On the numerical solution of stiff IVPs by Lobatto IIIA Runge–Kutta methods.

S. González Pinto [a], S. Pérez Rodríguez [a] and J.I. Montijano Torcal [b]

[a] *Dpto. de Análisis Matemático, Universidad de La Laguna, 38271 La Laguna-Tenerife, Spain.*

[b] *Dpto. de Matemática Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain.*

Abstract

In this paper, the numerical solution of stiff initial value problems by means of Lobatto IIIA Runge–Kutta methods is considered. New iterative schemes for an efficient solution of the implicit equations associated to the 4th and 6th–order Runge–Kutta Lobatto IIIA formulae are proposed. Finally, the results of some numerical experiments are presented to show that, when implemented with the new iterative schemes, Lobatto IIIA formulae can be competitive with standard codes used in solving stiff systems.

1 Introduction

We consider the numerical solution of stiff initial value problems of ordinary differential equations

$$y'(t) = f(t, y(t)), \quad t \in [t_0, t_f], \qquad y(t_0) = y_0 \in I\!R^m, \tag{1.1}$$

where the derivative function $f(t, y)$ is supposed to be sufficiently smooth so that (1.1) has a unique solution $y(t)$.

Let y_n be an approximation to $y(t_n)$, an s–stage Runge–Kutta (RK) method applied to (1.1) at the gridpoint t_n with stepsize h can be written in the form

[1] Partially supported by DGICYT, project PB93-0305

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \;\; i = 1, ..., s, \tag{1.2}$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_i f(t_n + c_i h, Y_i), \tag{1.3}$$

where $c_i = \sum_{j=1}^{s} a_{ij}$ and a_{ij}, b_i are the coefficients of the method, usually represented by the matrix $A = (a_{ij})_{i,j=1}^{s}$ and the vector $b = (b_1, \ldots, b_s)^T$.

As it is widely recognized, the implementation difficulties of implicit Runge–Kutta methods have precluded their general use. However there are some families of RK methods such as the Radau IIA of Ehle that possess very good stability and convergence properties for stiff problems. Thus, they are L–stable and in the frame of the B–theory they are algebraically and diagonally–stable. Such a theoretical support has lead to Hairer & Wanner to construct a code RADAU5 (see [16], pp. 118) based in the three–stages Radau IIA formula of order 5, paying special attention to the implementation aspects in order to get a reliable and efficient code.

In this paper we are interested in a special family of RK methods, the so called Lobatto IIIA formulae, whose coefficients satisfy (see e.g. Butcher [6], pp. 227):

$$a_{1j} = 0, \quad j = 1, ..., s \qquad a_{sj} = b_j, \quad j = 1, ..., s. \tag{1.4}$$

These methods are A–stable, but they are neither L–stable nor algebraically–stable, thus their potential properties are clearly inferior to other families like Radau IIA, Gauss, etc. in the light of the B–theory (for a detailed study of the stability properties of RK methods for stiff problems we refer to Dekker & Verwer [12] or also to Hairer & Wanner [16]). In view of this, one may wonder why to study these type of methods for the numerical solution of stiff systems. As we will see, there are other theoretical and practical features that make it worthwhile to analyze its potential properties.

First of all, let us recall that the Lobatto IIIA methods are stiffly accurate. This means that when applied to the scalar test problem of Prothero & Robinson [18] $y' = \lambda(y - \varphi(t)) + \varphi'(t)$, $\mathrm{Re}\lambda \leq 0$, the local error for the s–stage formula behaves as $z^{-1}h^{s+1}$ with $z = h\lambda$, and therefore it gives an asymptotically exact result for $z \to \infty$ and the global errors for variable stepsizes behave as $z^{-1}h^{s}$. Of course, the same behaviour is shared by the s–stages Lobatto IIIC and Radau IIA methods, although the global error of the last method has one order higher than the others (see [16], pp. 242). Therefore, as the stiffly accuracy concerns, Lobatto IIIA are among the best methods. Further as remarked in [16], Chap. VI, the stiffly accuracy has relevant implications in the numerical solution of singularly perturbed and semi–explicit differential–algebraic problems.

On the other hand, since the first stage of Lobatto IIIA methods is explicit, that is, $Y_1 = y_n$, the implicit system of sm algebraic equations (1.2) reduces to a system of $(s-1)m$ implicit equations, and therefore they have a computational cost equivalent to a full (e.g. Gauss, Radau IIA or Lobatto IIIC) $(s-1)$–stages Runge–Kutta method. This means that from a computational point of view an s–stages Lobatto IIIA should be compared with an $(s-1)$–stages Gauss, Radau IIA or Lobatto IIIC method.

Further, although Lobatto IIIA methods are not B–stable and therefore the B–theory can not be applied, it has been proved by Calvo et al. [8] that they are stable and convergent (independently of the stiffness) for a class of semilinear, variable–coefficient stiff problems. Also the same authors have proved that they are stable and convergent of order s [9] for a class of non–linear stiff problems assuming some property on the relative variation of the Jacobian matrix. They also have shown that this property is satisfied by many stiff problems that appear in practice.

Concerning the L–stability requirement on the methods for the numerical solution of stiff systems, for symmetric schemes, like Gauss or Lobatto IIIA, the infinity point belongs to the boundary of the stability region and as a consequence the very stiff components in the solution of the problem are damped out only very slowly, while for L–stable methods the stiff components are damped out very fast. This difference of behaviour is shown in [16], pp. 44-46, by applying the implicit Euler and the trapezoidal rule to the simple linear problem $y' = -2000(y - \cos x)$, $y(0) = 0$. Integrating the problem with fixed stepsize, it is observed that implicit Euler damps out the transient phase much faster than the trapezoidal rule. For the last method, it is shown that the numerical solution oscillates violently around the true solution and the global errors decrease slowly. However, in practice numerical codes adjust the stepsize according to a local error user requirement. Then such a requirement makes the code select during the transient phase a small enough steplength such that the errors are below the prescribed tolerance and afterwards the stepsize is increased to a ”normal” value when this fast transient is dead. Therefore, in the context of such a variable stepsize the role of L–stability is not so crucial as in the fixed stepsize.

L–stable methods can also present some inconvenient compared to symmetric methods. Thus, when the problem has a non–dissipative wave in the stationary solution, the oscillations of the true solution may be damped by an L–stable method, and the numerical solution may lead to wrong conclusions on the qualitative behaviour of the solution. A discussion on L-stable methods and A–stable but non L–stable methods can also be found in [17], pp. 224–231.

On the other hand, Ascher in [1] has shown that symmetric schemes like Gauss or Lobatto IIIA present a potential risk when applied to initial value differential algebraic equations and some very stiff problems. To illustrate this fact, he gives a linear DAE equation of index 2, showing that when solved with the implicit midpoint rule, although second order convergent, their stability constants may become arbitrary large. He claims that this phenomenon appears in higher order symmetric schemes and also for very stiff problems. Here again, we have found that solving the singularly perturbed example presented in [1] with the standard stepsize changing technique, the code manages to control the integration so that the global errors are in accordance with the tolerance prescribed by the user.

The paper is organized as follows:

In Section 2 we introduce some iterative schemes to the solution of the algebraic equations of the stages which arises in the Lobatto IIIA methods that follow the ideas in [14]. A study of the linear stability properties of the iterative schemes is carried out proposing the requirements to be satisfied by the schemes. In Section 3 we obtain efficient iterative schemes for the 3 and 4-stage methods that satisfy the requirements of the above section. Finally, in Section 4 we present some numerical experiments to compare the efficiency of the new iterative schemes with simplified Newton methods and also, to compare the performance of a code based on the Lobatto IIIA method with other standard codes. In particular we present a code based on the sixth–order Lobatto IIIA formula and compare it with EPISODE [7], based on BDF formulas, RADAU5 [16], based on the 5th–order Radau IIA method, STRIDE [3], based on the so called Singly–Implicit RK formulas and the code developed by Gonzalez et al. in [14], based on the 5th–order Radau IIA method and that uses an iterative scheme of the type considered here for Lobatto IIIA.

2 Iterative schemes for Lobatto IIIA formulae

Let us consider an s-stage Lobatto IIIA method with coefficient matrix $A = (a_{ij})_{i=1}^{s}$, that, as mentioned before, has the form

$$A = \begin{pmatrix} 0 & O^T \\ w & \bar{A} \end{pmatrix},$$

where $w = (a_{21}, \ldots, a_{s1}) \in I\!R^{s-1}$, $O^T = (0, 0, ..., 0) \in I\!R^{s-1}$, and $\bar{A} = (a_{ij})_{i,j=2}^{s}$ is the $(s-1) \times (s-1)$ lower submatrix that is nonsingular.

Introducing the vectors

$$Y = \begin{pmatrix} Y_2 \\ \vdots \\ Y_s \end{pmatrix}, \qquad F(Y) = \begin{pmatrix} f(t_n + c_2 h, Y_2) \\ \vdots \\ f(t_n + c_s h, Y_s) \end{pmatrix},$$

equations (1.2) (1.3) can be written in matrix form as follows:

$$\begin{aligned} Y_1 &= y_n, && (2.1)\\ Y &= \bar{e} \otimes y_n + h(w \otimes f(t_n, y_n)) + h(\bar{A} \otimes I)F(Y), && (2.2)\\ y_{n+1} &= y_n + hb_1 f(t_n, y_n) + h(\bar{b}^T \otimes I)F(Y), && (2.3) \end{aligned}$$

where $\bar{e} = (1, ..., 1)^T \in I\!R^{s-1}$, $\bar{b}^T = (b_2, ..., b_s)$ and $\otimes$ denotes the Kronecker product of matrices.

A standard approach to solve (2.2) in stiff problems is to use a Newton type method [16]. In this case, each iteration step requires the solution of a set of $m(s-1)$ linear equations with a matrix which depends on J, the jacobian matrix of f evaluated at some previously computed point. To reduce the computational cost of the LU matrix factorization, a similarity transformation is introduced with the purpose to diagonalize the Runge–Kutta submatrix $\bar{t}ildeA$ following the lines of [16], Section IV.8.

Other algorithms which in each iteration step require the solution of s linear systems of order m with the same matrix of coefficients have been proposed [2,4,5,10,11,13,14]. In particular, Cooper & Butcher [10] and Cooper & Vignesvaran [11] proposed efficient iterative schemes for Gauss methods, and more recently González et al. [14] studied the construction of iterative schemes for three–stage implicit RK methods, obtaining efficient schemes for the case of Gauss and Radau IIA formulas.

Here, following the ideas in [14] we will consider iterative schemes, that we will call single–Newton schemes, in which for a given approximation Y^{k-1} to the solution Y of (2.2), a new approximation Y^k is given by

$$\begin{cases} [I - h\gamma(I \otimes J)]\, E^k = ((I-L)S^{-1} \otimes I)D^{k-1} + h(L \otimes I)E^k, \\ Y^k = Y^{k-1} + (S \otimes I)E^k, \quad k = 1, 2, ..., \end{cases} \qquad (2.4)$$

where D^{k-1} denotes the defect of Y^{k-1}, i.e. $D^{k-1} = \bar{e} \otimes y_n + h(w \times f(t_n, y_n)) - Y^{k-1} + h(\bar{A} \otimes I)F(Y^{k-1})$. The real constant $\gamma > 0$ and the $(s-1) \times (s-1)$ real constant matrices S and L, where S is non–singular and L is strictly lower triangular, must be chosen appropriately.

Next, we want to find the scalar γ and the matrices S, L so that (2.4) solves (1.2) efficiently. To this end, we will try to make the scheme optimum in some sense for linear problems.

Applying (2.4) to the classical test equation $y' = \alpha y$, and putting $z = h\alpha$, it can be seen that the errors in two consecutive iteration steps are related by

$$Y - Y^k = M(z)(Y - Y^{k-1}) = M(z)^k (Y - Y^0) \tag{2.5}$$

with

$$M(z) = z(I - zT)^{-1}(\bar{A} - T), \qquad T = \gamma S(I - L)^{-1} S^{-1}. \tag{2.6}$$

Moreover, for $f(t,y) = \alpha y$, it follows from (2.2) $Y = (I - z\bar{A})^{-1}(\bar{e} + zw)y_n$, that replaced in (2.5) gives $Y^k = M(z)^k Y^0 + (I - M(z)^k)(I - z\bar{A})^{-1}(\bar{e} + zw)y_n$.

Now, taking into account that $y_{n+1}^k = Y_s^k = e_{s-1}^T Y^k$ with $e_{s-1}^T = (0, \ldots, 0, 1) \in I\!R^{s-1}$, we arrive at

$$\begin{aligned} y_{n+1}^k = e_{s-1}^T M(z)^k Y^0 + e_{s-1}^T (I - z\bar{A})^{-1}(\bar{e} + zw)y_n - \\ e_{s-1}^T M(z)^k (I - z\bar{A})^{-1}(\bar{e} + zw)y_n\ . \end{aligned}$$

Denoting $R_k(z)$ the amplifying function of the kth approximation defined by $y_{n+1}^k = R_k(z)y_n$ and $R(z) = e_{s-1}^T (I - z\bar{A})^{-1}(\bar{e} + zw)$ the amplifying function of the Lobatto IIIA method, we have

$$\begin{aligned} R_k(z) = R(z) + e_{s-1}^T M(z)^k Y^0 / y_n + \\ e_{s-1}^T M(z)^k \left(I - z^{-1}\bar{A}^{-1}\right)^{-1} \bar{A}(z^{-1}\bar{e} + w)\,. \end{aligned} \tag{2.7}$$

Clearly, if the spectral radius of $M(z)$, $\rho(M(z))$, is less than one then $M^k(z) \to 0$ when $k \to \infty$, the iterative scheme converges and $R_k(z)$ converges to $R(z)$.

We are now in position to state the following

Theorem 1 $R_k(\infty) = R(\infty)$ *for all* $k \geq 1$ *and for all starting values* Y^0 *if and only if*

$$e_{s-1}^T M(\infty) = e_{s-1}^T (I - T^{-1}\bar{A}) = 0.$$

PROOF. It is an immediate consequence of (2.7). ♢

Theorem 2 *If* $\rho(M(\infty)) = 0$*, then for all fixed starting values* Y^0

i) $R_k(\infty) = R(\infty)$ *if* $k \geq s - 1$.

ii) $R_k(z) = R(z) + \mathcal{O}(1/z^q)$ *for all* $k \geq (s-2)q+1$.

PROOF. Since $M(\infty)$ is an $(s-1)\times(s-1)$ matrix and $\rho(M(\infty)) = 0$, it is clear that $M(\infty)^k = 0$ for all $k \geq s-1$ and from (2.7), i) holds.

On the other hand, calling $M = M(\infty)$ and expanding $M(z)$ in powers of $x = 1/z$ we have $M(z) = (I + xT^{-1} + x^2T^{-2} + \cdots)M$. Therefore,

$$M(z)^k = M_0^{(k)} + M_1^{(k)}x + M_2^{(k)}x^2 + \dots$$

with $M_0^{(k)} = M^k$, $\quad M_1^{(k)} = \sum_{j=0}^{k-1} M^jT^{-1}M^{k-j}$ and $M_r^{(k)}$ is a sum containing T^{-1} r times and M k times. If $k \geq (s-2)q+1$, at least a power of M^{s-1} occurs in $M_r^{(k)}$ for $r \leq q-1$ and hence $k \geq (s-2)q+1$ implies $M_r = 0$. $\diamondsuit$

Theorem 3 *If* $\rho(M(\infty)) = 0$ *and* $e_{s-1}^T M(\infty) = 0$*, then* $R_k(z) = R(z) + \mathcal{O}(1/z^q)$ *for all* $k \geq (s-2)q+3-s$ *and any starting values* Y^0.

PROOF. If we expand $e_{s-1}^T M(z)^k$ in powers of $x = 1/z$ we obtain

$$e_{s-1}^T M(z)^k = e_{s-1}^T M_0 + x e_{s-1}^T M_1 + \cdots$$

For $r \leq -1$ if $k \geq r(s-2)+1$ then $e_{s-1}^T M_r$ contains either a leading term $e_{s-1}^T M(\infty)$ or at least a power $M(\infty)^{s-1}$. $\diamondsuit$

Under conditions of Theorem 3, the 3-stage Lobatto IIIA formula satisfies

$$R_k(z) = R(z) + \mathcal{O}(1/z^k)\ \forall k \geq 1$$

that is, the order of approximation of the amplifying function $R_k(z)$ at infinity increases one unity at each iteration. For the 4–stage Lobatto IIIA,

$$R_k(z) = R(z) + \mathcal{O}(1/z^q)\ \forall k \geq 2q-1$$

and the order is increased by one unity after two iterations of the scheme.

Let us note finally that, since $(I - T^{-1}\bar{A})\bar{A}^{-1}T = -(I - \bar{A}^{-1}T)$, then $e_{s-1}^T M(\infty) = 0$ if and only if $e_{s-1}^T(I - \bar{A}^{-1}T)) = 0$.

After the above results and following the ideas in [14], to obtain efficient iterative schemes of the form (2.4) for Lobatto IIIA methods, we will look for matrices T with a unique eigenvalue $\gamma > 0$, so that they can be factorized in the form (2.6), and satisfying if possible the conditions

P1.– $\rho(I - T^{-1}\bar{A}) = 0$

P2.– $e_{s-1}^T(I - \bar{A}^{-1}T) = 0$

that guarantee a good stability behaviour of the approximations and also a good rate of convergence for linear problems in a neighbourhood of the infinity point.

3 Developing iterative schemes for 3 and 4–stage Lobatto IIIA methods.

In this section we consider the effective construction of iterative schemes of the form (2.4) for the Lobatto IIIA methods with three and four stages. According to the above section, our aim will be to find for each of these two formulas a matrix T that yields iterative schemes with a suitable rate of convergence as well as adequate linear stability properties. Therefore, we will look for matrices T with a unique eigenvalue and satisfying conditions P1 and P2.

3.1 Three-stage, fourth-order Lobatto IIIA.

In the case $s = 3$, the submatrix $\bar{A}$ and the vector w are given by

$$\bar{A} = \begin{pmatrix} 1/3 & -1/24 \\ 2/3 & 1/6 \end{pmatrix}, \qquad w = \begin{pmatrix} 5/24 \\ 1/6 \end{pmatrix}.$$

If we denote $P = I - \bar{A}^{-1}T$, condition P2 demands

$$P = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}$$

Moreover, condition P1 is equivalent to $\rho(P) = 0$, which implies $a = 0$. Finally, $T = \bar{A}(I - P)$ has a unique eigenvalue $\gamma > 0$ if and only if $det(T) = det(\bar{A}) = \gamma^2$ and $tr(T) = 2\gamma$. Therefore, $\gamma = \sqrt{det(\bar{A})} = 1/\sqrt{12}$ and $tr(T) = tr(\bar{A}(I - P)) = \dfrac{3 - 4b}{6} = \dfrac{1}{\sqrt{3}}$. Then $b = (3 - 2\sqrt{3})/4$, and consequently for 3-stage Lobatto IIIA method there is a unique matrix T with only one eigenvalue that satisfies P1 and P2. This matrix is given by

$$T = \begin{pmatrix} 1/3 & -\dfrac{7}{24} + \dfrac{1}{2\sqrt{3}} \\ 2/3 & -\dfrac{1}{3} + \dfrac{1}{\sqrt{3}} \end{pmatrix}$$

In the practical use of the scheme we will use the expression (2.4) and since the choice of S does not influence $M(z)$ when the matrix T is given, from a practical point of view it is convenient to have a matrix S as simple as possible. In this particular case it is possible to get an upper triangular matrix S given by

$$S = \begin{pmatrix} 1 & 0.0669872981077806766 \\ 0 & 1 \end{pmatrix},$$

such that $S^{-1}TS$ is lower triangular and

$$L = I - \gamma S^{-1}T^{-1}S = \begin{pmatrix} 0 & 0 \\ 2.30940107675850306 & 0 \end{pmatrix}.$$

We have also computed for this scheme the spectral radius of the matrix $M(z)$ over the real axis and also the imaginary one, obtaining the following maximum values:

$$\max\{\rho(M(z))\,|z \in (-\infty, 0)\} = \rho(M(-2\sqrt{3})) = \frac{2-\sqrt{3}}{4} = 0.066987298...$$

$$\max\{\rho(M(iy))\,|y \in I\!R\} = \rho(M(2\sqrt{3}i)) = \frac{2-\sqrt{3}}{2} = 0.133974596...$$

which give a measure of the rate of convergence of the proposed scheme for linear problems.

To evaluate the linear stability properties of the scheme, we have obtained for each iteration k the amplifying function $R_k(z)$, taking for simplicity $Y^0 = e \otimes y_n$ as starting values. Then we have computed the maximum value θ_k such that the k-th approximation is A(θ_k)–stable, that is, the maximum θ_k such that if $z \in \{|\arg(-z)| \leq \theta_k\}$ then $|R_k(z)| \leq 1$. In table 1 we present the values of θ_k for $k = 1, 2, \ldots, 6$. In the table we also include the minimum value α_k such that $|R_k(z)| \leq 1$ for all Re $z \leq -\alpha_k$. As it can be observed, the first and second iterations are A–stable and for $k \geq 3$ the stability regions approximate quickly to the negative complex half–plane, the stability region of Lobatto IIIA method.

It must be noticed that standard simplified Newton iterations gives the solution of the stage equations at the first iterate when the method is applied to constant coefficient linear differential equations while single–Newton schemes may require several iterations to achieve a sufficiently accurate solution.

On the other hand, it was proved in [13] that a nonlinear scheme related to (2.4) converges when it is applied to nonlinear dissipative systems (1.1)

Table 1
Stability regions of 3–stage Lobatto IIIA.

k	θ_k	α_k
1	90.00°	0.00
2	90.00°	0.00
3	89.9924594192623°	0.00171599547911613
4	89.9971724227644°	0.00042079502790675
5	89.9993993611608°	0.00007243526445789
6	89.9998952656572°	0.00001114772053784

satisfying

$$\langle f(t,y) - f(t,\tilde{y}), y - \tilde{y} \rangle \leq 0, \quad \forall t \in I\!R, y, \tilde{y} \in I\!R^m$$

if there exists a positive–definite diagonal matrix D such that

(a) $M = DT + T^T D = N^T N$ is positive–definite and
(b) $||N^{-T} D(T - \bar{A})N^{-1}||_2 < 1/2$.

with T given by (2.6).

Moreover, the quantity $\nu = 2 \inf_{D>0} ||N^{-T} D(T - \bar{A})N^{-1}||_2$, where $D > 0$ denotes all positive-definite diagonal matrices satisfying (a), is a good measure of the speed of convergence. For the proposed scheme, we have found that the matrix $D = \mathrm{diag}(1, 0.2470356378869429)$ gives the minimum value $\nu = 0.2046574854265076$.

3.2 Four-stage, sixth-order Lobatto IIIA.

For this method

$$\bar{A} = \begin{pmatrix} \dfrac{25-\sqrt{5}}{120} & \dfrac{25-13\sqrt{5}}{120} & \dfrac{-1+\sqrt{5}}{120} \\ \dfrac{25+13\sqrt{5}}{120} & \dfrac{25+\sqrt{5}}{120} & \dfrac{-1-\sqrt{5}}{120} \\ \dfrac{5}{12} & \dfrac{5}{12} & \dfrac{1}{12} \end{pmatrix}, \qquad w = \begin{pmatrix} \dfrac{11+\sqrt{5}}{120} \\ \dfrac{11-\sqrt{5}}{120} \\ \dfrac{1}{12} \end{pmatrix}.$$

Again we are looking for a matrix T which has a unique eigenvalue and satisfies conditions P1 and P2.

Since T is a 3×3 matrix, it has a unique eigenvalue if and only if

$$det(T) = \gamma^3, \quad tr(T) = 3\gamma, \quad tr(T^2) = 3\gamma^2. \tag{3.1}$$

Then, using Theorem 3.1 in [14], T satisfies condition P1 if and only if

$$\gamma = \sqrt[3]{det(\bar{A})} = \sqrt[3]{1/120}, \quad det(\bar{A} - T) = 0, \quad tr(\bar{A}^{-1}T) = 3. \tag{3.2}$$

Moreover, condition P2 implies $det(\bar{A} - T) = 0$ and so, to satisfy the requirements imposed to matrix T, seven conditions on its coefficients must be accomplished. Such a matrix is not unique and we have two free parameters available. Following the ideas in [14], these free parameters will be used to minimize the spectral radius of $M(z)$ over the nonpositive real axis, that is, to minimize the quantity $\max\{\rho(M(z))\,|z \in (-\infty, 0)\}$, and also to minimize the Frobenius norm $||I - \bar{A}^{-1}T||_F$.

Let us suppose that (3.1) and (3.2) hold. Since $det(T - \bar{A}) = 0$, one of the three eigenvalues of $M(z)$ vanishes and the other two eigenvalues $\{\mu_1(z), \mu_2(z)\}$ are the roots of the second-degree equation

$$\mu^2(z) - b(z)\mu(z) + c(z) = 0, \tag{3.3}$$

where

$$b(z) = tr(M(z)), \quad \text{and} \quad c(z) = \frac{det(I - z\bar{A})}{(1 - \gamma z)^3} + b(z) - 1.$$

Defining $\Lambda = (tr^2(\bar{A}) - tr(\bar{A}^2))/2$, then $det(I - z\bar{A}) = 1 - tr(\bar{A})z + \Lambda z^2 - det(\bar{A})z^3$, and taking into account the asymptotic expansions

$$\begin{aligned} tr(M(z)) &= ztr(\bar{A} - T) + \mathcal{O}(z^2), \quad (z \to 0) \\ tr(M(z)) &= tr(I - T^{-1}\bar{A}) + \frac{1}{z}tr(T^{-1} - T^{-2}\bar{A}) + \mathcal{O}(\frac{1}{z^2}) \quad (z \to \infty), \end{aligned}$$

and recalling that $tr(T^{-1}) = 3/\gamma$, the coefficients of the eigenvalue equation (3.3) can be expressed as rational functions of z as

$$b(z) = \frac{z}{(1 - \gamma z)^3}(\xi_1 z + tr(\bar{A}) - 3\gamma), \qquad c(z) = \frac{z^2}{(1 - \gamma z)^3}\xi_2$$

where $\xi_1 = \gamma^3\delta - 3\gamma^2$, $\xi_2 = \gamma^3\delta + \Lambda - 6\gamma^2$, and $\delta = tr(T^{-2}\bar{A})$.

In conclusion, the coefficients of the eigenvalue equation (3.3) only depend on the matrix $\bar{A}$ and on the parameter δ. Since we are trying to minimize $\rho(M(z))$ over the nonpositive real axis, we have to find a value δ_0 so that the quantity $\Gamma(\delta) = \max\{\rho(M(z)) : z \leq 0\}$ is minimum. Then, the matrix T will be chosen to satisfy $tr(T^{-2}\bar{A}) = \delta_0$.

Lemma 4 *For four-stage sixth-order RK Lobatto IIIA, there exists a unique value*

$$\delta_0 = 17.82460709187153934...$$

such that $\Gamma(\delta_0) \leq \Gamma(\delta)$ *for all* $\delta \neq \delta_0$*, that is, for* $tr(T^{-2}\bar{A}) = \delta_0$ *the quantity* $\max\{\rho(M(z)) : z \leq 0\}$ *is minimum.*

PROOF. . First, let us suppose that the coefficient $c(z)$ vanishes for all z, that is, $\xi_2 = 0$ or equivalently $\delta = (-\Lambda + 6\gamma^2)/\gamma^3$. Then 0 is a double eigenvalue of $M(z)$, $\xi_1 = -\Lambda + 3\gamma^2$ and $\rho(M(z)) = |b(z)|$, which clearly attains a maximum:

$$\max\{\rho(M(z)) : z \leq 0\} = r_0 = 0.1376449122...$$

If $c(z)$ does not vanish, let us suppose that the maximum of $\rho(M(z))$ is smaller than the value r_0 obtained above. Then, there exists $0 < r < r_0$ such that $|\mu_i(z)| < r$, for all $z \leq 0$, $i = 1, 2$. Making the change of variables

$$t = \frac{1}{(1 - \gamma z)} \in (0, 1), \text{ and } x(t) = \frac{\mu(t)}{r}, \; \forall t \in (0, 1)$$

it is not difficult to prove that the eigenvalue equation (3.3) is equivalent to

$$r^2 x^2(t) - rB(t)x(t) + C(t) = 0, \quad t \in (0, 1) \tag{3.4}$$

with $B(t) = -(B_1 t + B_2)(1 - t)t$ and $C(t) = B_3(1 - t)^2 t$ where

$$B_1 = \gamma\delta + \frac{tr(\bar{A})}{\gamma} - 6, \quad B_2 = -\gamma\delta + 3, \quad \text{and } B_3 = \gamma\delta + \frac{\Lambda}{\gamma^2} - 6.$$

Thus, for $z \leq 0$ the roots $\mu_i(z)$ of equation (3.3) will have modulus smaller than r if and only if the roots $x_1(t)$ and $x_2(t)$ of (3.4) satisfy $|x_i(t)| < 1, \; \forall t \in (0, 1)$.

Applying the Schur's Criterion this condition is satisfied if and only if for all $t \in (0, 1)$

$$|B_3| < \frac{r^2}{(1 - t)^2 t^2}, \qquad r|B(t)||r^2 - C(t)| < r^4 - |C(t)|^2.$$

that are equivalent to

$$|B_3| < 27r^2/4, \tag{3.5}$$

$$r|B(t)| < r^2 + C(t). \tag{3.6}$$

Now, taking into account that $B_1 = B_3 + K_1, \quad, B_2 = -B_3 + K_2$, with

$$K_1 = \frac{tr(\bar{A})}{\gamma} - \frac{\Lambda}{\gamma^2}, \quad K_2 = \frac{\Lambda}{\gamma^2} - 3,$$

the inequality (3.6) is equivalent to

$$B_3 > \max_{t\in(0,1)} \{P_1(r,t), P_2(r,t)\}$$

where

$$P_1(r,t) = \frac{r}{(1+r)(1-t)}\left[(K_1 t + K_2) - \frac{r}{(1-t)t}\right]$$
$$P_2(r,t) = \frac{r}{(1-r)(1-t)}\left[-(K_1 t + K_2) - \frac{r}{(1-t)t}\right]$$

Since for each $r \in (0, r_0)$, $\lim_{t\to 0^+} P_1(r,t) = \lim_{t\to 0^+} P_2(r,t) = \lim_{t\to 1^-} P_1(r,t) = \lim_{t\to 1^-} P_2(r,t) = -\infty$, and $\max_{t\in(0,1)} P_2(r,t) > \max_{t\in(0,1)} P_1(r,t)$, (3.6) is equivalent to

$$B_3 > \max_{t\in(0,1)} P_2(r,t) \tag{3.7}$$

For each δ, or equivalently, for each B_3, there is a minimum value $r(B_3)$ such that (3.5) and (3.7) are satisfied for all $t \in (0,1)$. We are looking for a value B_3^* such that $r(B_3^*) = \min_{B_3} r(B_3)$. Such a value will be obtained by solving the equations

$$27r^2/4 = P_2(r,t)\ , \qquad \frac{dP_2(r,t)}{dt} = 0$$

With the help of an algebraic manipulator, it can be shown that these equations admit in $r \in (0, r_0)$ a solution only for $r = 0.08312670...$ and the corresponding value of δ is $\delta_0 = 17.8246070918715393...$ ♢

In this case, the maximum value of the spectral radious of $M(z)$, $z \in I\!R^-$, is attained at the point $z \simeq -2.6576$ and for the imaginary axis, the maximum is attained at $z \simeq 6.0907322i$, giving a value

$$\max\{\rho(M(iy)), y \in I\!R\} \simeq 0.253668$$

After the above considerations, we are looking for a matrix T that satisfies conditions (3.1) and (3.2) and P2 and $||I - \bar{A}^{-1}T||_F$ is minimum and for which

$\max_{z\leq 0} \rho(M(z))$ is minimum, or equivalently, a 3×3 matrix T such that

$$
\begin{aligned}
&(C1)\ det(T) = \gamma^3 = det(\bar{A}) = 1/120,\\
&(C2)\ tr(T) = 3\gamma,\\
&(C3)\ tr(T^2) = 3\gamma^2,\\
&(C4)\ tr(\bar{A}^{-1}T) = 3,\\
&(C5)\ tr(T^{-2}\bar{A}) = \delta_0,\\
&(C6)\ (0,0,1)\,(I-\bar{A}^{-1}T) = (0,0,0),\\
&(C7)\ ||I-\bar{A}^{-1}T||_F \text{ is as small as possible.}
\end{aligned}
$$

On the other hand, since T has an unique eigenvalue γ, its characteristic polynomial is given by $p(\lambda) = det(\lambda I - T) = (\lambda-\gamma)^3$, and then

$$p(T) = T^3 - 3\gamma T^2 + 3\gamma^2 T - \gamma^3 I = 0.$$

Thus, (C5) can be replaced by

$$(C5')\ tr(\bar{A}T) = \gamma^3\delta_0 + 3\gamma tr(\bar{A}) - 9\gamma^2.$$

To simplify the problem, instead of the matrix T we can obtain equivalently the matrix $P = I - \bar{A}^{-1}T$. It is not difficult to prove that T satisfies C1 to C7 if and only if

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$
\begin{aligned}
&(D1)\ tr(P) = 0,\\
&(D2)\ det(I-P) = 1,\\
&(D3)\ tr(\bar{A}P) = tr(\bar{A}) - 3\gamma,\\
&(D4)\ tr(\bar{A}^2P) = tr(\bar{A}^2) - K,\\
&(D5)\ tr((\bar{A}P)^2) = tr(\bar{A}^2) + 3\gamma^2 - 2K,\\
&(D6)\ ||P||_F \text{ is minimum,}
\end{aligned}
$$

where $K = \gamma^3\delta_0 + 3\gamma tr(\bar{A}) - 9\gamma^2$.

Clearly, conditions $(D1)$,$(D3)$ and $(D4)$ are linear in the coefficients of P. Thus, all these conditions are reduced to two nonlinear equations and one condition of minimization $(D6)$.

The construction of matrix P has been carried out numerically by means of a minimization package and after extensive computations we have obtained the matrix

$$P = \begin{pmatrix} -0.01832522921162447 & -0.1240694135537153 & 0.02056578515084587 \\ 0.002706662472561666 & 0.01832522921162447 & -0.1475471666586898 \\ 0 & 0 & 0 \end{pmatrix},$$

that yields a local minimum value of $||P||_F = 0.1956151613768685$.

The corresponding matrix $T = \bar{A}(I-P)$ of the scheme is given by

$$T = \begin{pmatrix} 0.1932674949117222 & -0.009750106539280771 & 0.001396313165263860 \\ 0.4582165795963249 & 0.2787104623506828 & -0.002745269684755689 \\ 0.4231744028079428 & 0.4607267434758711 & 0.1362422422949350 \end{pmatrix}$$

In this case, it is also possible to obtain a simple matrix

$$S = \begin{pmatrix} 1 & -0.0013313944847890405 & -0.021160953394204083 \\ 0 & 1 & 0.16376865269504141 \\ 0 & 0 & 1 \end{pmatrix}$$

that transforms T to triangular form and

$$L = I - \gamma S^{-1} T^{-1} S = \begin{pmatrix} 0 & 0 & 0 \\ 1.91828820257772989 & 0 & 0 \\ -2.26670285249783297 & 2.26972072817430417 & 0 \end{pmatrix}$$

In Table 2 we present again the maximum values θ_k for which the k–th iteration, $k = 1, \ldots, 6$, is A(θ_k)–stable, and also the minimum α_k for which $|R_k(z)| \leq 1$ for all Re $z \leq -\alpha_k$. For this method the first iteration is A–stable and again for $k \geq 2$ the stability regions of the successive iterations approximate to the negative complex half–plane as k increases.

In this case, once again we have found a positive–definite diagonal matrix $D = \text{diag}(1, 0.34516405, 0.081325562)$ such that $\nu = 2 \parallel N^{-T} D(T - \bar{A}) N^{-1} \parallel_2 = 0.71036694...$ is minimum.

Table 2
Stability regions of 4–stage Lobatto IIIA.

k	θ_k	γ_k
1	90.00°	0.00
2	89.3492085759°	0.3737593155565
3	89.9272843260°	0.0161715124278
4	89.9784185933°	0.0036772308997
5	89.9909818221°	0.0013306151786
6	89.9967005782°	0.0004405981063

4 Numerical experiments

In this section we present some numerical experiments with two purposes: Firstly, to show that the new iterative schemes of the above section are for many stiff problems a more efficient alternative that simplified Newton type methods in the solution of the algebraic equations of stages. Secondly, to show that the Lobatto IIIA formulas behave very satisfactorily when they are applied to stiff systems. To this end we have implemented a variable stepsize code, estimating the local error by the classical extrapolation technique. The stepsize changing policy has been based on the usual formula $h_{n+1} = \theta(\text{Tol}/\|\text{Est}\|)^{1/7} h_n$, after an accepted step, but halving the stepsize when the local error estimate Est is greater than the prescribed tolerance Tol.

To solve the implicit equations (1.2) we proceed at each step from t_n to $t_{n+1} = t_n + h$ as follows:

- We evaluate the Jacobian matrix $J = (\partial f/\partial y)(t_n, y_n)$ at each gridpoint.
- We compute initial approximations $Y_j^{(0)}$, $j = 2, 3, 4$, by $Y_j^{(0)} = P_n(t_n + c_j h)$ where $P_n(t)$ is the Lagrange polynomial that interpolates the values Y_j, $j = 1, \dots, 4$ of the previous step.
- We iterate the scheme until

$$\|Y_i^{(l)} - Y_i^{(l-1)}\|_\infty \leq 0.01 \text{ Tol}, \qquad i = 2, 3, 4$$

- If the convergence condition has not been attained after ten iterations or if for some r some of the quotients

$$\theta_{ir} = \|Y_i^{(r)} - Y_i^{(r-1)}\|_\infty / \|Y_i^{(r-1)} - Y_i^{(r-2)}\|_\infty$$

  is greater than one, we consider that there is no convergence and in this case we restart the step with stepsize $h/2$.

For the sake of brevity we present here only the results obtained with the three following problems taken from related literature

Problem 1.- The Oregonator (see e. g. [16] pp. 156).

$$\begin{aligned}
y_1' &= 77.27(y_2 + y_1(1 - 8.375 \times 10^{-6} y_1 - y_2)),\\
y_2' &= (y_3 - (1 + y_1)y_2)/77.27\\
y_3' &= 0.161(y_1 - y_3)\\
y_1(0) &= 1,\ y_2(0) = 2,\ y_3(0) = 3, \quad t \in [0, 3600]
\end{aligned}$$

Problem 2.- The Van der Pol oscillator (see e. g. [16] pp. 157)

$$\begin{aligned}
y_1' &= y_2\\
y_2' &= ((1 - y_1^2)y_2 - y_1)/\varepsilon, \qquad \varepsilon = 10^{-6}\\
y_1(0) &= 2, y_2(0) = 0, t \in [0, 20],
\end{aligned}$$

Problem 3.- The CUSP problem (see e. g. [16] pp. 163).

$$\begin{aligned}
y_i' &= -(1/\varepsilon)\,(y_i^3 + a_i y_i + b_i) + D(y_{i-1} - 2y_i + y_{i+1}),\\
a_i' &= b_i + 0.07v_i + D(a_{i-1} - 2a_i + a_{i+1}) \qquad i = 1, \ldots, N,\\
b_i' &= (1 - a_i^2)b_i - a_i - 0.4y_i + 0.035v_i + D(b_{i-1} - 2b_i + b_{i+1})
\end{aligned}$$

where

$$v_i = \frac{u_i}{u_i + 1}, \quad u_i = (y_i - 0.7)(y_i - 1.3), \quad D = \frac{N^2}{100}, \varepsilon = 10^{-8}, \quad N = 32$$

and

$$y_0 = y_N, \quad a_0 = a_N, \quad b_0 = b_N, \quad y_{N+1} = y_1, \quad a_{N+1} = a_1, \quad b_{N+1} = b_1.$$

We have taken the initial values

$$y_i(0) = 0,\ a_i(0) = -2\cos(2i\pi/N),\ b_i(0) = 2\sin(2i\pi/N), \quad i = 1, \ldots, N$$

and $t \in [0, 1.1]$.

In order to compare the efficiency of our new iterative scheme, we have also implemented the code solving the stage equations by a simplified Newton scheme, implemented following the ideas in [16], pp.128–135, by diagonalizing the Runge–Kutta submatrix $\bar{A}$ so that the full Jacobian matrix splits into one $m \times m$ block and a $2m \times 2m$ block. Thus, while the standard simplified Newton method requires the factorization of the $3m \times 3m$ matrix $I - h(A \otimes J)$, that needs about $(3m)^3/3$ operations, with this technique we have to factorize at each step an $m \times m$ real matrix and an $m \times m$ complex matrix and the amount of work for the LU factorizations becomes approximately $5m^3/3$.

Note that our single–Newton scheme requires only the factorization of the $m \times m$ matrix $I - h\gamma J$ with about $m^3/3$ arithmetic operations. However, the convergence rate of the proposed iterations is in general $\mathcal{O}(1)$ while for the simplified Newton is $\mathcal{O}(h)$ and therefore this method is expected to need less iterations to converge than single–Newton method when the stepsize h is reduced.

On the other hand, to test the efficiency of the Lobatto IIIA method, we have compared our code based on this formulae with three well known codes for the solution of stiff systems. First, we have considered EPISODE [7], the variable–stepsize variable–order code based on BDF formulas of orders 1 to 5. Second, we have considered STRIDE [3], based on the so–called Singly–Implicit RK methods, that also uses a variable–stepsize variable–order strategy being able to use methods with orders from order 1 to 12. On the third place we have considered RADAU5, based on the 5th–order Radau IIA formula, developed by Hairer and Wanner [16]. This code uses a modification of simplified–Newton method to solve the implicit stage equations, based on complex arithmetic.

In addition, we have also compared the code with a code based on the 5th–order Radau IIA method (we will refer it as sn-Radau), developed by González–Pinto et al. [14], and that use for the solution of the implicit stage equations iterative schemes of the type (2.4) proposed in this paper.

For each code we have integrated each problem with tolerances from 10^{-4} to 10^{-10}, computing the maximum global error GE at the end of the integration interval as well as the required CPU time measured in seconds. In Figures 4 to 4 we have plotted the points (log(GE), CPU) obtained for each code and tolerance displayed as a polygonal line.

Concerning the efficiency of the proposed iterative scheme, we can observe in the figures that the Lobatto IIIA code based on the simplified Newton scheme is as efficient as the one based on the single–Newton scheme for problems 1 and 2, but the last scheme is much more efficient than the other for CUSP problem. Since both codes only differ on the iterative scheme, they achieve similar accuracy giving practically the same number of steps and hence, for the LU factorizations, simplified Newton will require about 5 times the work required by single–Newton scheme. The codes will probably differ on the number of iterations needed to converge and in consequence, on the number of linear

Table 3
Lobatto IIIA. Iterations average per step for Van der Pol problem.

Tol	Simplified Newton			Single–Newton		
Tol	First step	Second step	Double step	First step	Second step	Double step
10^{-4}	6.6	7.5	2.5	8.0	8.4	2.9
10^{-5}	6.4	7.3	2.6	7.9	8.2	3.2
10^{-6}	5.8	6.6	2.8	7.4	8.1	3.3
10^{-7}	5.4	6.3	2.8	7.2	7.7	3.7
10^{-8}	4.9	5.7	3.0	6.9	7.3	3.9
10^{-9}	4.6	5.2	3.1	6.7	7.0	4.1
10^{-10}	4.4	4.9	3.1	6.6	6.8	4.3

systems they have to solve and on the number of evaluations of the derivative function f. Let us note that single–Newton has to solve at each iteration 3 $m \times m$ linear systems with the same previously factorized matrix (about $3m^2$ operations) and simplified Newton has to solve at each iteration one $m \times m$ linear system and one $m \times m$ complex linear system(about $5m^2$ operations in total). In general, simplified Newton scheme needs less iterations to converge than single–Newton and, for low dimension problems, this compensates the computational effort done in the LU factorizations and the subsequent solution of linear systems and both codes have similar performance. However, for large dimension problems like CUSP, the computational cost is mainly due to the work involved in the LU factorizations and simplified Newton method is clearly less efficient than single–Newton. In Tables 3 and 4 we include for the Van der Pol and CUSP problems, and tolerances from 10^{-4} to 10^{-10}, the number of iterations per step used by both schemes at the first extrapolation step (from t_n to $t_n + h$), the second extrapolation step (from $t_n + h$ to $t_n + 2h$) and the double step (from t_n to $t_n + 2h$).

Concerning the efficiency of the Lobatto IIIA formula, from our experiments we have appreciated in general that the code based on the Lobatto IIIA has a fairly good behaviour, making it competitive with the other tested codes, and in some particular cases it has proved to be superior. It usually takes less steps than the other codes maintaining at the same time the required precision. Thus, the code has been particularly efficient with the Van der Pol and CUSP problems, which makes us expect that it will be very suitable for singularly perturbed problems. Further general comments on the other codes are:

EPISODE is usually very fast. It normally needs many steps but the computational cost per step is smaller than in RK codes. However, it sometimes

Table 4
Lobatto IIIA. Iterations average per step for CUSP problem.

Tol	Simplified Newton			Single–Newton		
Tol	First step	Second step	Double step	First step	Second step	Double step
10^{-4}	6.3	5.4	1.5	7.1	5.6	1.7
10^{-5}	6.7	6.1	1.7	7.6	6.7	1.5
10^{-6}	7.1	6.8	1.7	8.1	7.1	1.8
10^{-7}	6.7	7.1	2.2	8.0	7.7	2.4
10^{-8}	6.4	7.1	2.7	7.7	8.0	2.9
10^{-9}	5.9	6.6	2.8	7.3	7.5	3.3
10^{-10}	5.5	6.3	2.9	7.1	7.4	3.5

lacks precision in the computed solutions.

STRIDE usually does fewer LU matrix factorizations and Jacobian matrix evaluations than the other codes. However it is less efficient for low dimension problems. This can be explained taking into account that is usually takes more steps than the other RK codes. Moreover, since an sth–order SIRK method has s stages, it has to solve at each iteration s linear systems with the same previously factorized matrix and, for example a sixth order method requires about $6m^2$ operations while the sixth–order Lobatto IIIA method requires $3m^2$ operations. Also, a similarity transformation [3] associated to a 6th–order SIRK method require 6^2m operations while for Lobatto IIIA this quantity is 3^2m. Let us also note that for problems 1 and 2, with $m = 3$ and $m = 2$ respectively, the influence of the work involved on the similarity transformations on the total CPU time is relevant.

Concerning RADAU5, it presents a good behaviour for all tested problems, but it sometimes looses performance when the tolerance decreases, because it increases considerably the number of integration steps.

The other code that also use a single–Newton scheme presents in general a good behaviour. However it is remarkable the fact that for stringent tolerances Lobatto IIIA gives much fewer steps than sn–Radau. This could be explained taking into account that it has order 5 while Lobatto IIIA has order 6. It is also plausible that the reason of this behaviour is that Lobatto IIIA formula has stage–order 4 while Radau have stage–order 3.

Particularly interesting is the behaviour of the codes when they integrate large dimension problems like CUSP. In this particular case the computational cost is mainly due to the required number of LU matrix factorizations which is related to the number of steps. In Table 5 we give for each tolerance and code

Table 5
CUSP problem. Number of steps

Tol	LobIIIA Single–Newton	LobIIIA Simpl. Newton	STRIDE	RADAU5	EPISODE	sn-Radau
10^{-4}	208	202	399	115	310	228
10^{-5}	230	220	546	196	469	250
10^{-6}	262	256	691	316	698	304
10^{-7}	318	304	974	527	930	366
10^{-8}	382	356	1126	901	1239	464
10^{-9}	456	448	1305	1562	1726	608
10^{-10}	582	560	1788	2764	2433	856

Table 6
CUSP problem. Number of LU factorizations

Tol	LobIIIA Single–Newton	LobIIIA Simpl. Newton	STRIDE	RADAU5	EPISODE	sn-Radau
10^{-4}	250	236	178	180	220	270
10^{-5}	262	250	200	252	465	289
10^{-6}	297	285	217	372	738	346
10^{-7}	347	331	306	535	292	402
10^{-8}	419	385	325	763	627	498
10^{-9}	487	475	387	1079	402	639
10^{-10}	610	580	810	1582	9030	893

the number of steps and in Table 6 we include the number of LU factorizations given with CUSP problem. For the cases of RADAU5 and Lobatto IIIA with simplified Newton scheme, each LU factorization in the table amounts the factorization of a real $m \times m$ matrix plus the factorization of a complex $m \times m$ matrix. Hence, to compare with the other columns, it would be appropriate to introduce a factor of approximately 5.

The data in these tables agree with the results observed in figure 4. The good performance of EPISODE and Lobatto IIIA for low tolerances is in accordance with the low number of LU factorizations they require. Note at this point that the Lobatto IIIA code evaluates the Jacobian matrix at each gridpoint and therefore does one LU factorization at each step. A tuning of the code reusing the Jacobian matrix when it is possible, such as it is done in EPISODE or STRIDE for example, would reduce this number and consequently the efficiency of the code could be improved.

In conclusion, from our numerical experiments we can conclude that, even though they are not B–stable, Lobatto IIIA methods can be very efficient for the numerical solution of nonlinear stiff systems. Also, the iterative scheme developed is reliable and efficient and it makes the code competitive with other known codes.

Acknowledgements

We want to thank Professor M. Calvo for his careful reading of the manuscript and his useful suggestions and the anonymous referees for their valuable comments.

References

[1] Ascher, U., On symmetric schemes and differential–algebraic equations *SIAM J. Sci. Stat. Comput.* **10** (5) (1989) 937–949.

[2] Bickart, T. A., An efficient solution process for implicit Runge–Kutta methods *SIAM J. Numer Anal.* **14** (1977) 1022–1027.

[3] K. Burrage, J.C. Butcher and F.H. Chipman, An implementation of singly–implicit Runge–Kutta methods, *BIT***20** (1980) 326–340.

[4] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, *BIT* **16** (1976) 237–240.

[5] J.C. Butcher, Some implementation schemes for implicit Runge–Kutta methods, *Proceedings of the Dundee Conference on Numerical Analysis, 1979. Lecture Notes in Mathematics* **773** (1980) 12–24.

[6] J.C. Butcher, *The numerical analysis of ordinary differential equations* (John Wiley & Sons, Chichester, 1987).

[7] G.D. Byrne, and A.C. Hindmarsh, A polyalgorithm for the numerical solution of ordinary differential equations, *ACM Trans. Math. Soft.* **1** (1975) 71–96.

[8] M. Calvo, S. González–Pinto and J.I. Montijano, Runge–Kutta methods for the numerical solution of stiff semilinear initial value problems (to be published, 1996).

[9] M. Calvo, S. González–Pinto and J.I. Montijano, Convergence of Runge–Kutta methods for stiff nonlinear differential equations (to be published, 1996).

[10] G.J. Cooper and J.C. Butcher, An iteration scheme for implicit Runge-Kutta methods, *IMA J. of Num. Anal.* **3**(1983) 127-140.

[11] G.J. Cooper and R. Vignesvaran, A scheme for the implementation of implicit Runge-Kutta methods, *Computing* **45** (1990) 321-332.

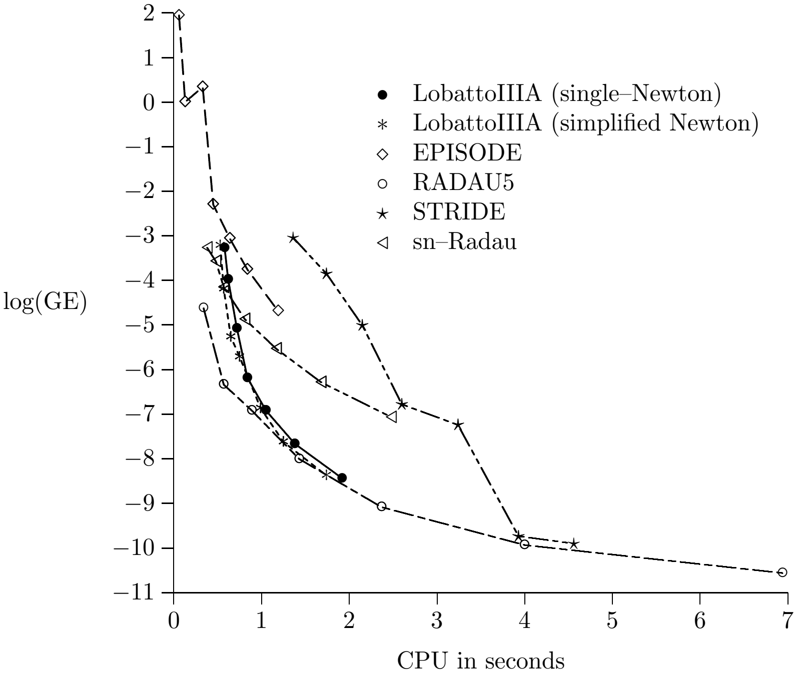

Fig. 1. Oregonator problem.

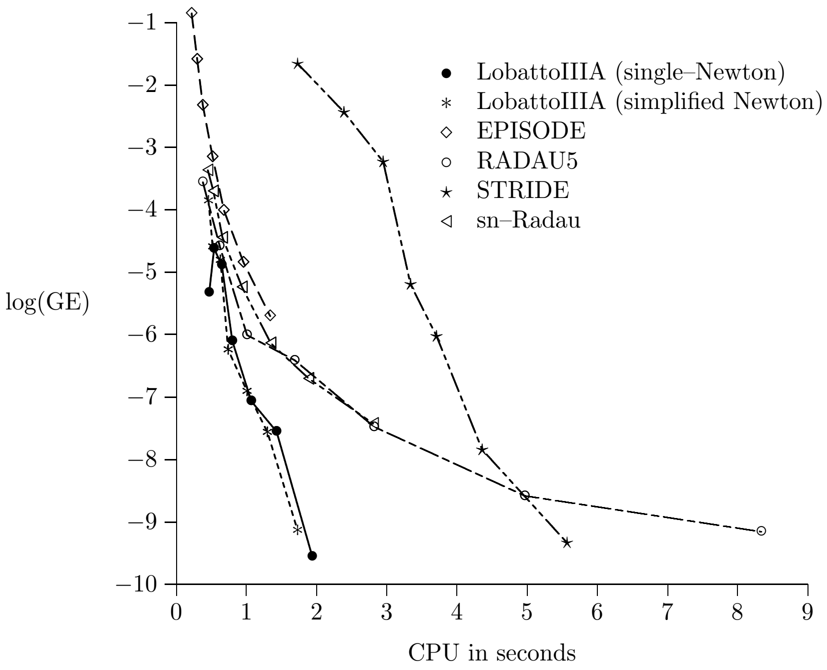

Fig. 2. Van der Pol problem.

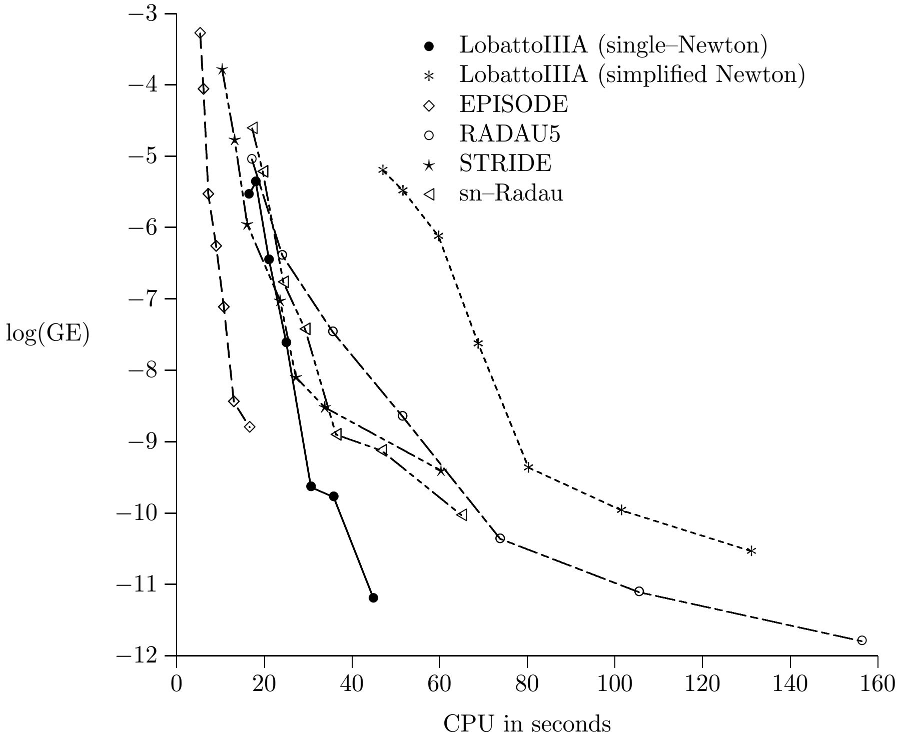

Fig. 3. CUSP problem.

[12] K. Dekker and J.G. Verwer, *Stability of Runge–Kutta methods for stiff nonlinear differential equations* (North Holland, Amsterdam, 1984).

[13] S. Gonzalez-Pinto, C. Gonzalez-Concepcion & J.I. Montijano, Iterative schemes for Gauss methods, *Computers Math. Applic.* **27** (7) (1994) 67–81.

[14] S. Gonzalez-Pinto, J.I. Montijano & L. Rández, Iterative schemes for three–stage implicit Runge–Kutta methods, *Appl. Numer. Math.* **17** (1995) 363-382.

[15] E. Hairer, Ch. Lubich and M. Roche, The numerical solution of differential algebraic systems by Runge–Kutta methods. *Lecture Notes in Math.* **1049**, Springer Verlag 1989.

[16] Hairer, E., Wanner, G. *Solving Ordinary Differential Equations II* (Springer–Verlag, 1991).

[17] Lambert, J.D. *Numerical Methods for Ordinary Differential Systems* (Wiley, 1991).

[18] A. Prothero, A. Robinson, On the stability and accuracy of one–step methods for solving stiff systems of ordinary differential equations. *Math. of Comp.* **28** (1974) 145-162.