any other component of (y^n) could have been considered. The accuracy notion will be central here and it is convenient to identify several aspects.

- **Cost:** The accuracy takes all its meaning when it is associated with a cost. Here it is a *computational* cost and is evaluated with the CPU time during a simulation. It is evaluated by our fortran 90 code for each simulation. The CPU times depend

on the computer used to perform the solutions and how the numerical solver is implemented. This is especially the case of implicit solvers where the Newton-Krylov algorithm type is used with the approximation of the Jacobian by the finite difference method.

4.2 Computation of t_a, t_r, APD and the associated errors

Let (y^n) be a numerical solution given and $V^n = y_N^n$, $0 \leq n \leq m$, the membrane potential associated to (y^n). Then there exist two unique indexes $n_a < n_r$ such that

$$V_{n_a+1} \leq V_{th} < V_{n_a+2} \quad , \qquad V_{n_r+1} \leq V_{th} < V_{n_r+2},$$

for the threshold potential V_{th} defined in Section 3.1.

On the intervals (t_{n_i}, t_{n_i+3}) , $i \in \{a, r\}$, we compute the Lagrangian interpolation polynomial of degree 3 $p_i(t)$ for the values V_j associated to t_j, $j = n_i, \ldots, n_i + 3$. The activation time t_a an the recovery time t_r are the computed as the solution of,

$$p_a(t_a) = V_{th}, \quad p_r(t_r) = V_{th}.$$

Again the use of interpolation of order 3 is necessary to observe the convergence order up to 4. In above, we suppose that all is well defined, which is the case if the numerical solution (y^n) is physiologically relevant.

The relative error between the activation, recovery time and APD predicted by a numerical solution (y^n) and a reference solution y_{ref} will be computed by,

$$e_{t_a} = \frac{|t_a - t_{a,ref}|}{|t_{a,ref}|}, \quad e_{t_r} = \frac{|t_r - t_{r,ref}|}{|t_{r,ref}|}, \quad e_{\mathrm{APD}} = \frac{|\mathrm{APD} - \mathrm{APD}_{ref}|}{|\mathrm{APD}_{ref}|}.$$

5 Numerical results

5.1 Accuracy

The relative error $e(h)$ is computed for various time-steps h and depicted in Table 2 where it can be observed that all the methods exhibit the expected order of convergence. A general view of Table 2 shows that the RL_k is always more accurate than EAB_k and unlike the classical explicit schemes, the stabilized schemes allows the use of large time-steps as the implicit except at the order four where it is not possible for $h = 0.2$.

Table 2(a) shows that the CN is the most accurate among the methods of order 2 with a factor in the range of 10. Table 2(b) shows that the BDF_3 method is better than the stabilized schemes for $h \geq 0.0125$ with a coefficient 10 for $h = 0.2$ while for $h < 6.25 \times 10^{-3}$ the RL_3 is more accurate. Table 2(c) shows that the RK_4 method is the most accurate among the methods of order 4 for $h \leq 0.025$ while for $h > 0.025$ the BDF_4 is more accurate than the stabilized schemes.

Table 2: Accuracy for the BR model for various classical and stabilized methods.

(a) AB_2, RL_2, EAB_2 and CN

h	AB_2	RL_2	EAB_2	CN
0.2	–	0.251	0.284	4.11×10^{-2}
0.1	–	0.107	9.26×10^{-2}	1.13×10^{-2}
0.05	–	3.35×10^{-2}	2.31×10^{-2}	2.65×10^{-3}
0.025	–	8.88×10^{-3}	5.39×10^{-3}	6.66×10^{-3}
0.0125	–	2.23×10^{-3}	1.29×10^{-3}	1.68×10^{-4}
6.25×10^{-3}	2.07×10^{-4}	5.6×10^{-4}	3.17×10^{-4}	4.25×10^{-5}

(b) AB_3, RL_3, EAB_3 and BDF_3

h	AB_3	RL_3	EAB_3	BDF_3
0.2	–	0.148	0.516	4.09×10^{-2}
0.1	–	4.07×10^{-2}	9.17×10^{-2}	1.04×10^{-2}
0.05	–	6.34×10^{-3}	1.09×10^{-2}	2.29×10^{-3}
0.025	–	7.57×10^{-4}	1.17×10^{-3}	3.84×10^{-4}
0.0125	–	9.07×10^{-5}	1.4×10^{-4}	5.25×10^{-5}
6.25×10^{-3}	1.13×10^{-5}	8.23×10^{-6}	1.72×10^{-5}	2.01×10^{-5}

(c) RK_4, RL_4, EAB_4 and BDF_4

h	RK_4	RL_4	EAB_4	BDF_4
0.2	–	–	–	4.98×10^{-2}
0.1	–	5.86×10^{-2}	0.119	1.27×10^{-2}
0.05	–	4.58×10^{-3}	8.96×10^{-3}	2.02×10^{-3}
0.025	4.65×10^{-5}	2.61×10^{-4}	4.33×10^{-4}	1.93×10^{-4}
0.0125	2.67×10^{-6}	1.62×10^{-5}	2.67×10^{-5}	3.52×10^{-5}
6.25×10^{-3}	1.65×10^{-7}	9.94×10^{-7}	1.73×10^{-6}	2.01×10^{-5}

5.2 Cost

A general observation of Figure 2 on the top shows that for the error between 1% and 10% the gain in terms of CPU time is high (with a factor in the range of 10) when moving from the order 1 to order 2 schemes. This gain remains important (with a factor in the range of 5) when moving from the order 2 to order 3 schemes while for the errors between 1% and 10% there is no gain when moving from the order 3 to the order 4 schemes. However the order 4 becomes advantageous for the errors less than 0.1%.

Figure 2 on the bottom shows that the RL_3 and the RL_4 are less costly than the EAB_3 and EAB_4 respectively. The factor is not so high but in terms of implementation, the RL is easier than the EAB schemes.

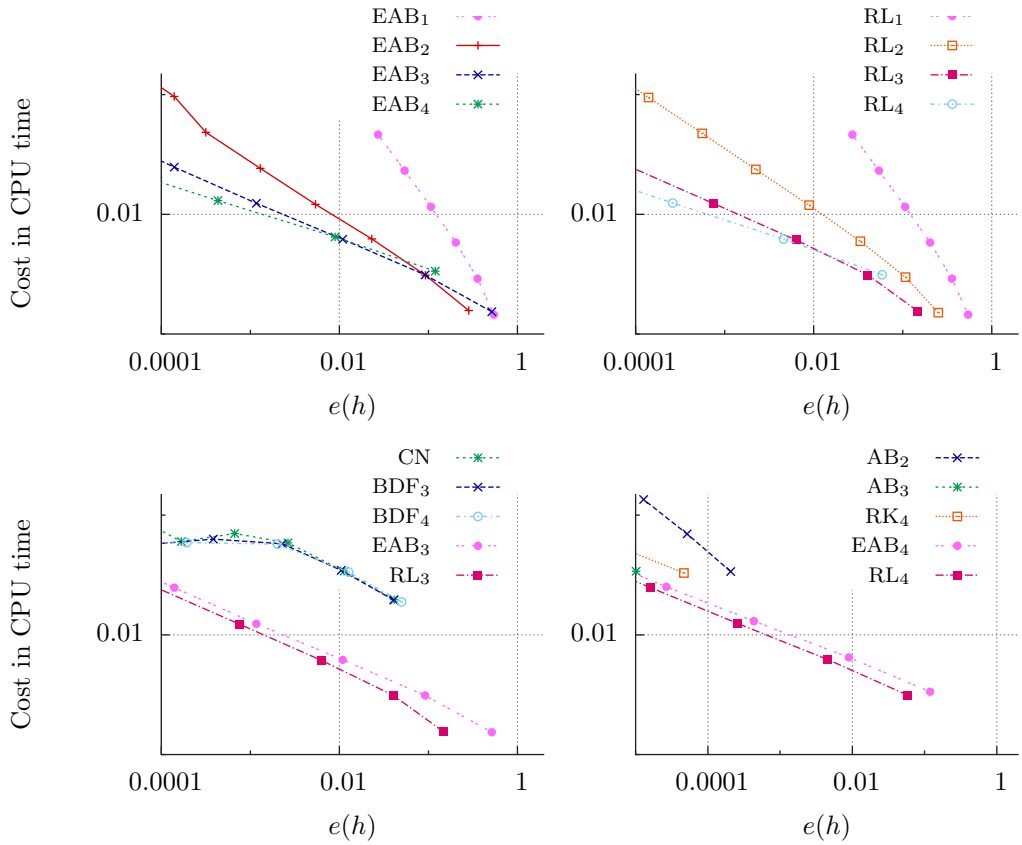

Figure 2: CPU time as a function of the error $e(h)$ in Log/Log scale.

Figure 2 on the bottom left shows that when using high order stabilized schemes instead of implicit schemes, the gain in time CPU is very high with a coefficient greater than 10. This is due to the fact that the nonlinear solver is very expensive and its cost become very high for large time-steps.

Figure 2 on the bottom right shows that the order 4 stabilized schemes are less costly than the classical explicit schemes but it is better to use the RL_4 scheme instead of the EAB_4 scheme. Because of their stability properties the classical explicit schemes require the use of small time-steps that make them sometimes useless. For instance the RK_4 is

very accurate but its use require to take a small time step. This small time steps produces a very small error that might be not needed and then its use will induce an additional cost.

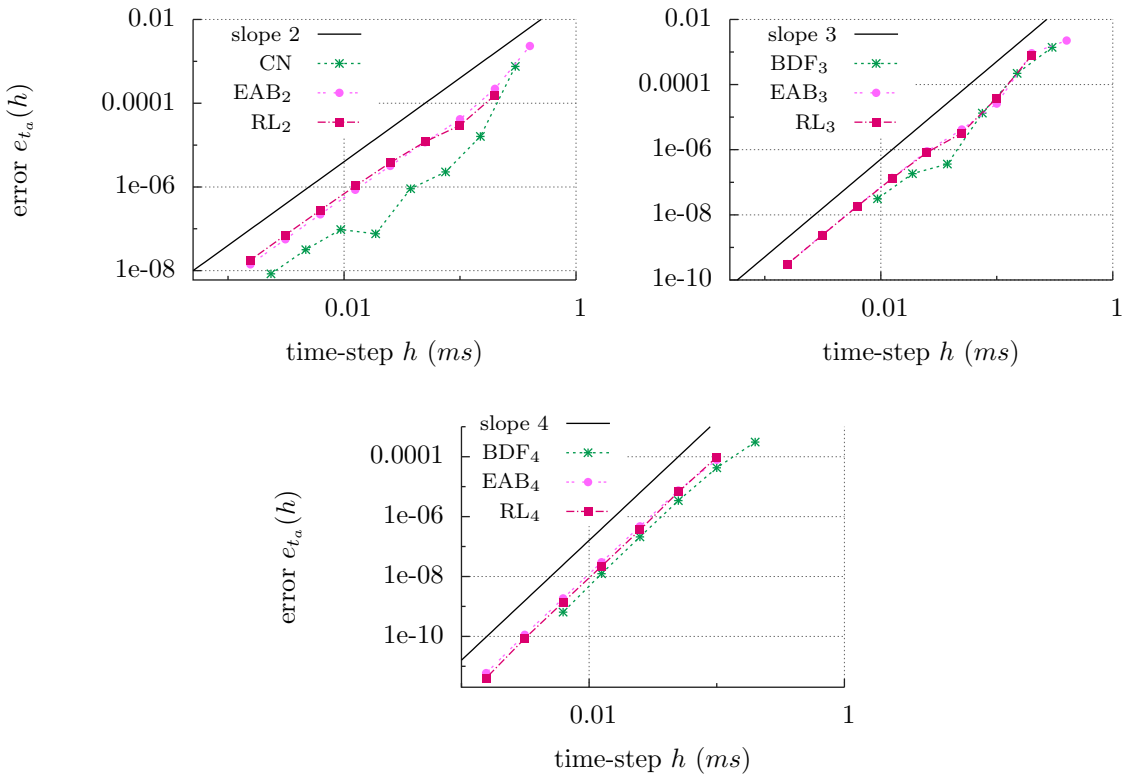

Figure 3: Relative error $e_{t_a}(h)$ for the CN, BDF$_3$, BDF$_4$, EAB$_k$ and RL$_k$ schemes $k = 2, 3, 4$ for the BR model.

5.3 Accuracy on t_a, t_r and APD

We investigate in this section the accuracy on t_a, t_r and APD. The previous results Section 5 showed that classical explicit methods induce very high computational costs because of their lack of stability. Only the implicit methods will then be considered to benchmark with the EAB$_k$ and RL$_k$ methods. For a given numerical solution, the method described in Section 4.2 is used to perform the values of t_a, t_r and APD . These values are then compared with the ones predicted by the reference solution. The errors for t_a and t_r are depicted on the figures 3 and 4 for various time-steps. These figures show that for a numerical solution computed with an order k numerical scheme, the values of t_a and t_r predicted converge to the ones predicted by the reference solution with the same convergence order. In the same figures, we can see that with equal time-step $h \leq 0.01$, the errors on the values predicted by the numerical solution decrease with a factor of 10

at least, when moving from the order k to the order $k+1$ schemes. We didn't show the pictures on APD but since $APD = t_r - t_a$, the results on APD will be the same as for t_a and t_r.

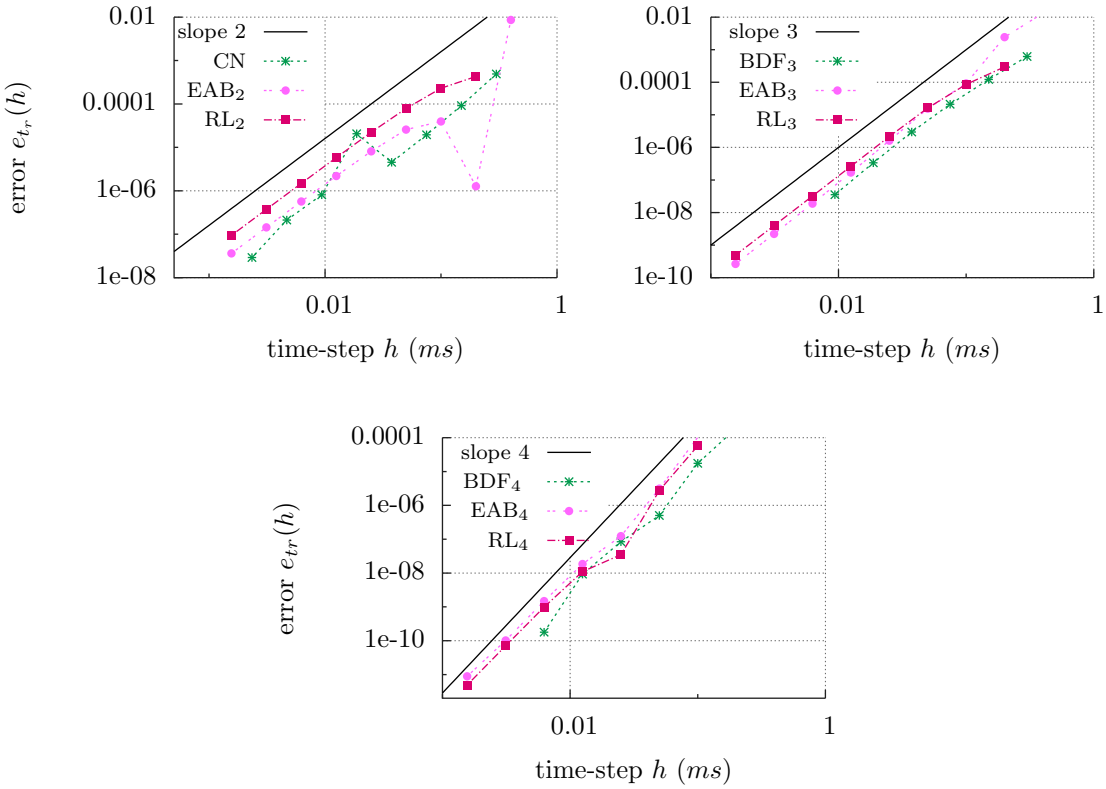

Figure 4: Relative error $e_{t_r}(h)$ for the CN, BDF$_3$, BDF$_4$, EAB$_k$ and RL$_k$ schemes $k = 2, 3, 4$ for the BR model.

6 Conclusion

Two families of explicit high order stabilized methods (EAB$_k$, RL$_k$) have been studied in this work. Excepted the order four, both have been shown to be as stable as the classical implicit methods for the test case we have chosen. Meanwhile the two families of schemes have been compared with some classical solvers (CN, BDF2, BDF3, AB$_2$, AB$_3$, RK$_4$). This comparison has shown (for the test case we chose)that EAB and RL are competitive when both the cost and the accuracy are taken in account. Otherwise, it has also been demonstrated that the use of high order (3 or 4) of the stabilized methods instead of the classical high order implicit methods allows to decrease the cost almost 50 times. A method permitting to compute accurately (without degrading the convergence order of the numerical scheme) the values of t_a, t_r and APD predicted by a numerical solution has been also described and numerically investigated in this work.

References

[1] G. Beeler and H. Reuter. Reconstruction of the action potential of ventricular myocardial fibres. *J. Physiol.*, 268(1):177–210, 1977.

[2] Y. Coudiére, C. Douanla-Lontsi, and C. Pierre. Exponential Adams Bashforth integrators for stiff ODEs, application to cardiac electrophysiology. *HAL Preprint no. hal-01394036*, 2017.

[3] Y. Coudiére, C. Douanla-Lontsi, and C. Pierre. Rush-Larsen time-stepping methods of high order for stiff problems in cardiac electrophysiology. *HAL Preprint no. hal-01557856*, 2017.

[4] M. Ethier and Y. Bourgault. Semi-implicit time-discretization schemes for the bidomain model. *SIAM Journal on Numerical Analysis*, 46(5):2443–2468, 2008.

[5] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1993.

[6] M. Hochbruck and A. Ostermann. Exponential multistep methods of Adams-type. *BIT*, 51(4):889–908, 2011.

[7] S. Norsett. An A-stable modification of the Adams-Bashforth methods. In J. L. Morris, editor, *Conference on the Numerical Solution of Differential Equations: Held in Dundee/Scotland*, pages 214–219. Springer, 1969.

[8] M. Perego and A. Veneziani. An efficient generalization of the Rush-Larsen method for solving electro-physiology membrane equations. *ETNA*, 35:234–256, 2009.

[9] S. Rush and H. Larsen. A practical algorithm for solving dynamic membrane equations. *IEEE Trans Biomed Eng*, 25(4):389–92, July 1978.

[10] R. J. Spiteri and R. C. Dean. Stiffness analysis of cardiac electrophysiological models. *Ann. Biomed. Eng.*, 38:3592–3604, 2010.

[11] J. Sundnes, R. Artebrant, O. Skavhaug, and A. Tveito. A second-order algorithm for solving dynamic cell membrane equations. *IEEE Transactions on Biomedical Engineering*, 56(10):2546–2548, 2009.

[12] K. ten Tusscher, D. Noble, P. Noble, and A. Panfilov. A model for human ventricular tissue. *Am. J. Physiol. Heart Circ. Physiol.*, 286(4):H1573–H1589, 2004.