Numer Algor (2008) 48:327–346
DOI 10.1007/s11075-008-9202-y

ORIGINAL PAPER

Exponential fitted Gauss, Radau and Lobatto methods of low order

J. Martín-Vaquero · J. Vigo-Aguiar

Received: 19 September 2007 / Accepted: 27 March 2008 /
Published online: 23 May 2008

Abstract Several exponential fitting Runge-Kutta methods of collocation type are derived as a generalization of the Gauss, Radau and Lobatto traditional methods of two steps. The new methods are capable of the exact integration (with only round-off errors) of differential equations whose solutions are linear combinations of an exponential and ordinary polynomials. Theorems of the truncation error reveal the good behavior of the new methods for stiff problems. Plots of their absolute stability regions that include the whole of the negative real axis are provided. A different procedure to find the parameter of the method is proposed. The variable step Radau method of two stages is derived. Finally, numerical examples underscore the efficiency of the proposed codes, especially when they are integrating stiff problems.

Keywords Runge-Kutta methods · Collocation type · Exponential fitting · Stiff problems

Mathematics Subject Classifications (2000) 34A45 · 65L06

1 Introduction

The numerical integration of ordinary differential equations (ODEs) has been one of the principal concerns of numerical analysis. In the early 1950s, after the

J. Martín-Vaquero (✉)
ETS Ingenieros industriales, Universidad de Salamanca, Bejar, Spain
e-mail: jesmarva@usal.es

J. Vigo-Aguiar
Facultad de Ciencias, Universidad de Salamanca, Salamanca, Spain
e-mail: jvigo@usal.es

pioneering work of Curtiss and Hirschfelder [1], it was realized that there was an important class of ODEs which presented a severe challenge to numerical methods available at that time. These problems have become known as stiff systems. Stiff problems (and highly oscillatory problems) are very common problems in many fields of applied sciences (see [2], for example): atmosphere (atmospheric phenomena involving transport with chemical reactions), biology, the combustion theory, control theory, dynamics of missile guidance, dispersed phases, electronic circuit theory, fluids, heat transfer, chemical kinetics, lasers, mechanics, molecular dynamics, nuclear, process industries, process vessels, reactor kinetics...

Although there has been much controversy about the mathematical definition (see [3]), and in fact, there is no good mathematical definition of the concept of stiffness, we can say that a problem

$$y'(x) = f(x, y(x)), \qquad y(x_0) = y_0\,, \tag{1}$$

(where $y = [y^1, \ldots, y^m]$, and $f = [f^1, \ldots, f^m]$, $y_0 = \left[y_0^1, \ldots, y_0^m\right]$, $x \in R$) is stiff if its Jacobian (in a neighborhood of the solution) has eigenvalues λ_i that verify $\frac{\max|Re\lambda_i|}{\min|Re\lambda_i|} \gg 1$ (usually, it is considered that $\max Re\,\lambda_i < 0$). Stiff systems are considered difficult because explicit numerical methods designed for non-stiff problems are forced to use very small step sizes increasing in this way the computational work. Looking for better methods for solving these systems, Curtiss and Hirschfelder [1] discovered the Backward Differentiation Formulae (BDF). Since then, a great effort has been made in order to obtain new numerical integration methods with strong stability properties desirable for solving stiff systems. For a survey on stiffness of ODE's see [4, 5] or [6].

A great number of schemes based on modifications of the classical BDF formulae have appeared. Among them, we may mention DIFSUB [7] or LSODE [8], VODE [9], which uses the so-called Fixed Leading Coefficient BDF methods, DASSL [10], which is also indicated for solving differential algebraic equations, MEBDF (see [11]), which considers two predicted values to compute a new corrected approximation to the solution using a modified multistep formula, A-BDF [12], which is a one-parameter family that is a generalization of the classical BDF codes, and exponential fitting BDF schemes (EF-BDF) as in [13–15] or [16].

Implicit Runge-Kutta methods are another kind of formulae very common with stiff problems. Radau [4], STRIDE [17] or [18], DIRK [17], SDIRK, Gauss, Lobatto, Rosenbrock, modified schemes [19–22], ..., have frequently been used with those kind of numerical problems.

Another kind of schemes has appeared with good results. Such methods are called exponential fitting and some examples could be [23] (in that paper exponential fitting methods are applied for the first time to stiff problems) or [24, 25].

In this paper, we are going to derive exponential fitting Runge-Kutta methods of collocation type through the Gauss, Radau and Lobatto traditional integrators. This is, we will impose both kind of conditions: the exact integration of differential equations whose solutions are linear combinations of an exponential with parameter λ and ordinary polynomials and the order conditions imposed to the traditional Runge-Kutta methods.

The paper is organized as follows. In Section 2, we construct several exponential fitted versions of the well-known classical collocation methods. In Section 3, an analysis of the convergence of these new methods is made. In Section 4, we derived the variable step Radau method of two stages. Finally, in Section 5 we show, with different test numerical examples, the efficiency of the proposed codes, especially when they are integrating stiff problems.

2 Derivation of the methods

Let us consider, first, the scalar (IVP) initial-value problem of the form

$$y'(x) = f(x, y(x)), \quad x \in [x_0, x_f], \quad y(x_0) = y_0 \tag{2}$$

and assume that the function $f : [x_0, x_f] \times R \to R$ satisfies all the necessary requirements for the existence of a unique solution.

For the description of EFRK methods we use the classical Butcher notation [6]

$$\begin{aligned} y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i f(x_n + c_i h, u_i), \\ u_i &= y_n + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, u_j), \end{aligned} \tag{3}$$

with $i = 1, \ldots, s$ and the coefficients are displayed as a Butcher array:

$$\begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ c_2 & a_{21} & \ldots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array}$$

Then, we will fix c_i the values of the classical Runge-Kutta methods, but, now, the new coefficients a_{ij}, b_i are those such that

$$y(x_n + c_i h) = y(x_n) + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, y(x_n + c_j h)), \tag{4}$$

$$y(x_n + h) = y(x_n) + h \sum_{i=1}^{s} b_i f(x_n + c_i h, y(x_n + c_i h) \tag{5}$$

when $y(x)$ belongs to the space $< 1, x, \ldots, x^{s-1}, e^{\lambda x} >$.

Case A Derivation of the new exponential fitting Gauss method of 2-stages.

The weights c_i of the new Gauss method are the same as in the traditional method: $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$. But, a_{ij}, b_i are those such that (4) and (5) are satisfied when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$.

In this way the new exponential fitting method can be written as

$$\begin{array}{c|cc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & -\frac{6-6e^{\left(-3+\sqrt{3}\right)\widehat{\lambda}/6}+\left(-3+\sqrt{3}\right)\widehat{\lambda}e^{\sqrt{3}\widehat{\lambda}/3}}{6\widehat{\lambda}\left(-1+e^{\sqrt{3}\widehat{\lambda}/3}\right)} & c_1 - a_{11} \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{e^{\left(-3+\sqrt{3}\right)\widehat{\lambda}/6}\left(6+e^{\left(3+\sqrt{3}\right)\widehat{\lambda}/6}\left(-6+\left(3+\sqrt{3}\right)\widehat{\lambda}\right)\right)}{6\widehat{\lambda}\left(-1+e^{\sqrt{3}\widehat{\lambda}/3}\right)} & c_2 - a_{21} \\
\hline
 & \frac{e^{\left(-3+\sqrt{3}\right)\widehat{\lambda}/6}\left(1-e^{\widehat{\lambda}}+e^{\left(3+\sqrt{3}\right)\widehat{\lambda}/6}\widehat{\lambda}\right)}{\widehat{\lambda}\left(-1+e^{\sqrt{3}\widehat{\lambda}/3}\right)} & 1 - b_1
\end{array}$$

$\widehat{\lambda}$ being the parameter λh in the method.

Case B Derivation of the new exponential fitting RadauIIA method of 2-stages.

The weights c_i of the new RadauIIA method are, then, the same as in the traditional method: $c_1 = \frac{1}{3}$ and $c_2 = 1$. The $a_{i,j}$ are those such that (4) hold when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$. And, in this case, $b_1 = a_{21}$, $b_2 = a_{22}$

So, in this way the new exponential fitting method can be written as

$$\begin{array}{c|cc}
\frac{1}{3} & \frac{3-3e^{-\widehat{\lambda}/3}-\widehat{\lambda}e^{2\widehat{\lambda}/3}}{3\widehat{\lambda}-3\widehat{\lambda}e^{2\widehat{\lambda}/3}} & \frac{1}{3} - a_{11} \\
1 & \frac{e^{-\widehat{\lambda}/3}\left(1+e^{\widehat{\lambda}}\left(-1+\widehat{\lambda}\right)\right)}{\widehat{\lambda}\left(-1+e^{2\widehat{\lambda}/3}\right)} & 1 - a_{21} \\
\hline
 & a_{21} & a_{22}
\end{array}$$

Case C Derivation of the new exponential fitting LobattoIIIA method of 2-stages.

The weights c_i of the new LobattoIIIA method are $c_1 = 0$ and $c_2 = 1$. The $a_{i,j}$ are those such that (4) hold when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$. And, as with RadauIIA, $b_1 = a_{21}$, $b_2 = a_{22}$.

So, in this way the new exponential fitting method can be written as

$$\begin{array}{c|cc}
0 & 0 & 0 \\
1 & \frac{1+e^{\widehat{\lambda}}\left(-1+\widehat{\lambda}\right)}{\widehat{\lambda}\left(-1+e^{\widehat{\lambda}}\right)} & 1 - a_{21} \\
\hline
 & a_{21} & a_{22}
\end{array}$$

Note: we have two options when we formulate the methods for vectorial examples.

a) We can formulate the methods using matrices instead scalars in the coefficients. We only have to change λ by a matrix A, 1 by the identity matrix and we need to consider $\frac{B}{C} = BC^{-1}$. In that case, (4) and (5) are satisfied when $y(x) = (c_1, \ldots, c_m)^T$, $y(x) = x\left(c_1', \ldots, c_m'\right)^T$, $y(x) = e^{Ax} \cdot$

$\left(c''_1, \ldots, c''_m\right)^T$, where $c_i, c'_i, c''_i \in R$. Additionally, the eigenvalues of the matrices should have a negative real part.

b) One can avoid the calculus of the exponential of the matrix taking one value λ at each step that contains the principal information of the problem. We have chosen this strategy in this paper.

3 Convergence of the exponential fitting Runge-Kutta methods

In this section we will study the consistency and stability properties of the new methods. Since they can be written as Runge-Kutta algorithms they are zero-stable and we only need to study consistency and absolute stability of these formulas.

3.1 Consistency of the exponential fitting BDF-Runge-Kutta methods

If we want to know the local truncation error of the methods in a classical way, we need to consider the Runge-Kutta algorithms and study the order conditions following the theory of elementary differentials (the Fréchet derivatives) and rooted trees (see [6] chapter five, for example).

The local truncation error of a Runge-Kutta method with constant coefficients is given by (formula (5.47) in [6])

$$LTE = \frac{h^{p+1}}{(p+1)!} \sum_{r(t)=p+1} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) + O\left(h^{p+2}\right) \tag{6}$$

$\alpha(t)$ being the number of essentially different ways of labelling the nodes of the tree t with the integers $1, 2, \ldots, r(t)$. An easy way of computing $\alpha(t)$ is

$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)},$$

where the order $r(t)$, symmetry $\sigma(t)$ and density $\gamma(t)$ of a tree t are defined as in [6], p. 164.

The function F is defined on the set T of all trees as in (5.39) in [6] (see Table 5.2 and the definition of the Mth. Fréchet derivative, p. 158, too) and the relations between $y^{(q)}$ and all elementary differentials of order q is the following theorem (see Butcher [26]):

Theorem 1 *Let $y' = f(y)$, f: $R^m \to R^m$. Then*

$$y^{(q)} = \sum_{r(t)=q} \alpha(t)F(t).$$

Finally $\psi(t)$ depends on the elements of the Butcher array as in [6], *p.* 167: *for $i = 1, 2, \ldots, s, s+1$ define on the set T of all trees the functions ψ_i by*

$$\psi_i(\tau) = \sum_{j=1}^{s} a_{ij}$$

$$\psi_i([t_1 t_2 \ldots t_M]) = \sum_{j=1}^{s} a_{ij}\psi_j(t_1)\psi_j(t_2)\ldots\psi_j(t_M),$$

then, $\psi(t) := \psi_{s+1}(t)$.

Now, we can study the local truncation error of the exponential fitting Runge-Kutta methods in a similar way to [27].

Theorem 2 *The leading term of the local truncation error of the new exponential fitting Gauss-2s is*

$$\frac{h^5(C_{H24} + C_{H5})}{4320},$$

where

$$C_{H24} = -\lambda^3 y'' + 5\lambda^2 f_y^2 y' + 10\lambda\left(f_{yy} f f_y - f_y^3\right) y',$$

$$C_{H5} = \left(f_{yyyy} f^3 + 2 f_{yyy} f_y f^2 - 6 f_{yy}^2 f^2 + 4 f_{yy} f_y^2 f + 6 f_y^4\right) y'.$$

Proof 2

i) The only condition of a Runge-Kutta method to be consistent (at least) is $\sum_{i=1}^{s} b_i = 1 + O(\widehat{\lambda})$ (again $\widehat{\lambda}$ is the parameter λh in the method). In this case $\sum_{i=1}^{s} b_i = 1$.

ii) One method with order higher than one has to verify $2\sum_{i=1}^{s} b_i c_i = 1$, in this case

$$2\sum_{i=1}^{2} b_i c_i = -\frac{2\sqrt{3}e^{(-3+\sqrt{3})\widehat{\lambda}/6} - 2\sqrt{3}e^{(3+\sqrt{3})\widehat{\lambda}/6} + (3+\sqrt{3})\widehat{\lambda} + (-3+\sqrt{3})e^{\sqrt{3}\widehat{\lambda}/3}}{3\widehat{\lambda}\left(-1 + e^{\sqrt{3}\widehat{\lambda}/3}\right)},$$

whose Taylor series are $1 + \frac{\widehat{\lambda}^3}{2160} + O(h^5)$. Then, one part of the local truncation error is $\frac{h^2}{2} y^{(2)}(x_n)(-\frac{\widehat{\lambda}^3}{2160} + O(h^5))$.

iii) The two conditions of a third-order method are

$$3\sum_{i=1}^{s} b_i c_i^2 = 1 + O(h)$$

and

$$6 \sum_{i=1,\, j=1}^{s} b_i a_{ij} c_j = 1 + O(h).$$

In this case

$$3 \sum_{i=1}^{s} b_i c_i^2 = 1 + \frac{\widehat{\lambda}^3}{1440} + O\left(h^5\right)$$

and

$$6 \sum_{i=1,\, j=1}^{s} b_i a_{ij} c_j = 1 - \frac{\widehat{\lambda}^2}{144} + \frac{\widehat{\lambda}^3}{720} + O\left(h^4\right).$$

Then,

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{\widehat{\lambda}^2}{144} f_y^2 f + O\left(h^6\right),$$

we are considering the scalar problem and the notation as in [6].

iv) We study the four conditions to be a fourth-order method and we got that

$$\frac{h^4}{4!} \sum_{r(t)=4} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^4}{4!} \left(3\frac{\widehat{\lambda}}{54} f_{yy} f f_y f - \frac{\widehat{\lambda}}{18} f_y^3 f\right) + O\left(h^6\right).$$

v) Finally, when we studied the conditions to be a fifth-order method, we got that they are not satisfied

$$\frac{h^5}{5!} \sum_{r(t)=5} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^5}{5!} C_5 + O\left(h^6\right)$$

where

$$C_5 = \frac{1}{36}\left(f_{yyyy} f^4 + 6 f_{yyy} f_y f^3\right) - \frac{1}{24}\left(4 f_{yy}^2 f^3 + 4 f_{yy} f_y^2 f^2\right) -$$

$$-\frac{1}{9}\left(f_{yyy} f_y f^3 + 3 f_{yy} f_y^2 f^2\right) + \frac{1}{6}\left(f_{yy} f_y^2 f^2 + f_y^4 f\right).$$

If we add the leading terms of local truncation error that we got in i) to v) and simplify we get the total expression of the local truncation error. □

Theorem 3 *The leading term of the local truncation error of the new exponential fitting Radau-2s is*

$$\frac{h^4\left(\lambda^2 y'' + \left(-4\lambda f_y^2 - f_{yyy} f^2 + f_{yy} f f_y + 3\left(f_y f_{yy} f + f_y^3\right)\right) y'\right)}{216}.$$

Fig. 1 Absolute stability regions (in grey) of the exponential fitting Gauss-2s and Radau-2s methods. The parameters in the method are real. Horizontal and vertical axes represent $Re(\mu h)$ and $Im(\mu h)$

Proof 3

i) The condition to be a consistent method is satisfied since $\sum_{i=1}^{s} b_i = 1$.
ii) The method has at least first-order because $2\sum_{i=1}^{s} b_i c_i = 1 - \frac{\widehat{\lambda}^2}{108} + O(h^3)$. Then, one part of the local truncation error is $\frac{h^2}{2} y^{(2)}(x_n) \left(\frac{\widehat{\lambda}^2}{108} + O(h^3) \right)$.
iii) The method has order higher than two because

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{-\widehat{\lambda}}{9} f_y^2 f + O\left(h^5\right).$$

iv) We studied the four conditions to be a forth-order method but they are not satisfied

$$\frac{h^4}{4!} \sum_{r(t)=4} \alpha(t)[1 - \gamma(t)\psi(t)]F(t)$$
$$= \frac{h^4}{4!} \left(\frac{-1}{9} \left(f_{yyy} f^3 + f_{yy} f f_y f \right) + \frac{1}{3} \left(f_y f_{yy} f^2 + f_y^3 f \right) \right) + O\left(h^5\right).$$

□

Theorem 4 *The leading term of the local truncation error of the new exponential fitting Lobatto-2s is*

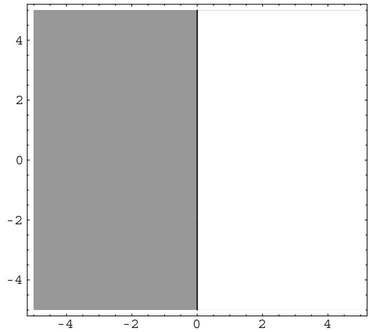
$$\frac{h^3(\lambda y'' - y''')}{12}.$$

Proof 4

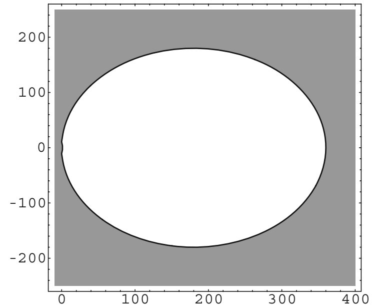
i) The condition to be a consistent method is satisfied since $\sum_{i=1}^{s} b_i = 1$.
ii) While in the condition of a first-order method, we got $2\sum_{i=1}^{s} b_i c_i = 1 - \frac{\widehat{\lambda}}{6} + O(h^3)$. Then, one part of the local truncation error is $\frac{h^2}{2} y^{(2)}(x_n) \left(\frac{\widehat{\lambda}}{6} + O(h^3) \right)$.
iii) And we get that the method has order two because

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{-1}{2} \left(f_{yy} f^2 + f_y^2 f \right) + O\left(h^5\right).$$

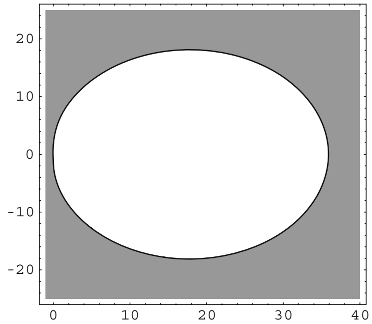
□



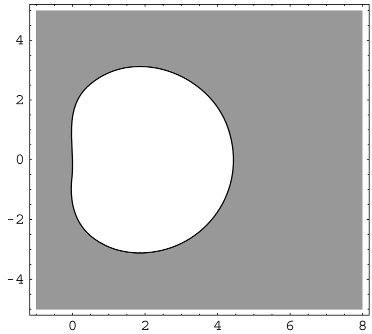
a $\lambda h \to 0$. *EF-Gauss-2s method.*



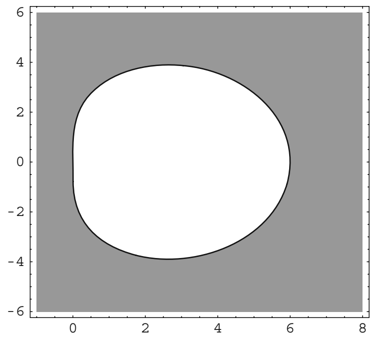
b λh =–0.1. *EF-Gauss-2s method.*



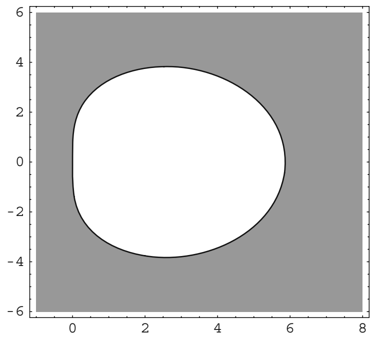
c λh=–1. *EF-Gauss-2s method.*



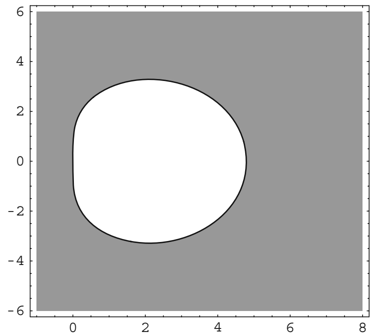
d λh= –7. *EF-Gauss-2s method.*



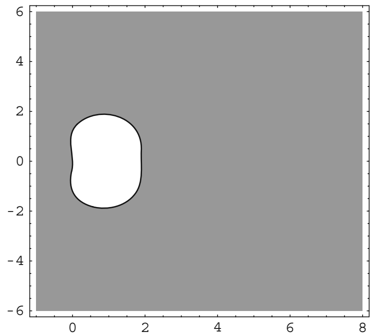
e $\lambda h \to 0$. *EF-Radau-2s method.*



f λh= –0.1. *EF-Radau-2s method.*



g λh = –1. *EF-Radau-2s method.*



h λh =–5. *EF-Radau-2s method.*

Fig. 2 Absolute stability regions (in *grey*) of the exponential fitting Radau-2s and Lobatto-2s methods. The parameters in the method are complex. *Horizontal* and *vertical axes* represent $Re(\mu h)$ and $Im(\mu h)$

3.2 Absolute stability of the exponential fitting Runge-Kutta methods

The classical definitions of absolute stability regions and A-stability were stated for linear multistep methods with constant coefficients. The stability properties of the proposed methods are analyzed to demonstrate their relevance especially in the integration of stiff oscillatory problems. In this section the definitions are extended to exponential fitting methods. The way is very similar to that used in [28] to extend those definitions.

In [28] Coleman and Ixaru studied the stability properties of existing exponential fitting methods that integrate exactly the problem

$$y''(x) = g(x, y(x)), \tag{7}$$

when $y(x) = exp(\pm ikx)$, but when they want to study their stability properties, they apply the method to the test equation

$$y''(x) = -w^2 y(x), \tag{8}$$

and then, they plot their regions of stability on the $\mu - \theta$ plane (being $\mu = wh$ and $\theta = kh$).

In our case, we shall consider, first, the test problem

$$y'(x) = Ay(x), \tag{9}$$

while we introduce in the method the estimated parameter A^*, so the weights are $b_i(A^*h)$ and $a_{ij}(A^*h)$.

We are going to use this analysis considering that there exist a nonsingular matrix Q such that

$$Q^{-1}AQ = \Lambda = diag[\lambda_1, ..., \lambda_m],$$

and

$$Q^{-1}A^*Q = \Lambda^* = diag[\lambda_1^*, ..., \lambda_m^*].$$

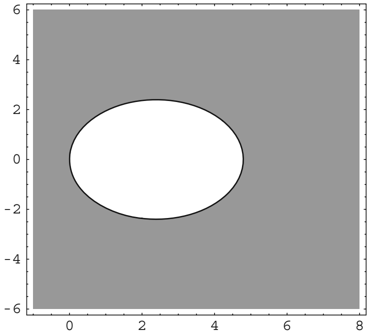
If $Q^{-1}A^*hQ = \Lambda^*h$, $\Lambda^* = diag[\lambda_1^*, ..., \lambda_m^*]$, as $b_i(A^*h)$ and $a_{ij}(A^*h)$ depend only on e^{A^*h}, Id and A^*h, then we have $Q^{-1}b_i(A^*h)Q = b_i(\Lambda^*h)$ and $Q^{-1}a_{ij}(A^*h)Q = a_{ij}(\Lambda^*h)$ and the system can be coupled.

So we can consider as a test problem the very famous Dahlquist's equation

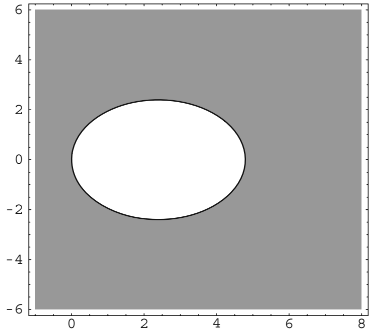
$$y'(x) = \mu y(x), \quad y_0 = 1, \quad z = h\mu, \tag{10}$$

where $Re(\mu) < 0$, with $\mu = \lambda + \nu$. That is, we have introduced the value λ in the method while the true solution depends on the exponential of μ. And we are going to calculate the set

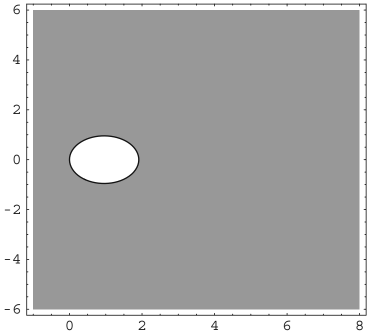
$$S = \{z \in C; \mid R(z) \mid \leq 1\},$$



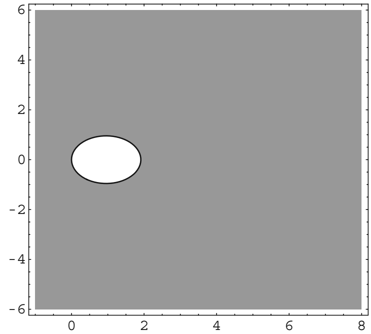
a $\lambda h = -1 + 4i$. *EF-Lobatto-2s method.*



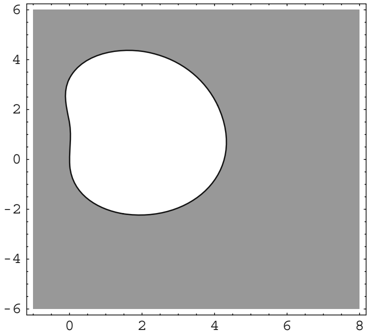
b $\lambda h = -1 - 4i$. *EF-Lobatto-2s method.*



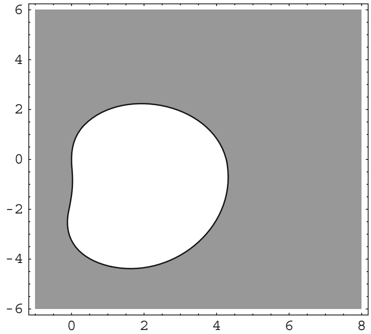
c $\lambda h = -3 + 12i$. *EF-Lobatto-2s method.*



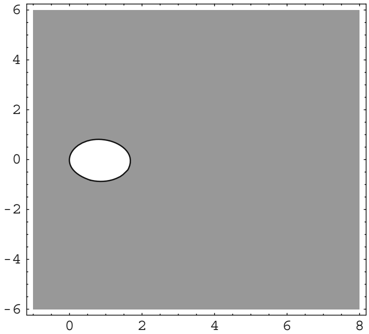
d $\lambda h = -3 - 12i$. *EF-Lobatto-2s method.*



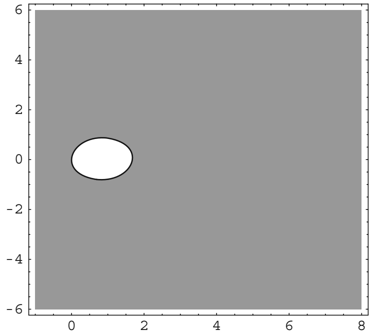
e $\lambda = -1 + 4i$. *EF-Radau-2s method.*



f $\lambda h = -1 - 4i$. *EF-Radau-2s method.*



g $\lambda h = -3 + 12i$. *EF-Radau-2s method.*



h $\lambda h = -3 - 12i$. *EF-Radau-2s method.*

called the stability domain of the method, where $R(z)$, the stability function of the method, is that proposed by Hairer and Wanner in [4]:

$$R(z) = 1 + zb^t(Id - zA)^{-1}\mathbf{1},$$

being $b^t = (b_1, \ldots, b_s)$, $A = (a_{ij})_{i,j=1}^s$ and $\mathbf{1} = (1, \ldots, 1)^t$, or (see [4, 29] or [30]), they are both the same (see [4], Proposition 3.2, p. 41)

$$R(z) = \frac{det\left(Id - zA + z\mathbf{1}b^t\right)}{det(Id - zA)}.$$

Again, as we mentioned in [16, 31, 32] or [33], it is impossible to plot regions of absolutely stability. However we can fix $u = h\lambda$ with different real and complex values and plot in the complex plane the values of νh that makes the method absolute stable.

We can begin by showing some regions of absolute stability of the Gauss-2s and Radau-2s method when $\lambda h \in R^-$. Since the methods approach to the classical methods when $\lambda h \to 0$, then both methods are A-stable in this case and when $\lambda h \to -\infty$, the regions of absolute instability are smaller and smaller, this means that the error (when we calculate the parameter on the method) can be bigger and the method continues being stable for the problem. We have shown some of these regions in Fig. 1.

The behavior of the regions of these methods when $\lambda h \in C^-$ is very similar as we can see in Fig. 2, where we have shown some stability regions of the Radau-2s and Lobatto methods. Again, when $ah \to -\infty$ (a being the real part of λ), the regions of absolute instability are smaller and smaller. We can check in this figure that if we choose $\lambda_1 h = a + ib \in \mathbb{C}^-$ $a, b \in R$ and $\lambda_2 h = a - ib$, then the regions of absolute stability were symmetric.

4 Derivation of the variable step Radau-2s method

We have used here an embedded pair of methods which is easy to program. We have derived the variable step Radau-2s method, but in a similar way we can derived other variable step methods of higher order or other kind of exponential fitting Runge-Kutta methods.

Since the EF-Radau-2s method is of optimal order, it is impossible to embed it efficiently into one of still higher order. Therefore we search for a lower order method of the form

$$\hat{y_{n+1}} = y_n + h\left(\hat{b_0} f(x_n, y_n) + \sum_{i=1}^{2} \hat{b_i} f(x_n + c_i h, u_i)\right), \tag{11}$$

where u_1, u_2 are the values obtained from the EF-Radau-2s method and $\hat{b_0} \neq 0$. For the numerical examples we chose $\hat{b_0} = 1/2$, $\hat{b_1} = 0$, $\hat{b_2} = 1/2$, so the new method has order two.

If we consider

$$err = y_{n+1} - \hat{y_{n+1}} = O\left(h^2\right),$$

we can use it for error estimation, then the standard step size prediction leads to

$$h_{n+1} = fac\ h_n\ \|err\|^{-1/2},$$

where

$$\|err\| = \sqrt{\frac{\sum_{i=1}^{n} err_i^2}{n\ Tol^2}},$$

and $fac = 0.9$ or 0.95 (for the numerical tests, we chose the more conservative factor of $fac = 0.9$).

5 Numerical examples

In other papers (see [34–38] or [33], for example), the chosen parameter has been a matrix, so two big open questions appeared in this field: which is the best way to calculate the exponential matrix and which is the best procedure to choose the parameter for the methods.

In this case, we have used scalar parameters $\lambda h \in R^-$ in the method.

We are going to suppose that the IVP is

$$y'(x) = g(x, y(x)), \qquad y(x_0) = y_0, \tag{12}$$

(where $y = [y^1, \ldots, y^m]$, and $g = [g^1, \ldots, g^m]$, $y_0 = \left[y_0^1, \ldots, y_0^m\right]$, $x \in R$). The steps to calculate the parameter $\lambda h \in R^-$ are the following:

1) In the first three steps we take the coefficients of the classical methods.
2) We are going to suppose, now, that we want to calculate y_{n+1} and we have calculated y_n, y_{n-1}, y_{n-2} and y_{n-3}.
   Since

$$\frac{\alpha_1 y^i_{x_n} + \alpha_2 y^i_{x_{n-1}} + \alpha_3 y^i_{x_{n-2}} + \alpha_4 y^i_{x_{n-3}}}{h^3_{n-1}} \approx \left(y^i_{x_n}\right)''',$$

   where $\alpha_1 = \frac{6}{(1+\beta)(1+\beta+\gamma)}$, $\alpha_2 = \frac{-6}{\beta(\beta+\gamma)}$, $\alpha_3 = \frac{6}{(\beta+\beta^2)\gamma}$, $\alpha_4 = \frac{-6}{(\beta+\beta^2+\gamma+\gamma^2+2\beta\gamma)\gamma}$, $\beta = \frac{h_{n-2}}{h_{n-1}}$, $\gamma = \frac{h_{n-3}}{h_{n-1}}$, $h_{n-1} = x_n - x_{n-1}$, $h_{n-2} = x_{n-1} - x_{n-2}$ and $h_{n-3} = x_{n-2} - x_{n-3}$, in a similar way

$$\frac{\alpha'_1 y^i_{x_n} + \alpha'_2 y^i_{x_{n-1}} + \alpha'_3 y^i_{x_{n-2}} + \alpha'_4 y^i_{x_{n-3}}}{h^2_{n-1}} \approx \left(y^i_{x_n}\right)'',$$

   where $\alpha'_1 = \frac{2(3+2\beta+\gamma)}{(1+\beta)(1+\beta+\gamma)}$, $\alpha'_2 = \frac{-2(2+2\beta+\gamma)}{\beta(\beta+\gamma)}$, $\alpha'_3 = \frac{2(2+\beta+\gamma)}{(\beta+\beta^2)\gamma}$, $\alpha'_4 = \frac{-2(2+\beta)}{(\beta+\beta^2+\gamma+\gamma^2+2\beta\gamma)\gamma}$, then

$$\lambda_i = \frac{\alpha_1 y^i_{x_n} + \alpha_2 y^i_{x_{n-1}} + \alpha_3 y^i_{x_{n-2}} + \alpha_4 y^i_{x_{n-3}}}{h_{n-1}\left(\alpha'_1 y^i_{x_n} + \alpha'_2 y^i_{x_{n-1}} + \alpha'_3 y^i_{x_{n-2}} + \alpha'_4 y^i_{x_{n-3}}\right)} \approx \frac{\left(y^i_{x_n}\right)'''}{\left(y^i_{x_n}\right)''},$$

and when $\beta = 1, \gamma = 1, \lambda_i = \frac{y_n^i - 3y_{n-1}^i + 3y_{n-2}^i - y_{n-3}^i}{2y_n^i - 5y_{n-1}^i + 4y_{n-2}^i - y_{n-3}^i} \approx \frac{(y_n^i)'''}{(y_n^i)''}$. Then, we choose

$$\lambda h_n = \max \lambda_i h_n.$$

This way to choose λh_n allows to cancel the leading truncation error of the EF-Lobatto-2s algorithm and partially the truncation errors of EF-Gauss-2s and EF-Radau-2s (the three of them are constructed to hold (4) when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$).

3) Since positive parameters or very negative values could give inaccuracies, if $\lambda h_n \geq 0$, then we have taken the weights a_{ij}, b_i and c_i of the classical methods. Finally, if $\lambda \leq -100$, then we have taken $\lambda = -100$.
The methods were developed in Mathematica and we used an Intel Pentium 4 with 1.40 GHz.

Problem 1 the first stiff problem is known as Robertson equation (see, for example, [4]),

$$\begin{aligned} y_1'(x) &= -0.04 y_1(x) + 10^4 y_2(x) y_3(x), \\ y_2'(x) &= 0.04 y_1(x) - 10^4 y_2(x) y_3(x) - 3\ 10^7 y_2^2(x), \\ y_3'(x) &= 3\ 10^7 y_2^2(x), \\ y_1(0) &= 1, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad 0 \leq x \leq x_f. \end{aligned} \tag{13}$$

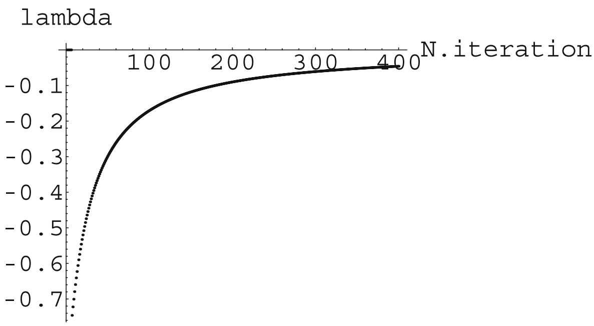
We show the evolution of the values of λ in Fig. 3, when we used constant $h = 0.1$ and the EF-Radau-2s method.

We have compared the numerical results at $x_f = 40$ of the traditional Radau-2s (of two steps) with constant step length and the EF-Radau-2s in Fig. 4.

In Fig. 5, we show the numerical errors at $x_f = 4000$ when we compared the variable step length methods Radau-2s and EF-Radau-2s.

We can observe that the error is smaller with the new algorithm, but the CPU Time (in seconds) is smaller, too. The reason is that the new scheme needed less iterations of the Newton's method to solve the nonlinear equation.

Fig. 3 Evolution of λ with constant $h = 0.1$ and the EF-Radau-2s method in numerical integration of Problem 1



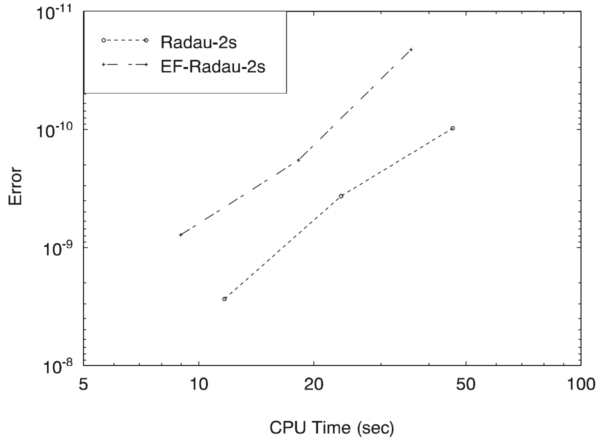


Fig. 4 Error in the numerical integration of Problem 1 at $x_f = 40$, using constant step length methods

Problem 2 the second stiff problem is known as Oregonator (see, for example, [4]). It is a famous model with a periodic solution describing the Belusov-Zhabotinskii reaction. The corresponding equations are:

$$\begin{aligned} y_1'(x) &= 77.27(y_1(x)(1 - 8.375 \times 10^{-6} y_1(x) - y_2(x)) + y_2(x)),\\ y_2'(x) &= \frac{1}{77.27}\left(-(1 + y_1(x))y_2(x) + y_3(x)\right),\\ y_3'(x) &= 0.161(y_1(x) - y_3(x)),\\ y_1(0) &= 1, \quad y_2(0) = 2, \quad y_3(0) = 3, \quad 0 \le x \le 30. \end{aligned} \tag{14}$$

In this case, we would like to show the evolution of the values of λ, when we used EF-Radau-2s with variable step length. In Fig. 6, we show these values when Tolerance is 10^{-3}.

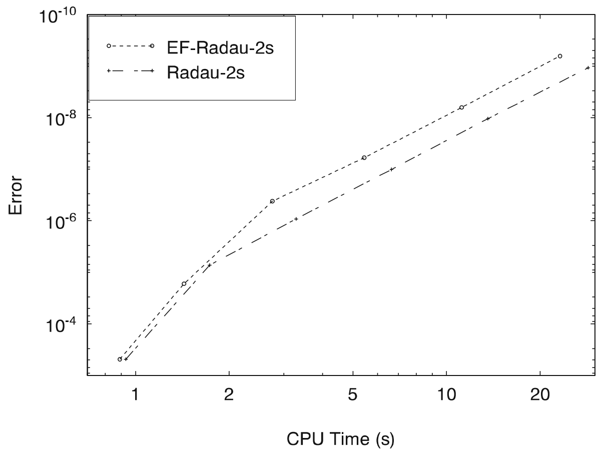


Fig. 5 Error in the numerical integration of Problem 1 at $x_f = 4000$. We used variable step Radau-2s and EF-Radau-2s methods

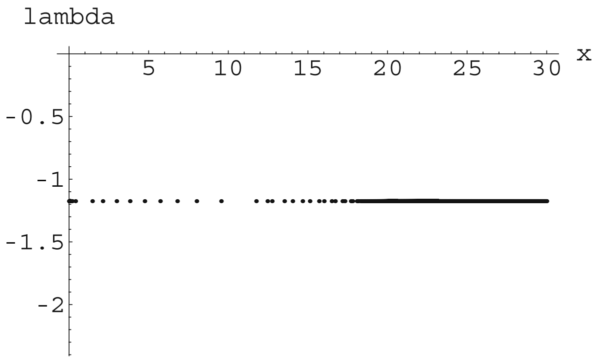


Fig. 6 Evolution of λ with variable h and the EF-Radau-2s method in the numerical integration of Problem 2

We have compared the numerical results of the traditional Radau-2s length and the EF-Radau-2s methods with constant step in Fig. 7 at the end point $x = 30$, and using variable step in Fig. 8 at the same point.

Problem 3 we integrate the nonlinear IVP proposed by Frank and van der Houwen (from CWI) [39] or Kaps [40]

$$\begin{cases} y_1'(x) = -1002 y_1(x) + 1000 y_2^2(x), \\ y_2'(x) = y_1(x) - y_2(x)(1 + y_2(x)), \\ y_1(0) = 1, \qquad y_2(0) = 1, \end{cases} \tag{15}$$

with solution

$$y_1(x) = e^{-2x}, \qquad y_2(x) = e^{-x},$$

the Jacobian of the right-hand side of this problem at the initial point has the eigenvalues $\lambda_1 = -1004$, $\lambda_2 = -1.00199$, then we can consider that this problem is stiff.

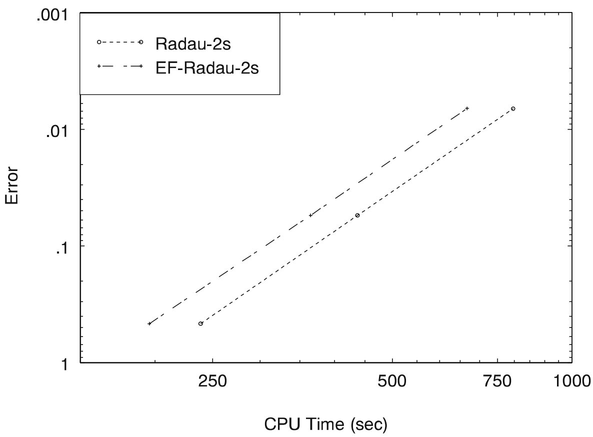


Fig. 7 Error in the numerical integration of Problem 2. In this case, we have used the constant step length Radau-2s and EF-Radau-2s methods

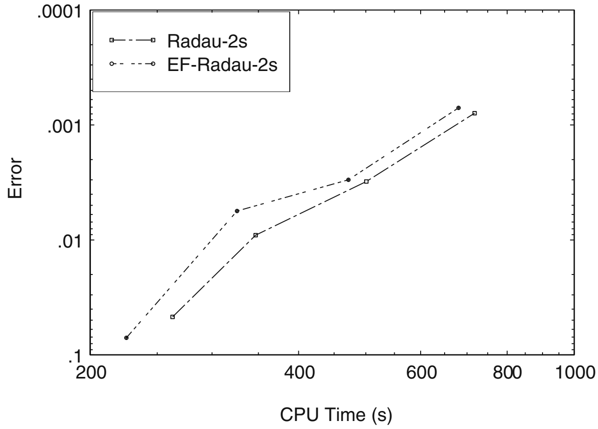


Fig. 8 Error in numerical integration of Problem 2. In this case, we have used the variable step length Radau-2s and EF-Radau-2s methods

In this case, we have used EF-Lobatto-2s with constant step length. In Fig. 9, we show the evolution of the values λ, when we used constant step length $h = 0.1$.

In Table 1 we have compared the results obtained at point $x = 5$ with Lobatto-2s and EF-Lobatto-2s, using different step lengths.

In that table, we can observe that the error is smaller with the new algorithm, but the CPU Time (in seconds) is smaller, too. The reason is that the new scheme needed less iterations of the Newton's method to solve the nonlinear equation.

Problem 4 this is a diffusion equation, from [41]. The problem is

$$u_t = u_{xx}, \quad x \in [0, 1],$$

with initial values $u(0, x) = a\sin(\sqrt{2}x) - \sin(x)$ and boundary values $u(t, 0) = 0$, $u(t, 1) = ae^{-2t}\sin(\sqrt{2}) - e^{-t}\sin(1)$, where $a = \cos(\sqrt{2})/(\sqrt{2}\cos(2^{-1/2}))$.

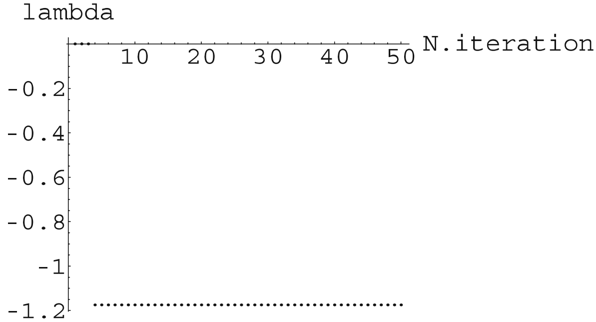


Fig. 9 Evolution of λ with constant $h = 0.1$ and the EF-Lobatto-2s method in the numerical integration of Problem 3

Table 1 Error in the numerical integration of Problem 3

Step length	Method	Error	CPU Time (s)
$h = 0.1$	Lobatto-2s	2.8094×10^{-5}	0.53
	EF-Lobatto-2s	4.18206×10^{-7}	0.54
$h = 0.02$	Lobatto-2s	1.1244×10^{-6}	1.181
	EF-Lobatto-2s	1.99152×10^{-8}	1.872

Consider the variables $y_i(t) = u(t, i/(N+1)), i = 1, \ldots, N$. Then we have the spatially discrete problem

$$y'(t) = (N+1)^2 \begin{pmatrix} -2 & 1 & 0 & 0 \ldots & 0 & 0 \\ 1 & -2 & 1 & 0 \ldots & 0 & 0 \\ 0 & 1 & -2 & 1 \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \ldots & -2 & 1 \\ 0 & 0 & 0 & 0 \ldots & 1 & -2 \end{pmatrix} y(t) + (N+1)^2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \phi(t) \end{pmatrix} \tag{16}$$

with $\phi(t) = ae^{-\nu t}\sin(\sqrt{2}) - e^{-\mu t}\sin(1)$ and

$$\mu = 2(N+1)^2\left(1 - \cos\left(\frac{1}{N+1}\right)\right), \quad \nu = 2(N+1)^2\left(1 - \cos\left(\frac{\sqrt{2}}{N+1}\right)\right).$$

The exact solution of the discretised problem is

$$y_i(t) = ae^{-\nu t}\sin\left(\frac{\sqrt{2}i}{N+1}\right) - e^{-\mu t}\sin\left(\frac{i}{N+1}\right), \tag{17}$$

and we have the property that $\mu \to 1$ and $\nu \to 2$ as $N \to \infty$.

If we consider $N = 100$ in this problem (for which μ and ν are approximately $\mu = 0.9999918308928846$ and $\nu = -1.9999673236788025$), we get the results at $t = 1$ showed in Fig. 10.

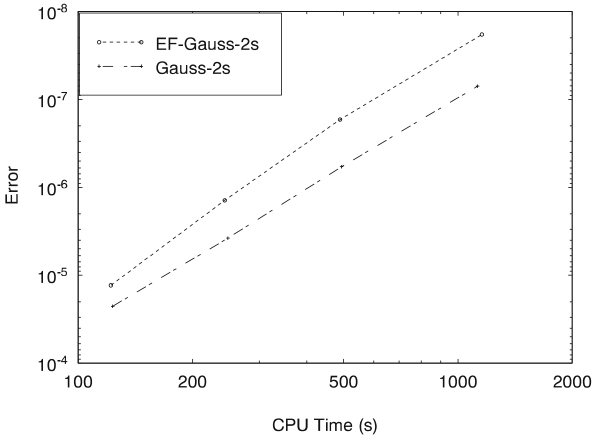


Fig. 10 Errors in the numerical integration of Problem 4 $N = 100$. In this case, we have used the constant step length Gauss-2s and EF-Gauss-2s methods

Acknowledgements This work has been supported by project CSD2006-0032 (program i-MATH of the Spanish ministry of education and science).

References

1. Curtiss, C.F., Hirschfelder, J.O.: Integration of stiff equations. Proc. Natl. Acad. Sci. U. S. A. **38**, 235–243 (1952)
2. Aitken, R.C.: Stiff Computation. Oxford University Press, New York (1985)
3. Spijker, M.N.: Stiffness in the numerical initial-value problems. J. Comput. Appl. Math. **72**, 393–406 (1996)
4. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II. Springer, Berlin (1993)
5. Higham, D.J.: Stiffness of ODEs. BIT **33**, 285–303 (1993)
6. Lambert, J.D.: Numerical Methods for Ordinary Differential Systems. The Initial Value Problem. Wiley, Chichester (1991)
7. Gear, C.W.: Algorithm 407-DIFSUB for solution of ordinary differential equations. Commun. ACM **14**, 185–190 (1971)
8. Hindmarsh, A.C.: LSODE and LSODI, two new initial value ordinary differential equation solvers. ACM Signum Newslett. **15**, 10–11 (1980)
9. Brown, P.N., Byrne, G.D., Hidmarsh, A.C.: VODE: a variable-coefficient ODE solver. SIAM J. Sci. Statist. Comput. **10**, 1038–1051 (1989)
10. Petzold, L.R.: A description of DASSL: a differential-algebraic system solver. In: Stepleman, R.S., et al. (eds.) IMACS Trans. Scientific Computing, pp. 65–68. North-Holland, Amsterdam (1993)
11. Cash, J.R.: The integration of stiff initial value problems in ODE's using modified extended backward differentiation formulae. Comput. Math. Appl. **9**, 645–657 (1983)
12. Fredebul, C.: A-BDF: a generalization of the backward differentiation formulae. SIAM J. Numer. Anal. **35**, 1917–1938 (1998)
13. Ixaru, L.Gr., Rizea, M., Vanden Berghe, G., De Meyer, H.: Weights of the exponential fitting multistep algorithms for ODEs. J. Comput. Appl. Math. **132**, 83–93 (2001)
14. Ixaru, L.Gr., Vanden Berghe, G., De Meyer, H.: Frequency evaluation in exponential fitting multistep algorithms for ODEs. J. Comput. Appl. Math. **140**, 423–434 (2002)
15. Ixaru, L.Gr., Vanden Berghe, G., De Meyer, H.: Exponentially fitted variable two-step BDF algorithms for first order ODEs. Comput. Phys. Commun. **100**, 56–70 (2003)
16. Martin-Vaquero, J., Vigo-Aguiar, J.: Adapted BDF Algorithms: higher-order methods and their stability. J. Sci. Comput. **32**, 287–313 (2007)
17. Alexander, R.: Diagonally implicit Runge-Kutta methods for stiff ODEs. SIAM J. Numer. Anal. **14**, 1006–1021 (1977)
18. Nørsett, S.P.: Semi explicit Runge-Kutta methods. Mathematics and Computing Rpt. N. 6/74, University of Trodheim (1974)
19. Darvishi, M.T., Khani, F., Kheybari, S.: Spectral collocation solution of a generalized Hirota-Satsuma coupled KdV equation. Int. J. Comput. Math. **84**, 541–552 (2007)
20. Khater, A.H., Temsah, R.S.: Numerical solutions of some nonlinear evaluation equations by Chebyshev spectral collocation methods. Int. J. Comput. Math. **84**, 305–316 (2007)
21. Simos, T.E., Vigo-Aguiar, J.: A modified Runge-Kutta method with phase-lag of order infinity for the numerical solution of the Schrodinger equation and related problems. Comput. Chem. **25**, 275–281 (2001)
22. Simos, T.E., Vigo-Aguiar, J.: Symmetric eighth algebraic order methods with minimal phase-lag for the numerical solution of the Schrodinger equation. J. Math. Chem. **31**, 135-144 (2002)
23. Liniger, W., Willoughby, R.: Efficient integration for stiff systems of ordinary differential equations. SIAM J. Numer. Anal. **7**, 47–66 (1970)
24. Van de Vyver, H.: Frequency evaluation for exponentially fitted Runge-Kutta methods. J. Comput. Appl. Math. **184**, 442–463 (2005)
25. Vanden Berghe, G., Van Daale, M., Vande Vyver, H.: Exponential fitted Runge-Kutta methods of collocation type: fixed or variable knot points. J. Comput. Appl. Math. **159**, 217–239 (2003)

26. Butcher, J.C.: The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley, Chichester (1987)
27. Vigo-Aguiar, J., Martín-Vaquero, J., Ramos, H.: Exponential fitting BDF-Runge-Kutta algorithms. Comput. Phys. Commun. **178**, 15–34 (2008)
28. Coleman, J.P., Ixaru, L.Gr.: P-stability and exponential-fitting methods for y”=f(x,y). IMA J. Numer. Anal. **16**, 179–199 (1996)
29. Scherer, R.: A necessary condition for B-stability. BIT **19**, 111–115 (1979)
30. Stetter, H.J.: Analysis of Discretization Methods for Ordinary Differential Equations. Springer, Berlin (1973)
31. Martín-Vaquero, J., Vigo-Aguiar, J.: Exponential fitting BDF algorithms: explicit and implicit 0-stable methods. J. Comput. Appl. Math. **192**, 100–113 (2006)
32. Vigo-Aguiar, J., Martín-Vaquero, J., Criado, R.: On the stability of exponential fitting BDF algorithms. J. Comput. Appl. Math. **175**, 183–194 (2005)
33. Vigo-Aguiar, J., Martín-Vaquero, J.: Exponential fitting BDF algorithms and their properties. Appl. Math. Comput. **190**, 80–110 (2007)
34. Edwards, W.S., Tuckerman, L.S., Friesner, R.A., Sorensen, D.C.: Krylov methods for the incompressible Navier-Stokes equations. J. Comput. Phys. **110**, 82–102 (1994)
35. Friesner, R.A., Tuckerman, L.S., Dornblaser, B.C., Russo, T.C.: A method for exponential propagation of large systems of stiff nonlinear differential equations. J. Sci. Comput. **4**, 327–354 (1989)
36. Gallopoulos, E., Saad, Y.: Efficient solution of parabolic equations by Krylov approximation methods. SIAM J. Sci. Stat. Comput. **13**, 1236–1264 (1992)
37. Kosloff, R.: Propagation methods for quantum molecular dynamics. Annu. Rev. Phys. Chem. **45**, 145–178 (1994)
38. Sidje, R.B.: Expokit: software package for computing matrix exponentials. ACM Trans. Math. Softw. **24**, 130–156 (1998)
39. Frank, J.E., van der Houwen, P.J.: Parallel Iteration of the Extended Backward Differentiation Formulas. Report MAS-R9913, CWI (1999)
40. Kaps, P.: Rosenbrock-type methods. In: Dahlquist, G., Jeltsch, R. (eds.) Numerical Methods for Stiff Initial Value Problems. Bericht nr. 9, Inst für Geometrie und Praktische Mathematik der RWTH Aachen, Aachen (1981)
41. Butcher, J.C., Rattenbury, N.: ARK methods for stiff problems. Appl. Numer. Math. **53**, 165–181 (2005)