Exponential Adams Bashforth integrators for stiff ODEs, application to cardiac electrophysiology

Yves Coudière, Charlie Douanla Lontsi, Charles Pierre

Exponential Adams-Bashforth integrators for stiff ODEs, application to cardiac electrophysiology

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

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

April, 2018

Keywords: stiff equations, explicit high-order multistep methods, exponential integrators, stability and convergence, Dahlquist stability

Acknowledgments. This study received financial support from the French Government as part of the "Investissement d'avenir" program managed by the National Research Agency (ANR), Grant reference ANR-10-IAHU-04. It also received fundings of the project ANR-13-MONU-0004-04.

Abstract

Models in cardiac electrophysiology are coupled systems of reaction diffusion PDE and of ODE. The ODE system displays a very stiff behavior. It is non linear and its upgrade at each time step is a preponderant load in the computational cost. The issue is to develop high order explicit and stable methods to cope with this situation.

In this article is analyzed the resort to exponential Adams Bashforth (EAB) integrators in cardiac electrophysiology. The method is presented in the framework of a general and varying stabilizer, that is well suited in this context. Stability under perturbation (or 0-stability) is proven. It provides a new approach for the convergence analysis of the method. The Dahlquist stability properties of the method is performed. It is presented in a new framework that incorporates the discrepancy between the stabilizer and the system Jacobian matrix. Provided this discrepancy is small enough, the method is shown to be A(alpha)-stable. This result is interesting for an explicit time-stepping method. Numerical experiments are presented for two classes of stiff models in cardiac electrophysiology. They include performances comparisons with several classical methods. The EAB method is observed to be as stable as implicit solvers and cheaper at equal level of accuracy.

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

Introduction

Computations in cardiac electrophysiology have to face two constraints. Firstly the stiffness due to heterogeneous time and space scales. This is usually dealt with by considering very fine grids. That strategy is associated with large computational costs, still challenging in dimension 3. Secondly, the resolution of the reaction terms that appears in the ionic models has an important cost. That resolution occur at each grid node. The total amount of evaluation of the reaction terms has to be maintained as low as possible by the considered numerical method. Implicit solvers therefore are usually avoided. Exponential integrators are well adapted to cope with these two constraints. Actually they both allow an explicit resolution of the reaction term and display strong stability properties. In this article is studied and analyzed exponential time-stepping methods dedicated to the resolution of reaction equations.

Models for the propagation of the cardiac action potential are evolution reaction diffusion equations coupled with ODE systems. The widely used monodomain model [7, 8, 9] formulates as,

$$\frac{\partial v}{\partial t} = Av + f_1(v, w, x, t), \quad \frac{\partial w}{\partial t} = f_2(v, w, x, t),$$

with space and time variables $x \in \Omega \subset \mathbb{R}^d$ and $t \in \mathbb{R}$. The unknowns are the functions $v(t, x) \in \mathbb{R}$ (the transmembrane voltage) and $w(t, x) \in \mathbb{R}^N$ (a vector that gathers variables describing pointwise the electrophysiological state of the heart cells). In the monodomain model, the diffusion operator is $A(:= \mathrm{div}(g(x)\nabla\cdot))$, and the reaction terms are the nonlinear functions f_1, f_2. These functions model the cellular electrophysiology. They are called ionic models. Ionic models are of complex nature, see e.g. [2, 27, 34, 23]. A special attention has to be paid to the number of evaluations of the functions f_1 and f_2, and implicit solvers are usually avoided. Though we ultimately use an implicit/explicit method to solve the PDE, we need an efficient, fast and robust method to integrate the reaction terms. Therefore, this article focuses on the time integration of the stiff ODE system

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \quad y(0) = y_0, \tag{1}$$

in the special cases where $f(t, y)$ is an ionic model from cellular electrophysiology. For that case, stiffness is due to the co existence of fast and slow variables. Fast variables are given in (1) by equations of the form,

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = f_i(t, y) = a_i(t, y)y_i + b_i(t, y). \tag{2}$$

Here $a_i(t, y) \in \mathbb{R}$ is provided by the model. This scalar rate of variation will be inserted in the numerical method to stabilize its resolution.

Exponential integrators are a class of explicit methods meanwhile exhibiting strong stability properties. They have motivated many studies along the

past 15 years, among which we quote e.g. [17, 11, 18, 21, 36, 26] and refer to [28, 19, 16] for general reviews. They already have been used in cardiac electrophysiology, as e.g. in [31, 3]. Exponential integrators are based on a reformulation of (1) as,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a(t,y)y + b(t,y), \quad y(0) = y_0, \tag{3}$$

(with $f = ay + b$) where the linear part $a(t,y)$ is used to stabilize the resolution. Basically $a(t,y)$ is assumed to capture the stiffest modes of the Jacobian matrix of system (1). Stabilization is brought by performing an exact integration of these modes. This exact integration involves the computation of the exponential $\exp(a(t_n,y_n)h)$ at the considered point. This computation is the supplementary cost for exponential integrators as compared to other time stepping methods.

Exponential integrators of Adams type are explicit multistep exponential integrators. They were first introduced by Certaine [5] in 1960 and Nørsettin [29] in 1969 for a constant linear part $A = a(t,y)$ in (3). The schemes are derived using a polynomial interpolation of the non linear term $b(t,y)$. It recently received an increasing interest [35, 4, 30] and various convergence analysis have been completed in this particular case [20, 1, 24]. Non constant linear parts have been less studied. Lee and Preiser [25] in 1978 and by Chu [6] in 1983 first suggested to rewrite the equation (1) at each time instant t_n as, rewritten as,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a_n y + g_n(t,y), \quad y(t_n) = y_n, \tag{4}$$

with $a_n = a(t_n,y_n)$ and $g_n(t,y) = b(t,y) + (a(t,y) - a_n)y$. In the sequel, a_n is referred to as the *stabilizer*. It is updated after each time step. Recently, Ostermann *et al* [20, 24] analyzed the linearized exponential Adams method, where the stabilizer a_n is set to the Jacobian matrix of $f(t,y)$ in (1). This choice requires the computation of a matrix exponential at every time step. Moreover, when the fast variables of the system are known, stabilization can be performed only on these variables. Considering the full Jacobian as the stabilizer implies unnecessary computational efforts. To avoid these problems, an alternative is to set the stabilizer as a part or as an approximation of the Jacobian. This has been analyzed in [37] and in [32] for exponential Rosenbrock and exponential Runge Kutta type methods respectively. That strategy is well adapted to cardiac electrophysiology, where a diagonal stabilizer associated with the fast variable is directly provided by the model with equation (2). The present contribution is to analyze general varying $a(t,y)$ in (3) for exponential integrators of Adams type, referred to as *exponential Adams-Bashforth*, and shortly denoted EAB. Together with the EAB scheme, we introduce a new variant, that we called *integral exponential Adams-Bashforth*, denoted I-EAB.

The convergence analysis held in [20] extends to the case of general varying stabilizers. However there is a lack of results concerning the stability in that

case: for instance, consider the simpler exponential Euler method, defined by

$$y_{n+1} = s(t_n, y_n, h) := \mathrm{e}^{a_n h}\, y_n + h\varphi_1(a_n h) b_n, \quad \varphi_1(z) = (\mathrm{e}^z - 1)/z.$$

Stability under perturbation (also called 0-stability) can be easily proven provided that the scheme generator $s(t, y, h)$ is globally Lipschitz in y with a constant bounded by $1 + Ch$. Therefore stability under perturbation classically is studied by analyzing the partial derivative $\partial_y s$. This can be done in the case where $a(t, y)$ is either a constant operator or a diagonal varying matrix. In the general case however things turn out to be more complicated. The reason is that we do not have the following general series expansion, as $\epsilon \to 0$,

$$e^{M+\epsilon N} \neq e^M + \epsilon e^M N + O(\epsilon^2),$$

unless the two matrices M and N are commuting. *As a consequence differentiating* $\mathrm{e}^{a(t,y)h}$ *in* y *cannot be done without very restrictive assumptions on* $a(t, y)$. We present here a stability analysis for general varying stabilizers. This will be done by introducing relaxed stability conditions on the scheme generator $s(t, y, h)$. Together with a consistency analysis, it provides a new proof for the convergence of the EAB schemes, in the spirit of [20].

Stability under perturbation provides results of qualitative nature. In addition, the Dahlquist stability analysis strengthens these results. It is a practical tool that allows to dimension the time step h with respect to the variations of $f(t, y)$ in equation (1). The analysis is made by setting $f(t, y) = \lambda y$ in (1). For exponential integrators with general varying stabilizer, the analysis must incorporate the decomposition of $f(t, y) = \lambda y$ used in (3). The stability domain of the considered method will depend on the relationship between λ and $a(t, y)$, following a concept first introduced in [31]. We numerically establish that EAB methods are $A(\alpha)$ stable provided that the stabilizer is sufficiently close to the system Jacobian matrix (precise definitions are in section 4). Moreover the angle α approaches $\pi/2$ when the stabilizer goes to the system Jacobian matrix. In contrast, there exists no $A(0)$ stable explicit linear multistep method (see [15, chapter V.2]. This property is remarkable for explicit methods.

Numerical experiments for the EAB and I-EAB scheme are provided in section 6, in the context of cardiac electrophysiology. Robustness to stiffness is studied with this choice. It is numerically shown to be comparable to implicit methods both in terms of accuracy and of stability condition on the time step. We conclude that EAB methods are well suited for solving stiff differential problems. In particular they allow computations at large time step with good accuracy properties and cheap cost.

The article is organized as follows. The EAB and I-EAB methods are introduced in section 1. The general stability and convergence results are stated and proved in section 2. The EAB and I-EAB stability under perturbation and convergence are proved in section 3. The Dahlquist stability is investigated in section 4, and the numerical experiments end the article, in section 6.

In all this paper, $h > 0$ is a constant time-step and $t_n = nh$ are the time instants associated with the numerical approximate y_n of the solution of the ODE (1).

1 Scheme definitions

1.1 The EAB$_k$ method

The exact solution at time t_{n+1} to the equation (4) (with $a_n = a(t_n, y_n)$) is given by the variation of the constants formula:

$$y(t_{n+1}) = e^{a_n h}\left(y(t_n) + \int_0^h e^{-a_n \tau} g_n(t_n + \tau, y(t_n + \tau))\,\mathrm{d}\tau\right). \tag{5}$$

Using the k approximations $y_{n-j} \simeq y(t_{n-j})$ for $j = 0 \dots k-1$, we build the Lagrange polynomial $\tilde{g}_n$ of degree at most $k-1$ that satisfies,

$$\tilde{g}_n(t_{n-j}) = g_{nj} := g(t_{n-j}, y_{n-j}), \quad 0 \le j \le k-1. \tag{6}$$

It provides the numerical approximation $y_{n+1} \simeq y(t_{n+1})$ as

$$y_{n+1} = \mathrm{e}^{a_n h}\left(y_n + \int_0^h \mathrm{e}^{-a_n \tau}\, \tilde{g}_n(t_n + \tau)\,\mathrm{d}\tau\right). \tag{7}$$

The Taylor expansion of the polynomial $\tilde{g}_n$ is $\tilde{g}_n(t_n+\tau) = \sum_{j=1}^{k} \frac{\gamma_{nj}}{(j-1)!}\,(\tau/h)^{j-1}$, where the coefficients γ_{nj} are uniquely determined by (6), and actually given in table 1 for $k = 1, 2, 3, 4$. An exact integration of the integral in equation (7) may be performed:

$$y_{n+1} = \mathrm{e}^{a_n h}\, y_n + h \sum_{j=1}^{k} \varphi_j(a_n h)\gamma_{nj}, \tag{8}$$

where the functions φ_j, originally introduced in [29], are recursively defined (for $j \ge 0$) by

$$\varphi_0(z) = \mathrm{e}^z, \quad \varphi_{j+1}(z) = \frac{\varphi_j(z) - \varphi_j(0)}{z} \quad \text{and} \quad \varphi_j(0) = \frac{1}{j!}. \tag{9}$$

The equation (8) defines the Exponential Adams-Bashforth method of order k, denoted by EAB$_k$.

Remark 1. When $a(t, y) = \mathrm{diag}(d_i)$ is a diagonal matrix, $\varphi_k(a_n h) = \mathrm{diag}(\varphi_k(d_i))$ can be computed component-wise. Its computation is straightforward.

Remark 2. With the definition (9), the functions φ_k are analytic on the whole complex plane. Therefore the EAB$_k$ scheme definition (8) makes sense for a matrix term $a(t, y)$ in equation (3) without particular assumption.

Remark 3. The computation of y_{n+1} in the formula (8) requires the computation of $\varphi_j(a_n h)$ for $j = 0, \dots, k$. This computational effort can be reduced with the recursive definition (9). In practice only $\varphi_0(a_n h)$ needs to be computed. This is detailed in section 6.1.

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

k	1	2	3	4
γ_{n1}	g_n	g_n	g_n	g_n
γ_{n2}		$g_n - g_{n-1}$	$\frac{3}{2}g_n - 2g_{n-1} + \frac{1}{2}g_{n-2}$	$\frac{11}{6}g_n - 3g_{n-1} + \frac{3}{2}g_{n-2} - \frac{1}{3}g_{n-3}$
γ_{n3}			$g_n - 2g_{n-1} + g_{n-2}$	$2g_n - 5g_{n-1} + 4g_{n-2} - g_{n-3}$
γ_{n4}				$g_n - 3g_{n-1} + 3g_{n-2} - g_{n-3}$

1.2 A variant: the I-EAB$_k$ method

If the matrix $a(t, y)$ is diagonal, we can take advantage of the following version for the variation of the constants formula:

$$y(t_{n+1}) = \mathrm{e}^{A_n(h)} \left(y(t_n) + \int_0^h \mathrm{e}^{-A_n(\tau)}\, b(y(t_n+\tau), t_n+\tau)\, \mathrm{d}\tau \right),$$

where $A_n(\tau) = \int_0^\tau a(t_n + \sigma, y(t_n + \sigma))d\sigma$. An attempt to improve the EAB_k formula (8) is to replace $a(t, y)$ and $b(t, y)$ in the integral above by their Lagrange interpolation polynomials. At time t_n, we define the two polynomials $\tilde{a}_n$ and $\tilde{b}_n$ of degree at most $k-1$ so that:

$$\tilde{a}_n(t_{n-j}) = a(t_{n-j}, y_{n-j}), \quad \tilde{b}_n(t_{n-j}) = b(t_{n-j}, y_{n-j}), \quad j = 0 \dots k-1,$$

and the primitive $\tilde{A}_n(\tau) = \int_0^\tau \tilde{a}(t_n+\sigma, y(t_n+\sigma))d\sigma$. The resulting approximate solution at time t_{n+1} is finally given by the formula

$$y_{n+1} = \mathrm{e}^{\tilde{A}_n(h)} \left(y_n + \int_0^h \mathrm{e}^{-\tilde{A}_n(\tau)}\, \tilde{b}_n(t_n+\tau)\, \mathrm{d}\tau \right). \tag{10}$$

The method is denoted I-EAB$_k$ (for Integral EAB$_k$). Dislike for the formula (7), no exact integration formula is available, because of the term $\mathrm{e}^{-\tilde{A}_n(\tau)}$. A quadrature rule is required for the actual numerical computation of the integral in formula (10). Implementation details are given in section 6.1.

2 Stability conditions and convergence

The equation (1) is considered on a finite dimensional vector space E with norm $|\cdot|_E$. We fix a final time $T > 0$ and assume that equation (1) has a solution y on $[0, T]$. We adopt the general settings for the analysis of k-multistep methods following [14]. The space E^k is equipped with the maximum norm $|Y|_\infty = \max_{1\le i\le k} |y_i|_E$ with $Y = (y_1, \dots, y_k) \in E^k$. A k-multistep scheme is defined by a mapping,

$$s : \ (t, Y, h) \in \mathbb{R} \times E^k \times \mathbb{R}^+ \mapsto s(t, Y, h) \in E.$$

For instance, the EAB_k scheme rewrites as $y_{n+1} = s(t_n, Y, h)$ with $Y = (y_{n-k+1}, \ldots, y_n)$ in the formula (8). The scheme generator is the mapping S given by

$$S \ : (t, Y, h) \in \mathbb{R} \times E^k \times \mathbb{R}^+ \longrightarrow (y_2, \ldots, y_k, s(t, Y, h)) \in E^k.$$

A numerical solution is a sequence (Y_n) in E^k for $n \geq k-1$ so that

$$Y_{n+1} = S(t_n, Y_n, h) \quad \text{for} \quad n \geq k-1, \tag{11}$$

$Y_{k-1} = (y_0, \ldots, y_{k-1})$ being its initial condition. A perturbed numerical solution is a sequence (Z_n) in E^k for $n \geq k-1$ such that

$$Z_{k-1} = Y_{k-1} + \xi_{k-1}, \quad Z_{n+1} = S(t_n, Z_n, h) + \xi_{n+1} \quad \text{for} \quad n \geq k-1, \tag{12}$$

with $(\xi_n) \in E^k$ for $n \geq k-1$. The scheme is said to be stable under perturbation (or 0-stable) if: being given a numerical solution (Y_n) as in (11) there exists a (stability) constant $L_s > 0$ so that for any perturbation (Z_n) as defined in (12) we have,

$$\max_{k-1 \leq n \leq T/h} |Y_n - Z_n|_\infty \leq L_s \sum_{k-1 \leq n \leq T/h} |\xi_n|_\infty. \tag{13}$$

Proposition 1. Assume that there exists constants $C_1 > 0$ and $C_2 > 0$ such that

$$1 + |S(t, Y, h)|_\infty \leq (1 + |Y|_\infty)\,(1 + C_1 h), \tag{14}$$
$$|S(t, Y, h) - S(t, Z, h)|_\infty \leq |Y - Z|_\infty\,(1 + C_2 h(1 + |Y|_\infty))\,, \tag{15}$$

for $0 \leq t \leq T$, and for $Y, Z \in E^k$. Then, the numerical scheme is stable under perturbation with the constant L_s in (13) given by

$$L_s = \mathrm{e}^{C^\star T}, \quad C^\star := C_2\,\mathrm{e}^{C_1 T}\,(1 + |Y_{k-1}|_\infty)\,. \tag{16}$$

Proof. Consider a numerical solution (Y_n) in (11). A recursion on condition (14) gives,

$$1 + |Y_n|_\infty \leq (1 + |Y_{k-1}|_\infty)\,(1 + C_1 h)^{n-k+1} \leq \mathrm{e}^{C_1 T}\,(1 + |Y_{k-1}|_\infty)\,,$$

since $(1+x)^p \leq \mathrm{e}^{px}$ and $(n-k+1)h \leq nh \leq T$. Now, consider a perturbation (Z_n) of (Y_n) given by (12). Using the condition (15) together with the previous inequality,

$$\begin{aligned}
|Y_{n+1} - Z_{n+1}|_\infty &\leq |S(t_n, Y_n, h) - S(t_n, Z_n, h)|_\infty + |\xi_{n+1}|_\infty \\
&\leq |Y_n - Z_n|_\infty\,(1 + C_2 h(1 + |Y_n|_\infty)) + |\xi_{n+1}|_\infty \\
&\leq |Y_n - Z_n|_\infty\,\big(1 + C_2\,\mathrm{e}^{C_1 T}(1 + |Y_{k-1}|_\infty)h\big) + |\xi_{n+1}|_\infty \\
&\leq |Y_n - Z_n|_\infty\,(1 + C^\star h) + |\xi_{n+1}|_\infty,
\end{aligned}$$

where $C^\star := C_2\,\mathrm{e}^{C_1 T}\,(1+|Y_{k-1}|_\infty)$. By recursion we get,

$$|Y_n - Z_n|_\infty \le (1+C^\star h)^{n-k+1}\,|Y_{k-1}-Z_{k-1}|_\infty + \sum_{i=0}^{n-k}(1+C^\star h)^i\,|\xi_{n-i}|_\infty$$
$$\le (1+C^\star h)^n \sum_{i=k-1}^{n}|\xi_i|_\infty \le \mathrm{e}^{c^\star T}\sum_{i=k-1}^{n}|\xi_i|_\infty,$$

which ends the proof. □

Like in the classical cases, stability under perturbation together with consistency ensures convergence. Let us specify this point. For the considered solution $y(t)$ of problem (1) on $[0,T]$, we define,

$$Y(t) = (y\,(t-(k-1)h)\,,\dots y(t)) \in E^k \quad \text{for} \quad (k-1)h \le t \le T. \tag{17}$$

The local error at time t_n is,

$$\epsilon(t_n,h) = Y(t_{n+1}) - S(t_n, Y(t_n), h). \tag{18}$$

The scheme is said to be consistent of order p if there exists a (consistency) constant $L_c > 0$ only depending on $y(t)$ such that

$$\max_{k-1\le n\le T/h}|\epsilon(t_n,h)|_\infty \le L_c h^{p+1}.$$

Corollary 1. If the scheme satisfies the stability conditions (14) and (15), and is consistent of order p, then a numerical solution (Y_n) given by (11) satisfies,

$$\max_{k-1\le n\le T/h}|Y(t_n)-Y_n|_\infty \le L_s L_c T h^p + L_s|\xi_0|_\infty, \tag{19}$$

where $\xi_0 = Y(t_{k-1}) - Y_{k-1}$ denotes the error on the initial data, and the constant L_s is as in equation (16).

Remark 4. Note that the stability constant L_s in (16) depends on $|Y_{k-1}|_\infty$, and then on h. This is not a problem since L_s can be bounded uniformly as $h \to 0$ for Y_{k-1} in a neighbourhood of y_0.

Proof. We have $Y(t_{k-1}) = Y_{k-1} + \xi_0$ and $Y(t_{n+1}) = S(t_n, Y(t_n), h) + \epsilon(t_n, h)$. Therefore the sequence $(Y(t_n))$ is a perturbation of the numerical solution (Y_n) in the sense of (12). As a consequence, proposition 1 shows that

$$\max_{k-1\le n\le T/h}|Y_n - Y(t_n)|_E \le L_s\left(|\xi_0| + \sum_{k\le n\le T/h}|\epsilon(t_n,h)|\right) \le L_s|\xi_0| + L_s L_c\left(\sum_{k\le n\le T/h} h\right)h^p,$$

and the convergence result follows. □

3 EAB_k and I-EAB_k schemes analysis

The space E is assumed to be $E = \mathbb{R}^N$ with its canonical basis and with $|\cdot|_E$ the maximum norm. The space of operators on E is equipped with the associated operator norm, and associated to $N \times N$ matrices. Thus $a(t, y)$ is a $N \times N$ matrix and its norm $|a(t, y)|$ is the matrix norm associated to the maximum norm on $\mathbb{R}^N$.
It is commonly assumed for the numerical analysis of ODE solvers that f in the equation (1) is uniformly Lipschitz in its second component y. With the formulation (3), the following supplementary hypothesis will be needed: on $\mathbb{R} \times E$,

$$|a(t, y)| \leq M_a, \quad a(t, y),\ b(t, y) \ \text{ and } f(t, y) \ \text{ uniformly Lipschitz in } y. \quad (20)$$

We denote by K_f, K_a and K_b the Lipschitz constant for f, a and b respectively.

Theorem 1. With the assumptions (20), the EAB_k and I-EAB_k schemes are stable under perturbations. Moreover, if a and b are $\mathcal{C}^k$ regular on $\mathbb{R} \times E$, then the EAB_k and I-EAB_k schemes are consistent of order k. Therefore they converge with order k in the sense of inequality (19), by applying corollary 1.

The stability and consistency are proved in sections 3.3 and 3.4, respectively. Preliminary tools and definitions are provided in the sections 3.1 and 3.2.

3.1 Interpolation results

Consider a function $x : \mathbb{R} \times E \longrightarrow \mathbb{R}$ and a triplet $(t, Y, h) \in \mathbb{R} \times E^k \times \mathbb{R}^+$ with $Y = (y_1, \ldots, y_k)$. We set to $\tilde{x}_{[t,Y,h]}$ the polynomial with degree less than $k - 1$ so that

$$\tilde{x}_{[t,Y,h]}(t - ih) = x(t - ih, y_{k-i}), \quad 0 \leq i \leq k - 1.$$

We then extend component-wise this definition to vector valued or matrix valued functions x (*e.g.* the functions a, b or f).

Lemma 1. There exists an (interpolation) constant $L_i > 0$ such that, for any function $x : \mathbb{R} \times E \mapsto \mathbb{R}$,

$$\sup_{t \leq \tau \leq t+h} |\tilde{x}_{[t,Y,h]}(\tau)| \leq L_i \max_{0 \leq i \leq k-1} |x(t - ih, y_{k-i})|, \quad (21)$$

$$\sup_{t \leq \tau \leq t+h} |\tilde{x}_{[t,Y_1,h]} - \tilde{x}_{[t,Y_2,h]}| \leq L_i \max_{0 \leq i \leq k-1} |x(t - ih, y_{1,k-i}) - x(t - ih, y_{2,k-i})|, \quad (22)$$

Consider a function $y : [0, T] \to E$ and assume that x and y have a $\mathcal{C}^k$ regularity. Then, when $[t - (k - 1)h, t + h] \subset [0, T]$,

$$\sup_{t \leq \tau \leq t+h} \left| x(t, y(t)) - \tilde{x}_{[t,Y(t),h]} \right|_E \leq \sup_{[0,T]} \left| \frac{\mathrm{d}^k f}{\mathrm{d}t^k} (f(t, y(t))) \right| h^k, \quad (23)$$

with $Y(t)$ defined in (17).
For a vector valued function in $\mathbb{R}^d$ the previous inequalities hold when considering the max norm on $\mathbb{R}^d$. For a matrix valued function in $\mathbb{R}^d \times \mathbb{R}^d$ this is also true for the operator norm on $\mathbb{R}^d \times \mathbb{R}^d$ when multiplying the constants in the inequalities (21), (22) and (23) by d.

Proof. The space $\mathbb{P}_{k-1}$ of the polynomials p with degree less than $k-1$ is equipped with the norm $\sup_{[0,1]} |p(\tau)|$. On $\mathbb{R}^k$ is considered the max norm. To $R = (r_1, \dots, r_k) \in \mathbb{R}^k$ is associated $\mathcal{L}R \in \mathbb{P}_{k-1}$ uniquely determined by $\mathcal{L}R(-i) = r_{k-1}$ for $i = 0 \dots k-1$. The mapping $\mathcal{L}$ is linear. Let $C_{\mathcal{L}}$ be its continuity constant (it only depends on k).
We fix $x : \mathbb{R} \times E \to \mathbb{R}$ and $(t, h) \in \mathbb{R} \times \mathbb{R}^+$. Consider the vector $Y = (y_1, \dots, y_k) \in E^k$ and set the vector $R \in \mathbb{R}^k$ by $R = (x(t-(k-1)h, y_1), \dots, x(t, y_k))$. We have $\tilde{x}_{[t,Y,h]}(t+\tau) = \mathcal{L}R(\tau/h)$. The relation (21) exactly is the continuity of $\mathcal{L}$ and $L_i = C_{\mathcal{L}}$.
Consider Y_1, $Y_2 \in E^k$ and the associated vectors R_1, R_2 as above. We have $(x_{[t,Y_1,h]} - x_{[t,Y_2,h]})(t+\tau) = \mathcal{L}\,(R_1 - R_2)\,(\tau/h)$. Again, relation (22) comes from the continuity of $\mathcal{L}$.
Let $\varphi : \mathbb{R} \to \mathbb{R}$ be a $\mathcal{C}^k$ function, its interpolation polynomial $\tilde{\varphi}$ at the points $t-(k-1)h, \dots, t$ is considered. A classical result on Lagrange interpolation applied to φ states that, for all $\tau \in (t, t+h)$, there exists $\xi \in (t-(k-1)h, t+h)$, such that $(\varphi - \tilde{\varphi})(\tau) = \frac{1}{k!}\varphi^{(k)}(\xi)\pi(\tau)$, where $\pi(\tau) = \prod_{i=1}^{k}(\tau - t_i)$. For $\tau \in (t, t+h)$, we have $|\pi(\tau)| \le k!\, h^k$. This proves (23) by setting $\varphi(t) = x(t, y(t))$.
For a vector valued function $x : \mathbb{R} \times E \to \mathbb{R}^d$, these three inequalities holds by processing component-wise and when considering the max norm on $\mathbb{R}^d$.
For a matrix valued function $x : \mathbb{R} \times E \to \mathbb{R}^d \times \mathbb{R}^d$, the extension is direct when considering the max norm $|\cdot|_\infty$ on $\mathbb{R}^d \times \mathbb{R}^d$ (*i.e.* the max norm on the matrix entries). The operator norm $|\cdot|$ is retrieved with the inequality $|\cdot|_\infty \le d|\cdot|$ □

3.2 Scheme generators

Let us consider $(t, Y, h) \in \mathbb{R} \times E^k \times \mathbb{R}^+$ with $Y = (y_1, \dots, y_k)$. With the notations used in the previous subsection, we introduce the interpolations $\tilde{a}_{[t,Y,h]}$ and $\tilde{b}_{[t,Y,h]}$ for the functions a and b in (3). Thanks to the definition (10), the I-EAB$_k$ scheme generator is defined by

$$s(t, Y, h) = z(t+h) \quad \text{with} \quad \frac{\mathrm{d}z}{\mathrm{d}\tau} = \tilde{a}_{[t,Y,h]}(\tau)z(\tau) + \tilde{b}_{[t,Y,h]}(\tau), \quad z(t) = y_k, \tag{24}$$

We introduce the polynomial $\bar{g}_{[t,Y,h]}$ with degree less than $k-1$ that satisfies,

$$\bar{g}_{[t,Y,h]}(t - ih) = f(t - ih, y_{k-i}) - a(t, y_k)y_{k-i}, \quad i = 0 \dots k-1.$$

The function $\tilde{g}_n$ in (6) is given by $\tilde{g}_n = \bar{g}_{[t_n,Y_n,h]}$ with $Y_n = (y_{n-k+1} \dots, y_n)$. With the definition (7), the EAB$_k$ scheme generator is defined by

$$s(t, Y, h) = z(t+h) \quad \text{with} \quad \frac{\mathrm{d}z}{\mathrm{d}\tau} = a(t, y_k)z(\tau) + \bar{g}_{[t,Y,h]}(\tau), \quad z(t) = y_k, \tag{25}$$

We will use the fact that $\bar{g}_{[t,Y,h]}$ is the interpolator of the function $g_{t,y_k} : (\tau, \xi) \to f(\tau, \xi) - a(t, y_k)\xi$, precisely:

$$\bar{g}_{[t,Y,h]} = \widetilde{g_{t,y_k}}_{[t,Y,h]}.$$

These scheme generator definitions will allow us to use the following Gronwall's inequality (see [13, Lemma 196, p.150]).

Lemma 2. Suppose that $z(t)$ is a C^1 function on E. If there exist $\alpha > 0$ and $\beta > 0$ such that $|z'(t)|_E \leq \alpha|t| + \beta$ for all $t \in [t_0, t_0 + h]$, then:

$$|z(t)|_E \leq |z(t_0)|_E \, \mathrm{e}^{\alpha h} + \beta h \, \mathrm{e}^{\alpha h} \quad \text{for} \quad t \in [t_0, t_0 + h]. \tag{26}$$

3.3 Stability

With proposition 1 we have to prove the stability conditions (14) and (15). Note that it is sufficient to prove these relations for $h \leq h_0$ for some constant $h_0 > 0$ since the limit $h \to 0$ is of interest here.

3.3.1 Case of the I-EAB$_k$ scheme

Consider $(t, h) \in \mathbb{R} \times \mathbb{R}^+$ and a vector $Y = (y_1, \ldots, y_k) \in E^k$. We simply denote $\tilde{a} = \tilde{a}_{[t,Y,h]}$ and $\tilde{b} = \tilde{b}_{[t,Y,h]}$. The scheme generator is given by (24). We first have to bound $z(t + h)$ where z is given by

$$z' = \tilde{a}z + \tilde{b}, \quad z(t) = y_k.$$

On the first hand, with the interpolation bound (21),

$$\sup_{t \leq \tau \leq t+h} |\tilde{a}(\tau)| \leq L_i \max_{0 \leq i \leq k-1} |a(t - ih, y_{k-1})| \leq L_i M_a.$$

On the second hand, the function $b(t, y)$ is globally Lipschitz in y and thus can be bounded as $|b(t, y)|_E \leq |b(t, 0)|_E + K_b|y|_E \leq R_b(|y|_E + 1)$, for $0 \leq t \leq T$ and for some constant R_b only depending on K_b and on T. Then with the bound (21),

$$\sup_{t \leq \tau \leq t+h} |\tilde{b}(\tau)|_E \leq L_i \max_{0 \leq i \leq k-1} R_b \left(|y_{k-i}|_E + 1\right) \leq L_i R_b(|Y|_\infty + 1).$$

By applying the Gronwall inequality (26), for $0 \leq \tau \leq h$,

$$|z(t + \tau)|_E \leq \mathrm{e}^{L_i M_a h} \left(|y_k|_E + h L_i R_b(|Y|_\infty + 1)\right).$$

Thus, there exists a constant C_1 only depending on L_i, M_a, R_b and h_0 so that, for $0 \leq \tau \leq h$ and for $0 \leq h \leq h_0$,

$$|z(t + \tau)|_E \leq C_1 h + |Y|_\infty (1 + C_1 h). \tag{27}$$

This gives the condition (14) taking $\tau = h$.

For j=1, 2 We consider $Y_j = (y_{j,1}, \dots y_{j,k}) \in E^k$ and denote $\tilde{a}_j = \tilde{a}_{[t,Y_j,h]}$ and $\tilde{b}_j = \tilde{b}_{[t,Y_j,h]}$ the interpolations of the functions a and b. With the definition (24) of the I-EAB$_k$ scheme we have: $|s(t,Y_1,h) - s(t,Y_2,h)|_E = |\delta(t+h)|$ with $\delta = z_1 - z_2$ and with z_j given by

$$z_j' = \tilde{a_j} z_j + \tilde{b_j}, \quad z_j(t) = y_{j,k}.$$

We then have,

$$\delta' = \tilde{a}_1 \delta + r, \quad r := (\tilde{a}_1 - \tilde{a}_2) z_2 + (\tilde{b}_1 - \tilde{b}_2).$$

Using that a and b are Lipschitz in y and with the interpolation bound (22),

$$\sup_{t\le\tau\le t+h} |\tilde{b}_1(\tau) - \tilde{b}_2(\tau)|_E \le L_i K_b |Y_1 - Y_2|_\infty,$$
$$\sup_{t\le\tau\le t+h} |\tilde{a}_1(\tau) - \tilde{a}_2(\tau)| \le L_i K_a |Y_1 - Y_2|_\infty.$$

With the upper bound (27), for $t \le \tau \le t+h \le T$ and for $h \le h_0$,

$$\begin{aligned} |r(\tau)|_E &\le L_i |Y_1 - Y_2|_\infty \left(K_b + K_a \left(C_1 h + |Y_2|_\infty (1 + C_1 h)\right)\right) \\ &\le C |Y_1 - Y_2|_\infty \left(1 + |Y_2|_\infty\right), \end{aligned} \tag{28}$$

For a constant C only depending on h_0, K_a, K_b, L_i and C_1. We finally apply the Gronwall inequality (26). It yields,

$$\begin{aligned} |\delta(t+h)| &\le \mathrm{e}^{L_i M_a h} \left(|y_{1,k} - y_{2,k}|_E + Ch |Y_1 - Y_2|_\infty \left(1 + |Y_2|_\infty\right)\right) \\ &\le |Y_1 - Y_2|_\infty \, \mathrm{e}^{L_i M_a h} \left(1 + Ch \left(1 + |Y_2|_\infty\right)\right), \end{aligned}$$

This implies the second stability condition (15) for $h \le h_0$.

3.3.2 Case of the EAB$_k$ scheme

Consider $(t,h) \in \mathbb{R} \times \mathbb{R}^+$ and a vector $Y = (y_1, \dots, y_k) \in E^k$. Following the definition of the EAB$_k$ scheme given in section 1.1, we denote $\bar{a} = a(t, y_k)$, g the function $g(\tau,\xi) = f(\tau,\xi) - \bar{a}\xi$ and $\bar{g} = \bar{g}_{[t,Y,h]}$. We have that $\bar{g}$ is the Lagrange interpolation polynomial of g, specifically $\bar{g} = \tilde{g}_{[t,Y,h]}$. The scheme generator is then given by equation (25): $s(t,Y,h) = z(t+h)$ with,

$$z' = \bar{a} z + \bar{g}, \quad z(t) = y_k.$$

We first have the bound $|\bar{a}| \le M_a$. As in the previous subsection, f being globally Lipschitz in y, one can find a constant R_f so that for $0 \le t \le T$, $|f(t,y)|_E \le R_f(1 + |y|_E)$. It follows that $|g(\tau,\xi)|_E \le R_f(|y|_E + 1) + M_a |y|_E \le C_0/L_i (|y|_E + 1)$, with $C_0/L_i = R_f + M_a$. Therefore, with the interpolation bound (21):

$$\sup_{t\le\tau\le t+h} |\bar{g}(\tau)|_E \le L_i \max_{0\le i\le k-1} |g(t - ih, y_{k-i})|_E \le C_0 (|y|_E + 1).$$

By applying the Gronwall inequality (26), for $0 \leq \tau \leq h$,

$$|z(t+\tau)|_E \leq \mathrm{e}^{M_a h} \left(|y_k|_E + hC(|Y|_\infty + 1) \right).$$

Thus, there exists a constant C_1 only depending on M_a and C_0 so that, for $0 \leq \tau \leq h$ and for $0 \leq h \leq h_0$, (27) holds. This gives the condition (14).

We now consider Y_1, $Y_2 \in E^k$ for j =1, 2 and denote as previously, $\bar{a}_j = a(t, y_{j,k})$, g_j the function $g_j(\tau, \xi) = f(\tau, \xi) - \bar{a}_j \xi$ and $\bar{g}_j = \bar{g}_{[t, Y_j, h]}$. With (25), $|s(t, Y_1, h) - s(t, Y_2, h)|_E = |\delta(t+h)|$ with $\delta = z_1 - z_2$ and with z_j given by

$$z_j' = \bar{a}_j z_j + \bar{g}_j, \quad z_j(t) = y_{j,k}.$$

The function δ satisfies the ODE

$$\delta' = \bar{a}_1 \delta + r(t), \quad r(t) := (\bar{a}_1 - \bar{a}_2) z_2 + (\bar{g}_1 - \bar{g}_2).$$

We have

$$\begin{aligned} |g_1(\tau, y_{1,i}) - g_2(\tau, y_{2,i})|_E &\leq |f(\tau, y_{1,i}) - f(\tau, y_{2,i})|_E + |\bar{a}_1||y_{1,i} - y_{2,i}|_E + |\bar{a}_1 - \bar{a}_2||y_{2,i}|_E \\ &\leq |Y_1 - Y_2|_\infty \left(K_f + M_a + K_a |Y_2|_\infty \right). \end{aligned}$$

Thus, with equation (22), for some $C > 0$,

$$\begin{aligned} \sup_{t \leq \tau \leq t+h} |\bar{g}_1(\tau) - \bar{g}_2(\tau)|_E &\leq L_i \max_{0 \leq i \leq k-1} |g_1(t - ih, y_{1,k-i}) - g_2(t - ih, y_{2,k-i})|_E \\ &\leq C |Y_1 - Y_2|_\infty \left(1 + |Y_2|_\infty \right). \end{aligned}$$

Meanwhile we have the upper bound (27) that gives, for $t \leq \tau \leq t + h \leq T$ and $h \leq h_0$,

$$\begin{aligned} |(\bar{a}_1 - \bar{a}_2) z_2|_E &\leq M_a \left| Y_1 - Y_2 \right|_\infty |z_2(\tau)|_E \\ &\leq M_a \left| Y_1 - Y_2 \right|_\infty \left(C_1 h + |Y_2|_\infty (1 + C_1 h) \right). \end{aligned}$$

Altogether, we retrieve the upper bound (28) on $r(t)$. We can end the proof as for the I-EAB$_k$ case and conclude that the stability condition (15) holds for the EAB$_k$ scheme.

3.4 Consistency

A solution $y(t)$ to problem (1) on $[0, T]$ is fixed. The functions a and b in (3) are assumed to be ^{k} regular so that y is $\mathcal{C}^{k+1}$ regular.

3.4.1 Case of the EAB$_k$ scheme

The local error (18) for the EAB$_k$ scheme has been analyzed in [20]. That analysis remains valid for the case presented here and we only briefly recall it. The local error is obtained by subtracting (7) to (5):

$$\begin{aligned} |\epsilon(t_n, h)|_E &\leq \int_0^h \mathrm{e}^{M_a(h-\tau)} \left| g_n(t+\tau, y(t+\tau)) - \tilde{g}_n(t+\tau) \right|_E \mathrm{d}\tau \\ &\leq h \varphi_1(M_a h) h^k \sup_{[0,T]} \left| \frac{\mathrm{d}^k}{\mathrm{d}t^k} \left(g_n(t, y(t)) \right) \right|, \end{aligned}$$

thanks to the interpolation error estimate (23). Finally, with the upper bound M_a on a_n, the last term can be bounded independently on n for $h \leq h_0$.

3.4.2 Case of the I-EAB$_k$ scheme

We denote $\tilde{a} = \tilde{a}_{[t_n, Y(t_n), h]}$ and $\tilde{b} = \tilde{b}_{[t_n, Y(t_n), h]}$. The local error (18) for the I-EAB$_k$ scheme satisfies $\epsilon(t_n, h) = |\delta(t_{n+1})|_E$ with $\delta = y - z$ and where z is defined by

$$z' = \tilde{a}z + \tilde{b}, \quad z(t_n) = y(t_n),$$

so that with (24) we have $s(t_n, Y(t_n), h) = z(t_{n+1})$. The function δ is defined with $\delta(t_n) = 0$ and,

$$\delta' = \tilde{a}\delta + r, \quad r(\tau) := (a(\tau, y(\tau)) - \tilde{a}(\tau))y(\tau) + (b(\tau, y(\tau)) - \tilde{b}(\tau)).$$

The following constants only depend on the considered exact solution y, on the functions a and b in problem (3) and on T,

$$C_y = \sup_{[0,T]} |y|_E, \quad C_{a,y} = \sup_{[0,T]} \left| \frac{\mathrm{d}^k}{\mathrm{d}t^k} a(t, y(t)) \right|, \quad C_{b,y} = \sup_{[0,T]} \left| \frac{\mathrm{d}^k}{\mathrm{d}t^k} b(t, y(t)) \right|.$$

With the interpolation bound (23), $|r(\tau)|_E \leq Ch^k$ on $[t_n, t_{n+1}]$ with $C = C_{a,y}C_y + C_{b,y}$. It has already been showed in section 3.3 that $\sup_{[t_n, t_{n+1}]} |\tilde{a}(\tau)| \leq L_i M_a$. Therefore, with the Gronwall inequality (26),

$$\epsilon(t_n, h) = |\delta(t_{n+1})|_E \leq \mathrm{e}^{L_i M_a h}\, hCh^K.$$

Thus the EAB$_k$ scheme is consistent of order k.

4 Dahlquist stability

4.1 Background

The classical framework for the Dahlquist stability analysis is to set $f(t, y)$ in problem (1) to $f = \lambda y$. For linear multistep methods, see e.g. [15], the numerical solutions satisfy $|y_{n+1}/y_n| \leq \rho(\lambda h)$, where $\rho : \mathbb{C} \to \mathbb{R}^+$. The function ρ is the stability function. Its definition is detailed below. The stability domain is defined by $D = \{z \in \mathbb{C}, \quad \rho(z) < 1\}$. The scheme is said to be:

- A stable if $\mathbb{C}^- \subset D$,
- $A(\alpha)$ stable if D contains the cone with axis $\mathbb{R}^-$ and with half angle α,
- $A(0)$ stable if $\mathbb{R}^- \subset D$,
- stiff stable if D contains a half plane Re $z < x \in \mathbb{R}^-$.

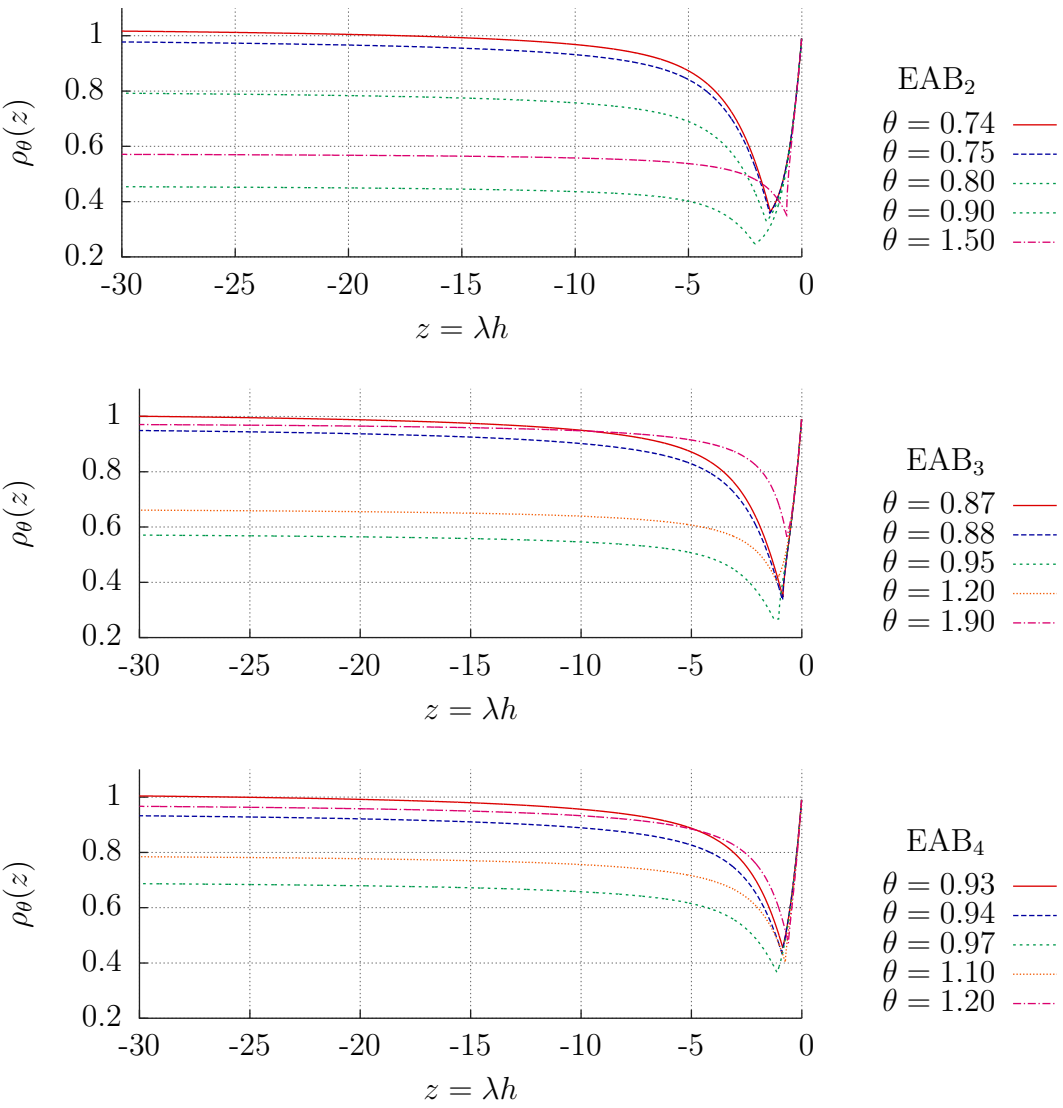

Figure 1: Stability function $\rho_\theta(z)$ for $z \in \mathbb{R}^-$, for various values of θ and for the three schemes EAB$_2$, EAB$_3$ and EAB$_4$.

For exponential integrators, when setting $a(t, y) = \lambda$ in the reformulation (3) of problem (1), the scheme is exact, and therefore also A stable. Such an equality does not hold in general. Then for exponential integrators the Dahlquist stability analysis has to incorporate the relationship between the stabilization term $a(t, y)$ in (3) and the test function $f = \lambda y$. This is done here by considering the splitting,

$$f = \lambda y = ay + b, \quad a = \theta\lambda \quad \text{and} \quad b = \lambda(1 - \theta)y, \tag{29}$$

The parameter $\theta > 0$ controls with what accuracy the exact linear part of f in equation 1 is captured by a in equation 3. In practice $\theta \neq 1$, though we may hope that $\theta - 1$ is small. In that framework, the stability function and the

stability domain depend on θ, following the idea of Perego and Venezziani in [31] (see the remark below). For a fixed θ, the stability function is ρ_θ so that

$$\left|\frac{y_{n+1}}{y_n}\right| \leq \rho_\theta(\lambda h),$$

and the stability domain is $D_\theta = \{z \in \mathbb{C}, \quad \rho_\theta(z) < 1\}$.

Remark 5. In [31] an alternative splitting was introduced:

$$f = \lambda y = ay + b, \quad a = \frac{r}{1+r}\lambda \quad \text{and} \quad b = \frac{1}{1+r}\lambda y,$$

involving the real parameter r. We modified that splitting in order to adress more easily the limit where $a = \lambda$ that corresponds to $r \to \pm\infty$ here whereas it corresponds to $\theta = 1$ in our case. Nevertheless the results can be translated from one splitting to the other with the change of variable $r = \theta/(1-\theta)$.

The stability function $\rho_\theta(z)$ is defined as in the classical case for linear multistep methods, see [15, Ch. V.1]. For a fixed value of θ, the EAB_k scheme applied to the right hand side in equation (29) reads

$$y_{n+1} + c_{\theta,1}(\lambda h)y_n + \ldots + c_{\theta,k}(\lambda h)y_{n-k+1} = 0.$$

The coefficients $c_{\theta,j}$ are explicit (for the EAB_2 scheme they are given by $c_{\theta,1}(z) = -1 - \varphi_1(\theta z)z - \varphi_2(\theta z)(1-\theta)z$ and $c_{\theta,2}(z) = \varphi_2(\theta z)(1-\theta)z$). We fix $z \in \mathbb{C}$ and consider the polynomial $\xi^k + c_{\theta,1}(z)\xi^{k-1} + \ldots + c_{\theta,k}$. It has k complex roots $\xi^1_\theta(z), \ldots, \xi^k_\theta(z)$. The stability function is defined as

$$\rho_\theta(z) = \max_{1\leq j\leq k} |\xi^j_\theta(z)|.$$

The stability function ρ_θ is numerically analyzed. For a given vakue of θ. The software *Maple* is used to compute the polynomial roots on a given grid. That grid either is a 1D grid of $[-30, 0] \subset \mathbb{R}^-$ with size $\Delta x = 0.01$ to study the $A(0)$-stability. Or it is a 2D cartesian grid in $\mathbb{C}$ to study the stability domain. In that case the grid size is $\Delta x = 0.05$ and the domain is $[-40, 2] \times [0, 60]$ (roughly 900 000 nodes). The isoline $\rho_\theta(z) = 1$ then is constructed with the software *gnuplot*.

4.2 $A(0)$ stability

The stability functions $\rho_\theta(z)$ are numerically studied for $z \in \mathbb{R}^-$. These functions have been plotted for different values of the parameter θ. The results are depicted on figure 1. A limit $\lim_{-\infty} |\rho_\theta|$ is always observed. The scheme is $A(0)$ stable when this limit is lower than 1. From Figure 1,

- EAB_2 scheme is $A(0)$ stable if $\theta \geq 0.75$,
- EAB_3 scheme is $A(0)$ stable if $0.88 \leq \theta \leq 1.9$,

- EAB_4 scheme is $A(0)$ stable if $0.94 \leq \theta \leq 1.2$.

Roughly speaking, $A(0)$ stability holds for the EAB_k scheme if the exact linear part of $f(t, y)$ in problem (1) is approximated with an accuracy of 75 %, 85 % or 95% for $k = 2$, 3 or 4 respectively.

4.3 $A(\alpha)$ stability

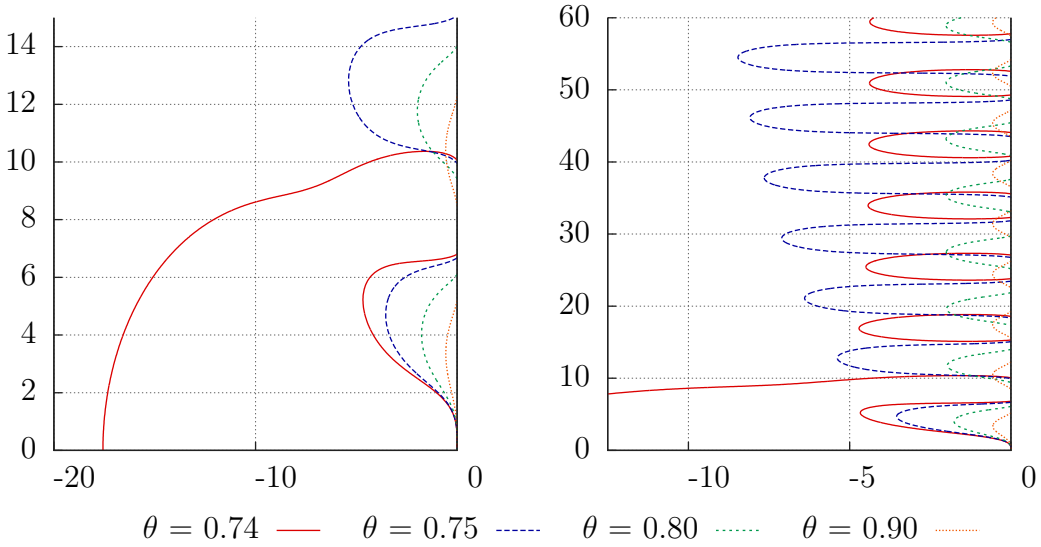

Figure 2: EAB_2: isolines $\rho_\theta(z) = 1$ for two different ranges. The stability domain D_θ is on the left of the isoline.

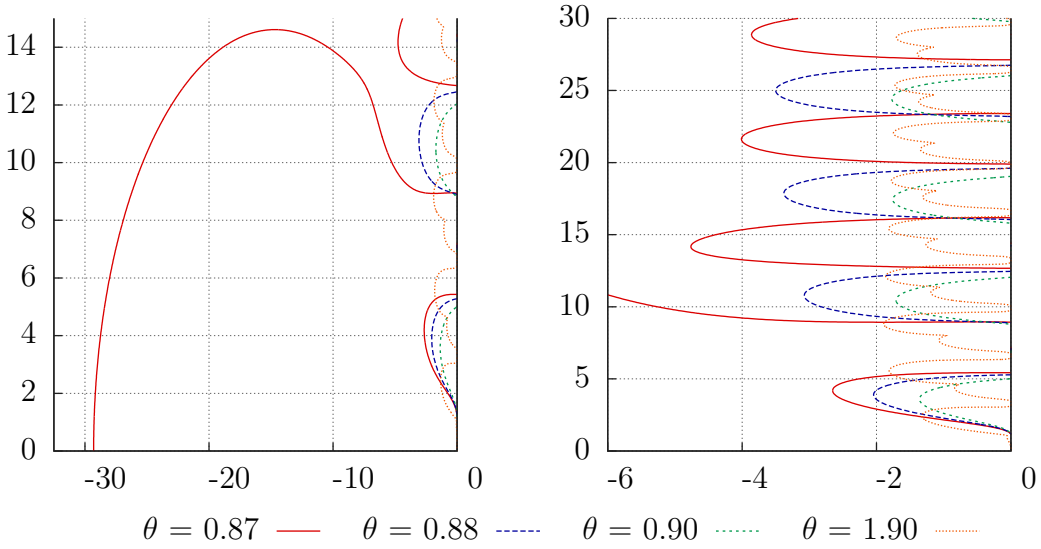

Figure 3: Same thing as figure 2 for the EAB_3 scheme.

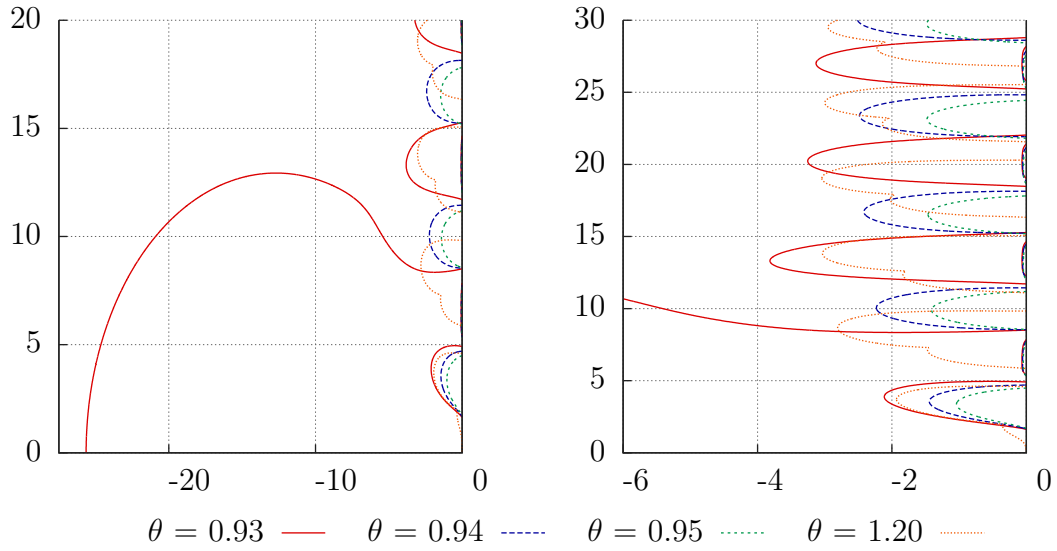

Figure 4: Same thing as figure 2 for the EAB_4 scheme.

The stability domains D_θ have been plotted for various values of θ taken from figure 1. The results are depicted on the figures 2 to 4 for $k = 2$ to 4 respectively. Each figure shows the isolines $\rho_\theta(z) = 1$. The stability domain D_θ is on the left of these curves.

- Figure 2 shows that the EAB_2 scheme is $A(\alpha)$ stable when $\theta = 0.75$, 0.8 and 0.9 with $\alpha \simeq 50$, 60 and 80 angle degrees respectively.
- Figure 3 displays $A(\alpha)$ stability with $\alpha \simeq 60$, 70 and 60 angle degrees for $\theta = 0.88$, 0.9 and 1.9 respectively for the EAB_3 scheme.
- For the EAB_4 scheme eventually, $A(\alpha)$ stability holds with an angle α approximately of 65, 70 and 60 degrees for $\theta = 0.94$, 0.95 and 1.2 respectively, as shown on figure 4.

In all cases, when $A(\alpha)$ stability is observed, the unstable region inside $\mathbb{C}^-$ is made of a discrete collection of uniformly bounded sets located along the imaginary axes. Hence, the stability domain D_θ also contains half planes of the form $\mathrm{Re}(z) \leq a < 0$. We conjecture that, when θ is so that the EAB_k scheme is $A(\alpha)$-stable, then it is also stiff stable.

4.4 Conclusion

For explicit linear multistep methods, $A(0)$ stability cannot occur, see [15, chapter V.2]. In contrast, EAB_k and $\mathrm{I\text{-}EAB}_k$ methods exhibit much better stability properties. When θ is close enough to 1, they are $A(\alpha)$ stable and stiff stable. Such stability properties are comparable with those of implicit linear multistep methods. Basically speaking, these properties will hold if the stabilization term $a(t, y)$ in (3) approximates the Jacobian of $f(t, y)$ in (1) with

an absolute discrepancy lower than 25 %, 10 % and 5 % for k =2, 3 and 4 respectively.
It is interesting to compare the results on the EAB_2 scheme with the results in [31] concerning another explicit exponential scheme of order 2 named AB2*. The AB2* scheme displays stronger absolute stability properties than the EAB_2 scheme. The AB2* scheme is $A(0)-$stable for $\theta \geq 2/3$ whereas the EAB_2 scheme requires $\theta \geq 0.75$. The AB2* scheme is A-stable for $\theta \geq 1$ whuich property is not achieved by the EAB_2 scheme when $\theta \neq 1$. Extensions of that scheme to higher orders have also been developped in [10] that however loses the $A(0)$-stability property.

5 Positivity properties

Consider the scalar equation $y' = f(t, y)$, with $f(t, y)$ chosen so that there exists a function $a(t, y)$ satisfying

$$a(t, y) \leq 0 \quad \text{and} \quad a(t, y)(y - K_1) \leq f(t, y) \leq a(t, y)(y - K_2) \tag{30}$$

for two constants $K_1 \leq K_2$. Then the solution $y(t)$ satisfies $K_1 \leq y(t) \leq K_2$ for $t > 0$ provided that $y(0) \in [K_1, K_2]$. A proof for that property is given in [31]. It can be important for the time-stepping method to preserve that positivity property. This is in particular the case for the application in section 6 where the gating variables $w(t)$ satisfy the property (30) with $K_1 = 0$, $K_2 = 1$ and with $a(t, y)$ defining the stabilizer of the EAB_k scheme.
For the AB_2^* scheme introduced in [31] (that is explicit exponential and with order 2), that property has been shown to be preserved for a constant linear part $a(t, y) = a$ provided that $h \leq \log(3)/|a|$ and adding a condition on the initial data. These results can be extended to the EAB_2 and to the EAB_3 schemes. It provides sufficient condition for the positivity. However the technique of the proof fails to apply to the EAB_4 scheme.
For simplicity in the sequel, we simply denote $\varphi_k = \varphi_k(ah)$.

Proposition 2. Consider the equation $y' = ay + b(t, y)$ with $a \leq 0$ and

$$-aK_1 \leq b(t, y) \leq -aK_2,$$

so that the condition (30) holds for $f(t, y) = ay + b(t, y)$. Denoting y_n the numerical solution for the EAB_k scheme, we have that $y_n \in [K_1, K_2]$ for $n \geq k$ provided that the following conditions are satisfied.

- EAB_2 scheme: $h \leq 1/|a|$ and

$$C_p K_1 \leq \mathrm{e}^{ah}\, y_1 - h\varphi_2 b_0 \leq C_p K_2. \tag{31}$$

  with $C_p = e^{ah} + ah\varphi_2$.

- EAB_3 scheme: $h \leq \beta_3/|a|$ for a constant $\beta_3 \simeq 0.331$ and
$$C_p K_1 \leq \mathrm{e}^{ah}\, y_2 - 2h(\varphi_2+\varphi_3)b_1 \leq C_p K_2. \tag{32}$$
with $C_p = e^{ah} + 2ah\big(\varphi_2+\varphi_3\big)$.

The proof of proposition 2 will be based on the following result.

Lemma 3. Consider the equation $y' = ay + b(t,y)$ with $a \leq 0$ as in proposition 2 and moreover assume that $b(t,y) \geq 0$. Its numerical solution with the EAB_k scheme is denoted y_n.
For the EAB_2 scheme, $y_n \geq 0$ for $n \geq 2$ if:
$$h \leq 1/|a| \quad \text{and} \quad 0 \leq \mathrm{e}^{ah}\, y_1 - h\varphi_2 b_0, \tag{33}$$
For the EAB_3 sceme: $y_n \geq 0$ for $n \geq 3$ if
$$h \leq \beta_3/|a| \quad \text{and} \quad 0 \leq \mathrm{e}^{ah}\, y_2 - 2h(\varphi_2+\varphi_3)b_1, \tag{34}$$
for a constant $\beta_3 \simeq 0.331$.

Proof of lemma 3. In the present case we have $g_n(t,y) = b(t,y)$. For the EAB_2 scheme we have
$$y_{n+1} = \underbrace{\big(\mathrm{e}^{ah}\, y_n - h\varphi_2 b_{n-1}\big)}_{:=s_n} \;+\; \underbrace{h\big(\varphi_1 b_n + \varphi_2 b_n\big)}_{\geq 0}.$$
It suffices to prove that $s_n \geq 0$, which is done by induction. Assume that $s_n \geq 0$. We have $s_{n+1} = \mathrm{e}^{ah}\, y_{n+1} - h\varphi_2 b_n$ and we get
$$s_{n+1} = \mathrm{e}^{ah}\, s_n + hb_n\big(\mathrm{e}^{ah}\,\varphi_1 + \mathrm{e}^{ah}\,\varphi_2 - \varphi_2\big) = \mathrm{e}^{ah}\, s_n + hb_n\varphi_1(\mathrm{e}^{ah} + ah\varphi_2).$$
By assumption $\mathrm{e}^{ah}\, s_n \geq 0$, $b_n \geq 0$ and so $hb_n\varphi_1 \geq 0$. We then must find the condition so that $\mathrm{e}^{ah} + ah\varphi_2 \geq 0$. A simple function analysis gives $ah \in [-1, 0]$ which is the stability condition in lemma 3. The condition (33) states that $s_1 \geq 0$.

For the EAB_3 scheme we define
$$s_n = \mathrm{e}^{ah}\, y_n - 2hb_{n-1}(\varphi_2+\varphi_3)$$
so that we have
$$y_{n+1} = s_n \;+\; hb_n\big(\varphi_1 + 3/2\varphi_2 + \varphi_3\big) \;+\; hb_{n-2}\big(\varphi_2+\varphi_3\big)$$
The two terms on the right are non-negative. We will prove by induction that $s_n \geq 0$. Assume that $s_n \geq 0$, then:
$$s_{n+1} = \mathrm{e}^{ah}\, s_n + hb_n\big(\varphi_1\,\mathrm{e}^{ah} + \varphi_2(3/2\,\mathrm{e}^{ah} - 2) + \varphi_3(\mathrm{e}^{ah} - 2)\big) + hb_{n-1}\big(\varphi_2/2 + \varphi_3\big).$$
In the right hand side, the first and last terms are non-negative. Thus the condition for s_{n+1} to be non-negative is that
$$\varphi_1\,\mathrm{e}^{ah} + \varphi_2(3/2\,\mathrm{e}^{ah} - 2) + \varphi_3(\mathrm{e}^{ah} - 2) \geq 0.$$
The study of that inequality gives $h \leq \beta_3/|a|$ and a numerical evaluation of β_3 is $\beta_3 \simeq 0.331$. With the initial condition $s_2 \geq 0$ we recorev (34). □

Proof of proposition 2. Consider the EAB_2 scheme. We define $u_n = y_n - K_1$. We have

$$\begin{aligned} u_{n+1} =& (y_n - K_1) + h\Big(\varphi_1\big(a(y_n - K_1) + (b_n + aK_1)\big) \\ &+ \varphi_2\big((b_n + aK_1) - (b_{n-1} + aK_1)\big)\Big) \\ =& u_n + h\Big(\varphi_1\big(au_n + \bar{b}_n\big) + \varphi_2\big(\bar{b}_n - \bar{b}_{n-1}\big)\Big), \end{aligned}$$

with $\bar{b}_j = b_j + aK_1$. Thus u_n is the numerical solution with the EAB_2 scheme of the equation $u' = au + \bar{b}(t,u)$ with $\bar{b}(t,u) = b(t,u) + aK_1$. By assumption $\bar{b}(t,u) \geq 0$ and the lemma 3 applies to u_n. It gives the lower bound in condition (31). The upper bound is similarly proven by considering $v_n = K_2 - y_n$. The proof is the same for the EAB_3 scheme. □

6 Numerical results

We present in this section numerical experiments that investigate the convergence, accuracy and stability properties of the I-EAB_k and EAB_k schemes. The membrane equation in cardiac electrophysiology is considered for two ionic models, the Beeler-Reuter (BR) and to the ten Tusscher *et al.* (TNNP) models. We refer to [2] and [34] for the definition of the models. The stiffness of these two models is due to the presence of different time scales ranging from 1 ms to 1 s, as depicted on figure 5. The stabilizer a_n always is a diagonal matrix in this section.

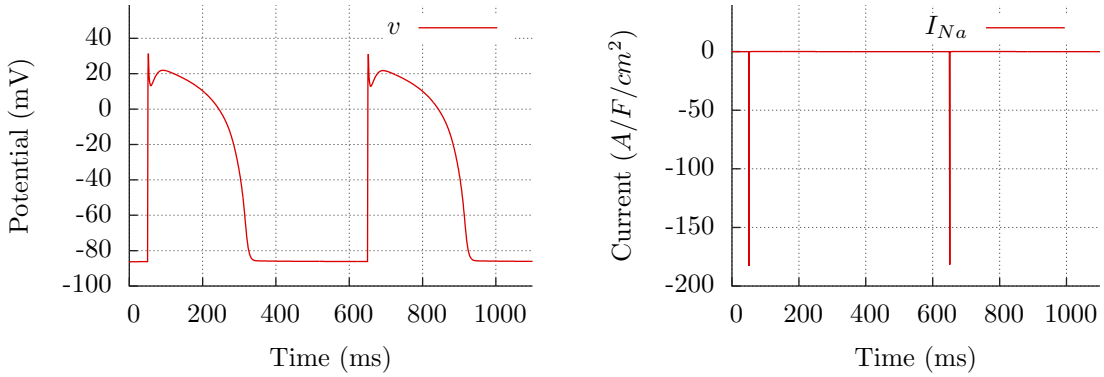

Figure 5: Two consecutive action potentials for the TNNP model: transmembrane potential v (left) and the fast sodium current I_{Na} (right), that is the main component of I_{ion} during the fast upstroke of the action potential.

The membrane equation has the general form, see [22, 2, 27, 34]:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \frac{w_{\infty,i}(v) - w_i}{\tau_i(v)}, \qquad \frac{\mathrm{d}c}{\mathrm{d}t} = q(w,c,v),, \qquad \frac{\mathrm{d}v}{\mathrm{d}t} = -I_{\mathrm{ion}}(w,c,v) + I_{\mathrm{st}}(t), \quad (35)$$

where $w = (w_1, \ldots, w_p) \in \mathbb{R}^p$ is the vector of the gating variables, $c \in \mathbb{R}^q$ is a vector of ionic concentrations or other state variables, and $v \in \mathbb{R}$ is the

cell membrane potential. These equations model the evolution of the transmembrane potential of a single cardiac cell. The four functions $w_{\infty,i}(v)$, $\tau_i(v)$, $q(w,c,v)$ and $I_{\text{ion}}(w,c,v)$ are given reaction terms. They characterize the cell model. The function $I_{\text{st}}(t)$ is a source term. It represents a stimulation current.
The formulation (3) is recovered with,

$$a(t,y) = \begin{pmatrix} -1/\tau(v) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad b(t,y) = \begin{pmatrix} w_\infty(v)/\tau(v) \\ q(y) \\ -I_{\text{ion}}(y) + I_{\text{st}}(t) \end{pmatrix},$$

for $y = (w,c,v) \in \mathbb{R}^N$ ($N = p+q+1$) and where $-1/\tau(v) = \text{diag}\left(-1/\tau_i(v)\right)_{i=1\dots p}$ and $w_\infty(v)/\tau(v) = \left(w_{\infty,i}(v)/\tau_i(v)\right)_{i=1\dots p}$.

6.1 Implementation and computational cost

The computation of y_{n+1} with the I-EAB$_k$ and EAB$_k$ schemes requires the data y_{n-i}, $a_{n-i} := a(t_{n-i}, y_{n-i})$ and $b_{n-i} := b(t_{n-i}, y_{n-i})$ for $i = 0 \dots k$.

EAB$_k$ practical implementation

Firstly, the $g_{ni} = b_{n-i} + (a_{n-i} - a_n)y_{n-i}$ are updated at each time step. Then the coefficients γ_{nj} in table 1 are computed. Secondly, the computation of y_{n+1} by formula (8) also requires the computation of the $\varphi_j(a_n h)\gamma_{nj}$. This is a matrix-vector product in general.
In the present case of a diagonal stabilizer, it becomes a scalar-scalar product per row. The $\varphi_j(a_n h)$ are computed on all diagonal entries of $a_n h$. This computation simply necessitates to compute $\varphi_0(a_n h) = \mathrm{e}^{a_n h}$ (one exponential per non zero diagonal entry) thanks to the recursion rule (9).
In general, the relation (9) can be used to replace the computation of the $\varphi_j(a_n h)\gamma_{nj}$ for $j = 0 \dots k$ by the computation of a single product $\varphi_k(a_n h)w_k$. Denoting by $w_1 = a_n y_n + b_n$ and $w_j = \gamma_{nj} + a_n h w_{j-1}$:

$$\begin{aligned} \text{EAB}_2 &: \quad y_{n+1} = y_n + h\left[w_1 + \varphi_2(a_n h)w_2\right], \\ \text{EAB}_3 &: \quad y_{n+1} = y_n + h\left[w_1 + w_2/2 + \varphi_3(a_n h)w_3\right], \\ \text{EAB}_4 &: \quad y_{n+1} = y_n + h\left[w_1 + w_2/2 + w_3/6 + \varphi_4(a_n h)w_4\right]. \end{aligned}$$

I-EAB$_k$ practical implementation

In addition, the I-EAB$_k$ method (10) requires a quadrature rule of sufficient order to preserve the scheme accuracy and convergence order. We used the Simpson quadrature rule for the cases $k = 2$, 3 and the three point Gaussian quadrature rule for k=4. We point out that a_n is assumed diagonal here so that the matrix exponentials below actually are scalar exponential.
The I-EAB$_k$ method with Simpson quadrature rule reads,

$$y_{n+1} = \mathrm{e}^{\tilde{g}_1}\left(y_n + b_n h/6\right) + \left(\tilde{b}_1 + 4\,\mathrm{e}^{\delta}\,\tilde{b}_{1/2}\right) h/6,$$

where (with the notations of section 1.2 $\tilde{g}_1 = \tilde{g}_n(t_{n+1})$, $\delta = \tilde{g}1 - \tilde{g}_n(t_n + h/2)$, $\tilde{b}_1 = \tilde{b}_n(t_{n+1})$ and $\tilde{b}_{1/2} = \tilde{b}_n(t_n + h/2)$. These coefficients are given for $k = 2$ by

$$\begin{aligned} \tilde{g}_1 &= (3a_n - a_{n-1})\, h/2, & \delta &= (7a_n - 3a_{n-1})\, h/8, \\ \tilde{b}_1 &= 2b_n - b_{n-1}, & \tilde{b}_{1/2} &= (3b_n - b_{n-1})/2, \end{aligned}$$

and for $k = 3$ by

$$\begin{aligned} \tilde{g}_1 &= (23a_n - 16a_{n-1} + 5a_{n-2})\, h/12, & \delta &= (29a_n - 25a_{n-1} + 8a_{n-2})\, h/24, \\ \tilde{b}_1 &= 3b_n - 3b_{n-1} + b_{n-2}, & \tilde{b}_{1/2} &= (15b_n - 10b_{n-1} + 3b_{n-2})\, /8, \end{aligned}$$

The I-EAB$_k$ method with the three point Gaussian quadrature rule reads,

$$y_{n+1} = \mathrm{e}^{\tilde{g}_1} \left(y_n + \frac{h}{18} \left(5\tilde{b}_l\, \mathrm{e}^{-\tilde{g}_l} + 8\tilde{b}_0\, \mathrm{e}^{-\tilde{g}_0} + 5\tilde{b}_r\, \mathrm{e}^{-\tilde{g}_r} \right) \right),$$

with $\tilde{b}_s = \tilde{b}_n(t_s)$, $\tilde{g}_s = \tilde{g}_n(t_s)$ for $s \in \{l, 0, r\}$ where $t_l = t_n + (1 - \sqrt{3/5}))h/2$, $t_0 = t_n + h/2$, $t_r = t_n + (1 + \sqrt{3/5}))h/2$ and with $\tilde{g}_1 = \tilde{g}_n(t_{n+1})$. These parameters are linear combination of the data a_{n-i}, b_{n-i} for $i = 0 \dots k-1$ with fixed coefficients. Formula for $k = 4$ follow. The parameters $\tilde{b}_s$ are given by:

$$\begin{aligned} 16\,\tilde{b}_0 =& 35\,b_n - 35\,b_{n-1} + 21\,b_{n-2} - 5\,b_{n-3} \\ 40\,\tilde{b}_r =& (95 + 179\sqrt{15}/15)b_n - (107 + 119\sqrt{15}/5)b_{n-1} \\ & + (69 + 79\sqrt{15}/5)b_{n-2} - (17 + 59\sqrt{15}/15)b_{n-3} \end{aligned}$$

and $\tilde{b}_l$ is the radical conjugate of $\tilde{b}_r$ (the radical conjugate of $x + \sqrt{y}$ is $x - \sqrt{y}$). Finally, the parameters $\tilde{g}_s$ definition is:

$$\begin{aligned} 24/h\ \tilde{g}_1 =& 55\,a_n - 59\,a_{n-1} + 37\,a_{n-2} - 9\,a_{n-3}, \\ 384/h\ \tilde{g}_0 =& 297\,a_n - 187\,a_{n-1} + 107\,a_{n-2} - 25\,a_{n-3}, \\ 200/h\ \tilde{g}_r =& (797/4 + 45\sqrt{15})a_n - (2233/12 + 47\sqrt{15})a_{n-1} \\ & + (1373/12 + 29\sqrt{15})a_{n-2} - (331/12 + 7\sqrt{15})a_{n-3}, \end{aligned}$$

and $\tilde{g}_l$ is the radical conjugate of $\tilde{g}_r$.

Computational cost

Consider an ODE system (1) whose numerical resolution cost is dominated by the computation of $(t, y) \mapsto f(t, y)$. This might be the case in general for *"large and complex models"*. For such problems explicit multistep methods are relevant since they will require one such operation per time step. In contrast, implicit methods, associated to a non linear solver, may necessitate a lot of

these operations, especially for large time steps when convergence is harder to reach.

In addition, the I-EAB$_k$ and EAB$_k$ schemes need several specific operations. In the case of a diagonal function $a(t, y)$ they have been previously described: the EAB$_k$ require one scalar exponential computation per non zero row of $a(t, y)$, the I-EAB$_k$ with Simpson rule needs twice more and the I-EAB$_3$ with 3 point Gaussian quadrature rule four times more. Such a cost is not negligible, but is at worst of same order than computing $(t, y) \mapsto f(t, y)$ for complex models. For the TNNP model considered here, computing $(t, y) \mapsto f(t, y)$ costs 50 scalar exponentials whereas the EAB$_k$ implementation adds 7 supplementary scalar exponentials per time step. In terms of cost per time step, the EAB$_k$ method is rather optimal. The relationship between accuracy and cost of the EAB$_k$ method has been investigated in [12]: more details are available in section 6.4.

6.2 Convergence

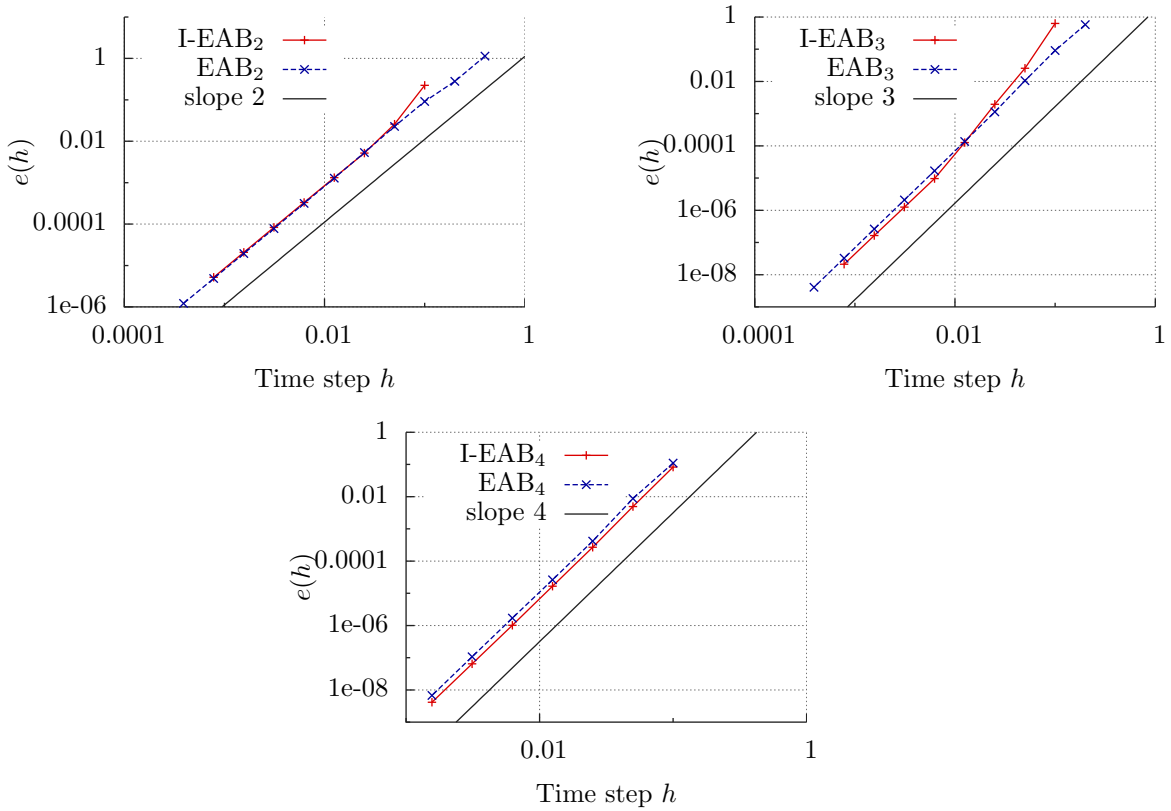

Figure 6: Relative L^∞ error $e(h)$ for the I-EAB$_k$ and EAB$_k$ schemes, $k = 2$, 3 and 4, and for the BR model.

For the chosen application, no theoretical solution is available. Convergence properties are studied by computing a reference solution y_{ref} for a reference time step h_{ref} with the Runge Kutta 4 scheme. Numerical solutions y are

computed to y_{ref} for coarsest time steps $h = 2^p h_{ref}$ for increasing p. Any numerical solution y consists in successive values y_n at the time instants $t_n = nh$. On every interval (t_{3n}, t_{3n+3}) the polynomial $\overline{y}$ of degree at most 3 so that $\overline{y}(t_{3n+i}) = y_{3n+i}$, $i = 0 \dots 4$ is constructed. On $(0, T)$, $\overline{y}$ is a piecewise continuous polynomial of degree 3. Its values at the reference time instants nh_{ref} are computed. This provides a projection $P(y)$ of the numerical solution y on the reference grid. Then $P(y)$ can be compared with the reference solution y_{ref}. The numerical error is defined by

$$e(h) = \frac{\max |v_{\text{ref}} - P(v)|}{\max |v_{\text{ref}}|}, \tag{36}$$

where the potential v is the last and stiffest component of y in equation 35.
The convergence graphs for the BR model are plotted on figure 6. All the schemes display the expected asymptotic behavior $e(h) = O(h^k)$ as $h \to 0$, as proved in theorem 1.

6.3 Stability

The stiffness of the BR and TNNP models along one cellular electrical cycle (as depicted on figure 5 has been evaluated in [33]. The largest negative real part of the eigenvalues of the Jacobian matrix during this cycle is of -1170 and -82 for the TNNP and BR models respectively. This means that the TNNP model is 15 times stiffer than the BR model ($15 \simeq 1170/82$).
We want to evaluate the impact of this increase of stiffness in terms of stability for the EAB_k and I-EAB_k schemes and to provide a comparison with some other classical time stepping methods. To this aim we consider the *critical time step* Δt_0. It is defined as the largest time step such that the numerical simulation runs without overflow nor non linear solver failure for $h < \Delta t_0$. The numerical evaluation of Δt_0 is easy for explicit methods. For implicit methods, the choice of the non linear solver certainly impacts Δt_0. Without considering more deeply this problem, we just carefully set up the non linear solver, so as to provide the largest Δt_0. In practice, we have been using a Jacobian free Krylov Newton method.

Results are on table 2. The Adams-Bashforth (AB_k) and the backward differentiation (BDF_k) methods of order k have been considered, together with the RK_4 scheme.
The AB_k and the RK_4 schemes have bounded stability domain (see [15, p. 243]). Then it is expected for the critical time step to be divided by a factor close to 15 between the BR and TNNP models. Results presented in table 2 show this behavior.
The BDF_2 scheme is A-stable whereas the BDF_3 and BDF_4 are $A(\alpha)$-stable with large angle α (see [15, p. 246]). Hence the critical time step is expected to remain unchanged between the two models. Table 2 shows that the Δt_0 actually are divided by approximately 2.
The critical time steps for the I-EAB_k and EAB_k models are presented in table

Table 2: Critical time step Δt_0

(a) Classical methods

	BR	TNNP
AB_2	0.124×10^{-1}	0.850×10^{-3}
BDF_2	0.306	0.158
AB_3	0.679×10^{-2}	0.464×10^{-3}
BDF_3	0.362	0.181
AB_4	0.372×10^{-2}	0.255×10^{-3}
RK_4	0.338×10^{-1}	0.255×10^{-2}
BDF_4	0.423	0.201

(b) I-EAB_k and EAB_k exponential methods

	BR	TNNP
I-EAB_2	0.121	0.103
EAB_2	0.424	0.233
I-EAB_3	0.103	0.123
EAB_3	0.203	0.108
I-EAB_4	0.133	0.106
EAB_4	0.122	0.756×10^{-1}

2. The critical time steps for the I-EAB_k schemes remain almost unchanged from the BR to the TNNP model. For the EAB_k, they are divided by approximately 2, which behavior is similar as for the BDF_k method.
As a conclusion, for the present application, the EAB_k and I-EAB_k methods are as robust to stiffness than the implicit BDF_k schemes, though being explicit. As a matter of fact, section 4 shows that the stability domains for the I-EAB_k and EAB_k schemes depend on the discrepancy between the complete Jacobian matrix and $a(t, y)$. In the present case, $a(t, y)$ only contains a part of the Jacobian matrix diagonal. It is very interesting to notice that robustness to stiffness is actually achieved with this choice. It is finally also interesting to see that the critical time steps of implicit and exponential methods are of the same order.

6.4 Accuracy

In terms of accuracy, the schemes can be compared using the relative error $e(h)$ in equation 36. The EAB_k and I-EAB_k schemes can be compared with the AB_k methods only at very small time steps, because of the lack of stability of AB_k schemes (see table 2). In table 3 are given the accuracies of these methods for a given time step $h = 10^{-3}$ and for the BR model. Ii is observed that the same level of accuracy is obtained with AB_k and EAB_k at fixed k. These figures illustrate that inside the asymptotic convergence region, the EAB_k, I-EAB_k and AB_k schemes are equivalent in terms of accuracy.

Comparison at large time steps between the EAB_k and BDF_k for the TNNP model is shown in table 4. These figures show that for large time steps BDF_k is more accurate than EAB_k. A gain in accuracy of factor 2.5, 5 and 6 is observed for $h = 0.05$ and for k=2, 3 and 4 respectively. However, compare row 3 for EAB_k ($h = 0.025$) with row 2 for BDF_k ($h = 0.05$). It shows that the numerical solutions with an accuracy close to 0.01 are obtained when dividing the time step by (less than) 2 between BDF_k and EAB_k. Meanwhile, EAB_k with $h = 0.025$ costs less than BDF_k with $h = 0.05$, as developed in section

Table 3: Accuracy $e(h)$ for the AB_k, I-EAB_k and EAB_k schemes: using the BR model and fixed time step $h = 10^{-3}$

	$k=2$	$k=3$	$k=4$
AB_k	5.32×10^{-6}	4.33×10^{-8}	8.69×10^{-10}
I-EAB_k	8.55×10^{-6}	4.44×10^{-8}	7.30×10^{-10}
EAB_k	7.90×10^{-6}	7.00×10^{-8}	1.16×10^{-9}

Table 4: Accuracy for the TNNP model

(a) EAB_k

h	$k=2$	$k=3$	$k=4$
0.1	0.351	0.530	
0.05	9.01×10^{-2}	5.59×10^{-2}	8.93×10^{-2}
0.025	2.14×10^{-2}	7.34×10^{-3}	8.34×10^{-3}

(b) BDF_k

h	$k=2$	$k=3$	$k=4$
0.1			0.129
0.05	3.57×10^{-2}	1.15×10^{-2}	1.44×10^{-2}
0.025	1.10×10^{-2}	2.58×10^{-3}	2.38×10^{-3}

6.1.

We conclude that EAB_k schemes provide a cheaper way to compute numerical solutions at large time step for a given accuracy. The same conclusion also holds for the BR model, see table 5. A deeper analysis of the relationship between accuracy and computational cost for the EAB_k scheme as compared to other methods is available in [12] with the same conclusion.

Table 5: Accuracy for the BR model

(a) EAB_k

h	$k=2$	$k=3$	$k=4$
0.2	0.284	0.516	
0.1	9.26×10^{-2}	9.17×10^{-2}	0.119
0.05	8.20×10^{-2}	1.09×10^{-2}	8.96×10^{-3}

(b) BDF_k

h	$k=2$	$k=3$	$k=4$
0.2	9.74×10^{-2}	4.09×10^{-2}	4.98×10^{-2}
0.1	3.44×10^{-2}	1.04×10^{-2}	1.27×10^{-2}
0.05	9.74×10^{-3}	2.29×10^{-3}	2.02×10^{-3}

In table 5 are given the accuracies at large time step now considering the BR model. Comparison with table 4 shows that accuracy is preserved by dividing h by 2 when switching from the BR to the TNNP model. As already said, the TNNP model is 15 times stiffer than the BR model.

We conclude that the EAB_k schemes also exhibit a large robustness to stiffness in terms of accuracy. This robustness is equivalent as for the implicit BDF_k schemes. This is remarkable for an explicit scheme, as for the robustness to stiffness in terms of critical time step discussed in the previous subsection.

References

[1] W. Auzinger and M. Lapińska. Convergence of rational multistep methods of Adams-Padé type. Technical report, ASC Report, Vienna University of Technology, 2011.

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

[3] C. Börgers and A. R. Nectow. Exponential time differencing for Hodgkin-Huxley-like ODEs. *SIAM J. Sci. Comput.*, 35(3):B623–B643, 2013.

[4] M. P. Calvo and C. Palencia. A class of explicit multistep exponential integrators for semilinear problems. *Numer. Math.*, 102(3):367–381, 2006.

[5] J. Certaine. The solution of ordinary differential equations with large time constants. In A. Ralston and H. S. Wilf, editors, *Mathematical Methods for Digital Computers*, pages 128–132. John Wiley & Sons, 1960.

[6] M. Chu. An automatic multistep method for solving stiff initial value problems. *J. Comput. Appl. Math.*, 9(3):229–238, 1983.

[7] J. Clements, J. Nenonen, P. Li, and B. Horacek. Activation dynamics in anisotropic cardiac tissue via decoupling. *Ann. Biomed. Eng.*, 32(7):984–990, 2004.

[8] P. Colli-Franzone, L. Pavarino, and B. Taccardi. Monodomain simulations of excitation and recovery in cardiac blocks with intramural heterogeneity. In *Functional Imaging and Modeling of the Heart*, volume 3504 of *Theoretical Computer Science and General Issues*, pages 267–277, 2005.

[9] P. Colli-Franzone, L. Pavarino, and B. Taccardi. Simulating patterns of excitation, repolarization and action potential duration with cardiac bidomain and monodomain models. *Mathematical Biosciences*, 197(1):35 – 66, 2005.

[10] Y. Coudiere, C. Douanla-Lontsi, and C. Pierre. Rush Larsen time stepping methods of high order for stiff problems in cardiac electrophysiology. *HAL Preprint*, 2015.

[11] S. Cox and P. Matthews. Exponential time differencing for stiff systems. *J. Comput. Phys.*, 176(2):430–455, 2002.

[12] C. Douanla-Lontsi, Y. Coudière, and C. Pierre. Efficient high order schemes for stiff ODEs in cardiac electrophysiology. In M. Lo, E. Badouel, and N. Gmati, editors, *Proceedings of CARI 2016*, pages 312–319, Hammamet, Tunisia, 2016.

[13] S. Dragomir. *Some Gronwall type inequalities and applications*. Nova Science Publishers, Inc., Hauppauge, New York, 2003.

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

[15] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*, volume 14 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 1996.

[16] M. Hochbruck. A short course on exponential integrators. In Z. Bai, W. Gao, and Y. Su, editors, *Matrix Functions and Matrix Equations*, volume 19 of *Contemp. Appl. Math.*, pages 28–49. Higher Ed. Press, Beijing, 2015.

[17] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19(5):1552–1574, 1998.

[18] M. Hochbruck and A. Ostermann. Explicit exponential Runge-Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.*, 43(3):1069–1090, 2005.

[19] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.

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

[21] M. Hochbruck, A. Ostermann, and J. Schweitzer. Exponential Rosenbrock-type methods. *SIAM J. Numer. Anal.*, 47(1):786–803, 2009.

[22] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117:500–544, 1952.

[23] V. Iyer, R. Mazhari, and R. L. Winslow. A computational model of the human left-ventricular epicardial myocyte. *Biophys. J.*, 87(3):1507–1525, 2004.

[24] A. Koskela and A. Ostermann. Exponential Taylor methods: analysis and implementation. *Comput. & Math. with Appl.*, 65(3):487–499, 2013.

[25] D. Lee and S. Preiser. A class of non linear multistep A-stable numerical methods for solving stiff differential equations. *Comput. & Math with Appl.*, 4:43–51, 1978.

[26] V. Luan and A. Ostermann. Explicit exponential Runge-Kutta methods of high order for parabolic problems. *J. Comput. Appl. Math.*, 256:168–179, 2014.

[27] C. H. Luo and Y. Rudy. A dynamic model of the cardiac ventricular action potential. I. Simulations of ionic currents and concentration changes. *Circ. Res.*, 74(6):1071–1096, 1994.

[28] B. Minchev and W. Wright. A review of exponential integrators for first order semi-linear problems. Preprint Numerics 2/2005, Norges Teknisk-Naturvitenskapelige Universitet, 2005.

[29] 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, June 23–27, 1969*, pages 214–219. Springer, Berlin, Heidelberg, 1969.

[30] A. Ostermann, M. Thalhammer, and W. Wright. A class of explicit exponential general linear methods. *BIT*, 46(2):409–431, 2006.

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

[32] G. Rainwater and M. Tokman. A new class of split exponential propagation iterative methods of Runge-Kutta type (sEPIRK) for semilinear systems of ODEs. *J. Comput. Phys.*, 269:40–60, 2014.

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

[34] 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.

[35] M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *J. Comput. Phys.*, 213(2):748–776, 2006.

[36] M. Tokman, J. Loffeld, and P. Tranquilli. New adaptive exponential propagation iterative methods of Runge-Kutta type. *SIAM J. Sci. Comput.*, 34(5):A2650–A2669, 2012.

[37] P. Tranquilli and A. Sandu. Rosenbrock-Krylov methods for large systems of differential equations. *SIAM J. Sci. Comput.*, 36(3):A1313–A1338, 2014.