Efficient high order schemes for stiff ODEs in cardiac electrophysiology

C. Douanla Lontsi* , Y. Coudière, C. Pierre

INRIA Bordeaux Sud Ouest
Université de Bordeaux
* charlie.douanla-lontsi@inria.fr

ABSTRACT. In this work, we introduce some exponential high order schemes for stiff $ODEs$ coming from the models used in cardiac electrophysiology. We show in this context that despite the stiffness of the equations, the use of high order (order 3) stabilized schemes is beneficial in terms of cost with the possibility to use large time-steps as for implicit schemes. We describe a methodology and introduce some tools allowing to compare the numerical schemes used to solve ODEs. These tools and methodology are then used through the Beeler Reuter ionic model (BR) [2] to compare the exponential schemes with the classical explicit and implicit schemes at various orders. It comes from this comparisons a good alternative in terms of cost, accuracy and ease of implementation.

RÉSUMÉ. Dans ce travail, nous définissons des schémas d'ordre élevé de type exponentiels appliqués aux $EDOs$ à caractères raides provenant des modèles utilisés en électrophysiologie cardiaque. Nous montrons dans ce contexte que malgré la raideur des équations, le recours à l'ordre élevé (ordre 3) des schémas stabilisés est profitable en terme de coût avec une possibilité d'utiliser des pas de temps aussi grands que lors de l'utilisation des schémas implicites. Nous décrivons des outils de comparaisons et une méthodologie permettant de comparer les schémas numériques pour la résolution des EDOs. Cette méthodologie est ensuite utilisée à travers le modèle ionique Beeler Reuter (BR) [2] pour comparer en terme de coût et de précision les schémas exponentiels aux schémas classiques implicites et explicites à différents ordres. Il en ressort de cette comparaison l'apport d'une bonne alternative à la fois en terme de coût, de précision et de facilité d'implémentation.

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

MOTS-CLÉS : Electrophysiologie cardiaque, schémas d'ordre élevé, équations aux dérivées ordinaires raides, schémas exponetiels

1. Introduction

The numerical resolution of stiff ordinary differential equations $ODEs$ is an issue encountered in many field of applied sciences. In cardiac electrophysiology, the model describing the electrical activity of the heart is a system of parabolic partial differential equations coupled with a system of $ODEs$ called *ionic models* that is highly nonlinear and exhibiting a stiff behavior, making their numerical resolution very challenging. The classical schemes have serious drawbacks to solve such $ODEs$. On one hand classical stable methods are implicit and lead to high computational cost associated with large time-steps due to nonlinear solvers, on the other hand explicit solvers require very small time steps also leading to high computational costs. Meanwhile current solvers in the field are usually based on order 1 or 2 schemes. 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,

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

EAB and RL schemes take advantage of a splitting of the model function F into some linear part a and a nonlinear part b,

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

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,

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

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 formulation (3), the exact solution satisfies the variation of the constant formula,

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

The aim of this paper is to study the efficiency of EAB and RL methods of order 1 up to 4. This efficiency is analyzed both in terms of accuracy and of 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 [3]).

The paper is organized as follows. In section 2 are presented the stabilized schemes. A brief description of the transmembrane action potential and ionic model is given in the section 3. The methodology used to compared the methods are developed in the section 4. The comparison of the methods follows in the section 5 where the methodology defined is used to compare the numerical schemes applied to the Beeler Reuter (BR) ionic model [2].

2. EAB_k and RL_k scheme statements

When the function $c_n(t,y)$ in (3) is a polynomial $P_n = \sum_{j=0}^{k-1} p_j(t-t_n)^j$, the relation (4) becomes $y_{n+1} = \mathrm{e}^{\alpha_n h} y_n + h \sum_{j=0}^{k-1} p_j j! h^j \varphi_{j+1}(\alpha_n h)$, with $\varphi_0(z) = \mathrm{e}^z$ and $\varphi_j(0) = \frac{1}{j!}$, $j \geq 0$. The schemes introduced in the sequel are multi-steps. We will use notation $a_n = a(t_n, y_n)$, $b_n = b(t_n, y_n)$.

– EAB_k : On one hand we set $\alpha_n = a_n$, on 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 $t_n, \dots, t_{n-k+1}$. This polynomial satisfies $\tilde{c}_n(t_{n-j}) = c_n(t_{n-j}, y_{n-j})$ for $j = 0, \dots, k-1$. The values $c_n(t_{n-j}, y_{n-j})$ for $j = 0, \dots, k-1$ are given by $c_n^{n-j} = b_{n-j} + (a_{n-j} - a_n) y_{n-j}$. 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 the table bellow.

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}	–	–	–	$c_n^n - 3c_n^{n-1} + 3c_n^{n-2} - c_n^{n-3}$

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

– 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 [1]. 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).

- $k = 1 : \alpha_n = a_n \quad , \quad \beta_n = b_n$.

- $k = 2 : \alpha_n = \frac{3}{2}a_n - \frac{1}{2}a_{n-1}$ and $\beta_n = \frac{3}{2}b_n - \frac{1}{2}b_{n-1}$.

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

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

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

The previous descriptions of the EAB_k and RL_k schemes here have been given very briefly but, more details for the EAB_k schemes can be found in [5] (for general $ODEs$) and in [4] for cardiac electrophysiology application, where the two methods are shown to be stable under perturbations and convergent at the order k.

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 ion 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 of 10 mV) in which it is maintained during a few tenth of seconds before returning to its resting state (see figure 1).

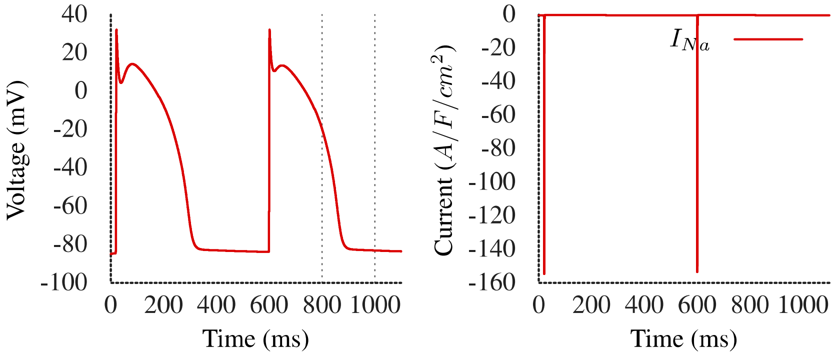

Figure 1. *BR Model [2] illustration. Left, two cellular action potentials : starting at a negative resting value, the transmembrane voltage $V(t)$ has a stiff depolarization followed by a plateau and repolarizing to the resting value. Right : each depolarization is induced by an ionic sodium current $I_{Na}(t)$*

3.2. Ionic Models

The variations of the ionic concentrations are described by ionic models and are systems of ODE. The innovated ones are consisting of the following variables with their associated ordinary differential equation.

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

$$\frac{dV}{dt} = -I_{ion}(y(t), I_{st}(t)) + I_{st}(t), \tag{7}$$

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 $i = 1 \ldots P$,

$$\frac{dW_i}{dt} = \frac{W_{\infty,i}(y) - W_i}{\tau_i(y)}, \tag{8}$$

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 previous variables can be collected in a vector $y \in \mathbb{R}^N$ 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 and X correspond to the lines of (1) for which the linear and nonlinear part is given or null respectively. 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, $A_1(t,y) = \text{Diag}(-1/\tau_i(y))$, and $B_1(t,y) = \{W_{\infty,i}(y)/\tau_i(y), i = 1 \dots P\} \in \mathbb{R}^P$.

4. Scheme analysis methods

– **Test case :** The evaluation and comparisons between different ODE solvers is done with a test case. Specifically, the Beeler Reuter [2] model is considered and written in the form (2) as described in the 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 (7) 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, in the range of 50 mA. We also impose to I_{st} a C^4 regularity in oder to observe the convergence orders of schemes up to 4.

– **Numerical solution:** Let $m \geq 1$ be an integer for which one associated the time-step $h = T/m$ and the regular mesh $\mathcal{T}_m = \{t_n = jh,\ j = 0 \dots m\}$ 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 dt and a mesh $\mathcal{T}_m$, 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 test case given, 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 $(0,T]$. Otherwise, we require to the interpolate $\pi_{m,i} y^n$ 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 $s = 0 \dots m/3$. 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 $n = 0 \dots m$. 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 here 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 associated. We denote $\pi y^n = \widehat{V}$ and $\pi Y_{ref} = \widehat{V}_{ref}$ the membrane potential interpolating associated. 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 :

$$e_\infty = \frac{\max |\widehat{V} - \widehat{V}_{ref}|}{\max |\widehat{V}_{ref}|}. \tag{9}$$

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 one associate it a cost. Here it is a *computational* cost and is evaluated with the CPU time during a simulation. It is evaluated by the fortran90 software for each simulation setting by a time-step h. The CPU times depend on the computer used to perform the solutions. This is balanced by using the ratio between them for comparisons.

5. Numerical results

5.1. Accuracy

The relative error $e(h)$ is computed for various time-steps h and collected in the table 2 where it can be observed that all the methods exhibit the expected order of convergence. A general view of the 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 $h = 0.2$.

The 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. The 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. The 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.

5.2. Cost

A general observation of the 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%.

(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}

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

The 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.

The 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.

The figure 2 on the bottom right shows that the order 4 stabilized schemes are less costly than the classical explicit schemes but it is much more better to use the RL_4 scheme instead of the EAB_4 scheme. Because of they stability properties the 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.

6. Conclusion

Two families of explicit high order stabilized methods (EAB_k, RL_k) have been introduced 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. 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.

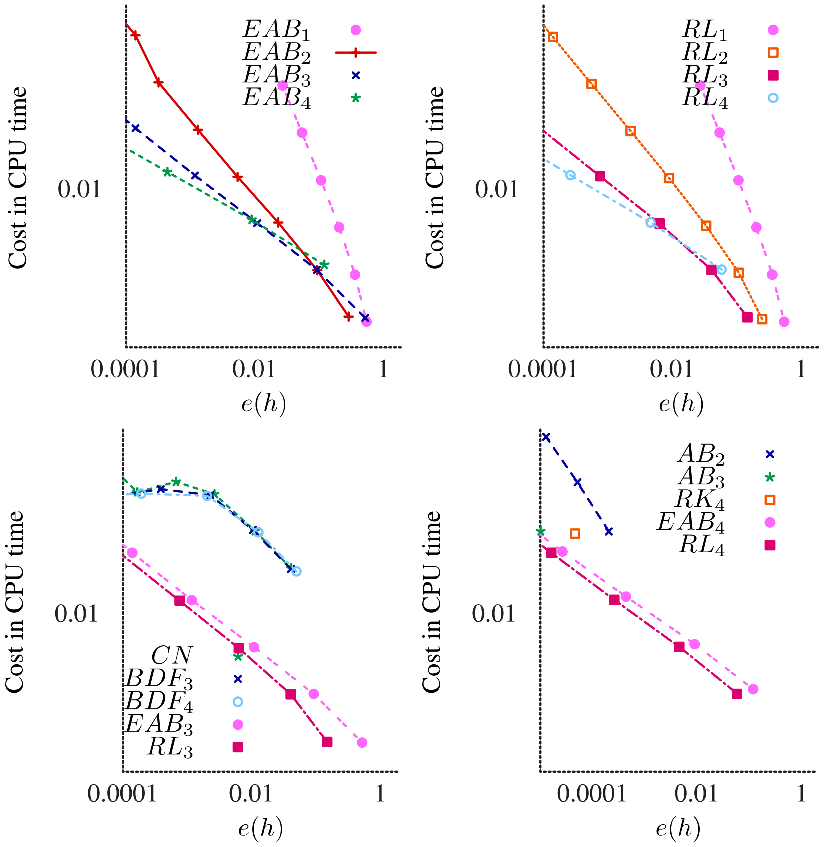

Figure 2. *The CPU time plotted in Log/Log scale against the error for various schemes*

7. References

[1] M. Perego, A.Veneziani, “An efficent generalization of the Rush-Larsen method for solving electro-physiology membrane equations”, *ETNA*, vol. 35, 2009.

[2] G.W. Beeller, H. Reuter, “Reconstruction of the Action Potential of Ventricular Myocardial Fibres”, *J. Physiol*, vol. 268, 1977.

[3] E. Hairer, S.P. Norsett, G. Wanner, “Solving ordinary differential equations I”, *Springer-Verlag, Berlin*, vol. 8, 1993.

[4] Y. Coudiére , C. Douanla Lontsi, C. Pierre, “High order Rush Larsen solver for stiff ODEs”, *Hal*, 2016.

[5] M. Hochbruck, A. Ostermann, “Exponential multistep methods of Adams-type”, *BIT*, vol. 51, 2011.