RESEARCH ARTICLE | SEPTEMBER 06 2017

Mono-implicit Runge Kutta schemes for singularly perturbed delay differential equations FREE

Fathalla A. Rihan; Nasser S. Al-Salti

AIP Conf. Proc. 1872, 020001 (2017)
https://doi.org/10.1063/1.4996658

Mono-Implicit Runge Kutta Schemes for Singularly Perturbed Delay Differential Equations

Fathalla A. Rihan[1,a,b] and Nasser S. Al-Salti[2]

[1]*Department of Mathematical Sciences, College of Science, United Arab Emirates University, Al-Ain, 15551, UAE*
[2]*Department of Mathematics and Statistics, College of Science, Sultan Qaboos University, P.O. Box 36, P.C. 123 Al-Khoudh, Oman*

[a)]Corresponding author: frihan@uaeu.ac.ae
[b)]URL: http://faculty.uaeu.ac.ae/frihan/

Abstract. In this paper, we adapt Mono-Implicit Runge-Kutta schemes for numerical approximations of singularly perturbed delay differential equations. The schemes are developed to reduce the computational cost of the fully implicit method which combine the accuracy of implicit method and efficient implementation. Numerical stability properties of the schemes are investigated. Numerical simulations are provided to show the effectiveness of the method for both stiff and non-stiff initial value problems.

Keywords: Mono-implicit RK methods; Singularly perturbed problems; Stability; Stiffness; Time-lags

2010 MSC: 65L06; 34K28; 65L20; 65L05

Introduction

It is well known that delay differential equations have a richer mathematical framework, and better consistency with real observations, compared with corresponding models without memory or after-effects. However, many problems that modeled by delay differential equations – especially in the study of chemical kinetics, or immune system interactions – are 'stiff', in the sense that they have properties that make them slow and expensive to solve using explicit numerical methods. The history of stiff IVPs goes back more than 60 years, when the first identification of stiff equations as a special class of problems has been given by chemists Curtiss and Hirschfelder in 1952 [1]. Since then, stiff equations presented serious difficulties and were hard to solve, both in chemical problems (reaction kinetics) and increasingly in other areas (electrical engineering, mechanical engineering, etc) until around seventies century when a variety of methods for solving such problems began to appear in the literature [2].

What makes a stiff problem stiff? The nature of the problems that leads to stiffness is the existence of physical phenomena with very different speeds (time constants) so that, while we may be interested in relative slow aspects of the model, there are features of the model that could change very rapidly. In the literature of ODEs, various definitions are seen for the stiffness [3, 4, 5], one being somewhat more precise than another. The essence of stiffness is that the solution to be computed is slowly varying but that perturbations exist which are rapidly damped. The presence of such perturbations complicates the numerical computation of the slowly varying solution. The problem of stiffness also occurs in DDEs [6, 7, 8], but the situation is more complicated than for ODEs because the existence of rapidly and slowly varying may not imply stiffness in DDEs [9].

The *stiffness* with DDEs models is characterized by phenomena such as: strong contractivity of neighbouring solutions, multiple time scales (fast transient phases) and the fact that explicit numerical integrators are not able to reproduce a correct approximation of the solution in an efficient way. Another definition: stiff differential equation is an equation for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. In other words, the step size is restricted by stability and not by accuracy considerations [10, 11]. In [12], stiff equations are defined to be those equations where implicit methods perform tremendously better than explicit ones.

Mathematical Methods and Computational Techniques in Science and Engineering
AIP Conf. Proc. 1872, 020001-1–020001-15; doi: 10.1063/1.4996658
Published by AIP Publishing. 978-0-7354-1552-2/$30.00

The efficient use of reliable numerical methods, which are generally based on implicit formulae [7, 8]), for dealing with stiff models is essential. Fully implicit Radau II-A method has been used by Guglielmi and Hairer [7], in code RADAR5[1] for stiff problems, including differential-algebraic and neutral DDEs with constant or state-dependent delay. *Linear multi-step methods* (LMMs) has also been used for stiff problem [6, 12]. However, the order of an A-stable LMM can not exceed 2. Also, the fully implicit method involves a degree of sophistication that are not necessarily available to nonspecialists[2]. Moreover, the user may not know whether his problem is stiff or not, where the character of the problem is not known in advance: Many problems may be stiff in some intervals and non-stiff in others. Plus, integrating of non-stiff problems with the stiff method is very expensive, whereas non-stiff methods are much better suited for this purpose. Therefore, we need an efficient technique to be suitable for both stiff and non-stiff problems. Explicit methods have lower computational costs, but with lower accuracy, compared to implicit methods. If the problem can be solved with comparable accuracy with both explicit and implicit methods then explicit method is the choice. But what will happen if explicit methods have higher computational cost and lower accuracy results or even fail to get a solution as in stiff problems? In this case, we have no choice but to use implicit methods [13].

Singulary perturbed problems are examples of stiff problems that occur in dynamical systems with nonsmooth signals or shock dynamics[3] [15]. This paper continues research initiated in [16]. We suggest unconditionally stable methods based on mono-implicit Runge-Kutta schemes, which are not fully implicit nor explicit, for numerical approximations of stiff singularly perturbed delay differential equations. The schemes are developed to reduce the computational cost of the fully implicit method. The paper is organized as follows: In Section 2, we introduce issues related to stiff DDEs. In Section 3, we provide mono-implicit Runge-Kutta method for numerical treatments of stiff DDEs. In Sections 4 & 5, we study the numerical stability properties of the method and numerical simulations, respectively. We finally conclude in Section 6.

Singular Perturbed Delay Differential Equations

Many dynamical systems are described by an IVP of DDEs [17, 18] of the form

$$y'(t) = f(t, y(t), y(t-\tau)), \quad t \in [t_0, T], \tag{1}$$

$$y(t) = \psi(t), \quad t \in [-\tau, t_0], \tag{2}$$

where the time-lag τ is assumed to be non-negative parameter. It could be constant, or variable as a function of t such that $0 \leq \tau(t) \leq \tau^*$, where τ^* is a constant. Because of the delay term, it is no longer sufficient to supply an initial value, at time $t = t_0$, to completely define the problem, but an initial function $\psi(t)$ defined in interval $[-\tau, t_0]$. The function f is assumed to be sufficiently smooth with respect to its arguments, and $\psi(t)$ is also assumed to be continuous. Let $\mathbb{C}^n$ be a complex Banach space with the norm $\|.\|$, $f : [t_0, +\infty] \times \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}^n$. Assume also that the function $f(t, u, v)$ satisfies the classical Lipschitz condition in the second and third variables u and v i.e.

$$\|f(t, u, v) - f(t, \overline{u}, \overline{v})\| \leq L(\|u - \overline{u}\| + \|v - \overline{v}\|), \quad L \in \mathbb{R}^+. \tag{3}$$

For non-stiff (classical) problems, the classical Lipschitz constant L in some domain $D \subset [0, T] \times \mathbb{C}^n$ is an important parameter not only in estimates for the sensitivity of the solution $y(t)$ with respect to data perturbations (condition estimates) but also in convergence analysis of numerical methods for solving these problems [12]. For stiff problems, however, L is typically very large, as the classical Lipschitz constant only measures variations of f but does not take into account if the direction field corresponding to the right-hand side f of (1) is diverging or converging [19]. The concept of one-sided Lipschitz constant for DDEs is given by the following Remark.

Remark 1 *For a stiff problem, there exists a unique smooth solution $y(t)$, such that f satisfies the condition (3) and*

$$\Re\langle f(t, u, v)\ ,\ f(t, \overline{u}, \overline{v})\rangle \leq M\Big(\|u - \overline{u}\|^2 + \|v - \overline{v}\|^2\Big), \tag{4}$$

[1] The code RADAR5 is developed in FORTRAN-90 and is based on an adaptation of the 3-stage Radau IIA method.

[2] The implicit schemes need to solve an auxiliary nonlinear system of equations in each step. These systems are approximated via Newton-type iterative methods, with a good initial guess inside of the ball of convergence of the iterative scheme. For certain stiff stiff IVPs, the order of convergence of implicit schemes can be much lower than for nonstiff problems.

[3] A key feature of the shock dynamics of nonlinear systems is the phenomenon of sudden change, when the periodic orbit reaches the barrier at zero speed, but nonzero acceleration [14].

where $\langle . , . \rangle$ denotes a given inner product. The one-sided Lipschitz constant M, represents the sensitivity of solution with respect to this initial perturbation, exists such that $M \ll L$ ($L \gg 0$) and possibly $M \leq 0$ applies.

Delay differential equations in which a small parameter $\varepsilon > 0$ multiplies the highest derivative are commonly termed as *singularly perturbed* delay differential equations [20]. This dimensionless parameter ε usually arises, in the differential equations, when one physical process occurs much more slowly than another, or when one geometrical length in the problem is much shorter than another. Examples occur in the so called human pupil-light reflex [21] and in a variety of models for physiological processes or diseases [22], and variational problems in control theory and depolarization in Stein's model [23].

We modify system (1) to become a singular perturbed DDE of the form

$$\begin{array}{rcl} \varepsilon y'(t,\varepsilon) & = & f(t, y(t,\varepsilon), y(t-\tau,\varepsilon)), \quad t \geq t_0, \\ y(t,\varepsilon) & = & \psi(t), \quad t \leq t_0, \end{array} \tag{5}$$

where $\varepsilon > 0$. Of course, the solutions of singularly perturbed DDEs and their dynamics are fundamentally different from those of singularly perturbed ODEs without time-lags (time-delays). Of course, the presence of the delay term, $t-\tau$, or/and the small parameter ε may (*i*) stabilize unstable solutions, (*ii*) destabilize stable solutions, (*iii*) play a role in bifurcation phenomena, (*vi*) introduce oscillatory and chaotic behaviours, and (*v*) propagate discontinuities associated with initial conditions into non-smoothness of the solution. A common form of such equations is the delay-recruitment equation

$$\varepsilon y'(t,\varepsilon) = -y(t,\varepsilon) + f(y(t-\tau,\varepsilon)),$$

which can be derived for respiratory control models and blood cell populations [24]. The decay term $-y(t,\varepsilon)$ is the mortality rate, whereas the delay term is the generation rate (or recruitment rate), taken as a nonlinear function of the population size at an earlier time. The numerical control example [25]

$$\varepsilon y'(t,\varepsilon) = -y(t,\varepsilon) + \alpha_1[1 + \alpha_2 \cos(y(t-\tau,\varepsilon))]$$

shows instability and chaotic behaviour for small ε and certain values of α_1, α_2, τ that has been confirmed experimentally by Gibbs *et al.* [26].

Analytical and numerical treatments of classical DDEs have been studied extensively by many authors [27, 10, 28, 29], however, the theory and numerical solution of nonclassical or singularly perturbed DDEs are still in the initial stage. Consider another system, defined by the same function $f(t,y,x)$ of (5) but with another initial condition, so that

$$\begin{array}{rcl} \varepsilon z'(t,\varepsilon) & = & f(t, z(t,\varepsilon), z(t-\tau,\varepsilon)), \quad t \geq t_0, \\ z(t,\varepsilon) & = & \phi(t), \quad t \leq t_0. \end{array} \tag{6}$$

Different types of stability are summarized in the following definition.

Definition 1 *If there exists a norm $\|.\|$ on $\mathbb{C}^n$ for every $t \geq t_0$, the solution $y(t,\varepsilon)$ of (5) is said to be*

(i) *stable (with respect to perturbing the initial function), if for each $\epsilon^* > 0$, there exists $\delta = \delta(\epsilon^*, t_0)$ such that $\|y(t,\varepsilon) - z(t,\varepsilon)\| \leq \epsilon^*$ when $\|\psi(t) - \phi(t)\| \leq \delta$;*

(ii) *asymptotically stable, if it is stable and $\|y(t,\varepsilon) - z(t,\varepsilon)\| \to 0 \quad as \quad t \to \infty$;*

(iii) *uniformly asymptotically stable, if under condition (i) the number δ is independent of t_0;*

(iv) *ξ-exponentially stable, if it is asymptotically stable and, given t_0, there exists a finite constant K such that $\|y(t,\varepsilon) - z(t,\varepsilon)\| \leq Ke^{-\xi(t-t_0)}$.*

Here $y(t,\varepsilon)$ and $z(t,\varepsilon)$ are solutions of (5) and (6), respectively, and $\psi(t)$, $\phi(t)$ are distinct and continuous functions.

For more discussions about conditions that ensure the exponential stability and contractivity of singular perturbed DDEs (6), for sufficiently small $\varepsilon > 0$, we refer to [16].

Mono-Implicit RK Schemes for Singularly Perturbed DDEs

Although, the computational cost of explicit methods is lower than that of implicit methods, the use of implicit methods is advantageous for stiff DDEs. In this section, we provide a subclass of implicit Runge-Kutta (IRK), called Mono-Implicit Runge-Kutta (MIRK) method for system (5). The important property of MIRKs for IVPs is that they are implicit only in the single unknown y_{n+1} while for boundary value problems they are implicit only in y_n and y_{n+1}. The analysis of these schemes was simplified considerably by the work of Enright and Muir [2], who introduced MIRK formulae, for solving ODEs, as a particular class of parameterized IRKs. Given the initial value ODE

$$\begin{array}{rcll} y'(t) & = & f(t,y(t)), & 0 < t \leq T, \\ y(0) & = & y_0, & t = 0 \end{array} \tag{7}$$

with the mesh $\Delta = \{t_0 < t_1 < \ldots < t_N = T\}$ and step size $h_n = t_{n+1} - t_n$ and $y_n = y(nh_n)$. The scheme of RK is then presented by

$$y_{n+1} = y_n + h_n \sum_{r=1}^{s} b_r K_{n+1}^r, \tag{8}$$

where stages are defined by

$$K_{n+1}^r = f(t_n + c_r h_n, y_n + h_n \sum_{j=1}^{s} a_{rj} K_{n+1}^j). \tag{9}$$

Here K_n^r is the r^{th} stage, b_r, c_r and a_{rj} are defined by the following tableau (with $1 \leq r, j \leq s$),

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1s} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\ \hline & b_1 & b_2 & \ldots & b_s \end{array} \equiv \frac{\mathbf{C} \mid \mathbf{A}}{\mid \mathbf{b}^T}.$$

The values c_i (usually $\in [0,1]$) are called the *abscissae* for $i = 1, 2, ..., s$. $\mathbf{b}^T = [b_1, b_2, \ldots, b_s]$, $\sum_{i=1}^{s} b_i = 1$, $\mathbf{C} = [c_1, c_2, \ldots, c_s]^T$ and $\mathbf{A} = [a_{ij}]_{i,j=1}^{s}$ is an $s \times s$ matrix. In order to have a consistent RK method, $\mathbf{C} = \mathbf{Ae}$ where $\mathbf{e} = [1, 1, \ldots, 1]^T$ is a unit vector of length s. For explicit methods (see e.g. [30, 31]), the matrix $\mathbf{A}$ is strictly lower triangular matrix. In implicit methods, $\mathbf{A}$ is a full matrix. In other words, in case of explicit methods, the solution of (9) is straightforward and the computational cost is low. However, in case of implicit methods, we need to solve a system of non-linear equations in all stages simultaneously which is usually done by a suitable iteration method such as that of Newton, and this is computationally expensive. To lower the computational cost, many special subclasses of IRK methods such as singly-implicit RK scheme [32, 33], diagonally implicit RK scheme [34], and MIRK scheme [35] have been proposed for ODEs. These schemes differ in the choice of the matrix $\mathbf{A}$.

When the MIRK schemes applied to (7), the stages (for the n^{th}-step) take the form

$$K_{n+1}^r = f(t_n + c_r h_n, (1 - v_r) y_n + v_r y_{n+1} + h_n \sum_{j=1}^{r-1} x_{rj} K_{n+1}^j). \tag{10}$$

Here the $s \times s$ matrix $\mathbf{A} = [a_{ij}]_{i,j=1}^{s}$ defined in IRK tableau is replaced by the coefficients $\{v_r\}_{r=1}^{s}$, $v_r \in [0,1]$ and $\{x_{ij}\}_{i,j=1}^{s,i-1}$. The abscissae $\{c_r\}_{r=1}^{s}$ are defined by $c_r = v_r + \sum_{i=1}^{s} x_{ri}$. The complexity of this scheme depends on the number of stages s, the coefficients $\{v_r\}_{r=1}^{s}$, $\{x_{ij}\}_{i,j=1}^{s,i-1}$ and the weights $\{b_r\}_{r=1}^{s}$. The tableau takes the form

$$\begin{array}{c|c|ccccc} c_1 & v_1 & 0 & 0 & \ldots & 0 & 0 \\ c_2 & v_2 & x_{2,1} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & v_s & x_{s,1} & x_{s,2} & \ldots & x_{s,s-1} & 0 \\ \hline & & b_1 & b_2 & b_3 & \ldots & b_s \end{array} \equiv \frac{\mathbf{C} \mid \mathbf{V} \mid \mathbf{X}}{\mid \mid \mathbf{b}^T}.$$

Here, we suppose that $\mathbf{C} = \mathbf{V} + \mathbf{Xe}$ with $\mathbf{C}$ & $\mathbf{e}$ are defined as before. $\mathbf{V}= [v_1, v_2, ..., v_s]^T$. Substituting y_{n+1} from equation (8) into (10) one gets the Butcher's tableau

$$\begin{array}{c|c} \mathbf{C} & \mathbf{A}^* \\ \hline & \mathbf{b}^T \end{array},$$

with $a^*_{rj} = v_r b_j + x_{rj}$. In IRK methods, the stages are implicitly related to the full matrix $\mathbf{A}$ which increases the computational cost. The aim of MIRK method is to achieve two goals: (i) to lower the computational cost, (ii) to keep the high accuracy of implicit methods. To achieve the first goal, the stages are related explicitly through the coefficients of the matrix $\mathbf{X} = [x_{rj}]^{s,i-1}_{r,j=1}$ which is strictly lower triangular matrix. To achieve the second goal, a coefficient v_r is used to add implicit term in y_{n+1} only.

Applying the above scheme to the ODE (7) gives

$$y_{n+1} = R(z)y_n. \tag{11}$$

The stability function $R(z)$ can then be expressed in the form

$$R(z) = \frac{P(z, \mathbf{e\text{-}V})}{P(z, -\mathbf{V})}, \text{ where } P(z, \mathbf{w}) = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{X})^{-1}\mathbf{w},$$

where $\mathbf{w} \in \mathbb{R}^s$. The stability properties for a MIRK scheme are given in terms of this stability function $R(z)$. A-stability requires $|R(z)| \leq 1$ for all $z \in \mathbb{C}^{-1}$; While L-stability requires A-stability and the requirement that $|R(z)| \to 0$ as $\Re(z) \to -\infty$; see e.g. [36].

The order conditions for MIRK is similar to the order condition of IRK schemes. A p^{th}-order MIRK scheme has a stage-order ρ ($\rho \leq p$) if its coefficients satisfy the stage-order conditions

$$\mathbf{X}\mathbf{c}^{j-1} + \frac{v}{j} = \frac{\mathbf{c}^j}{j}, \quad j = 1, 2, \ldots, \rho, \tag{12}$$

where $\mathbf{c}^0 = \mathbf{e}$ and $\mathbf{c}^j = [c_1^j, c_2^j, ..., c_s^j]^T$. When a MIRK scheme is applied to a system of *stiff* differential equations, an order reduction phenomenon can cause a scheme having stage-order ρ (with $\rho < p$) to behave as if its order were only ρ or $\rho + 1$. Thus it required to minimization of the number of stages and maximize stage-order as possible. However, it has been proved in [37] that the maximum stage-order of a p^{th}-order MIRK scheme is $\min(p, 3)$. The scheme is of order p (*i.e.* has local[4] error of order $p + 1$) if for the local problem $y'(t_i) = f(t_i, y(t_i))$, $y(t_{i-1}) = y_{i-1}$, the numerical solution given by (11) satisfies $|y(t_i) - y_i| = O(h_i^{p+1})$. A family of schemes can be derived by requiring its coefficients to satisfy a set of equations of order conditions; see [33].

We extend the MIRK scheme with s-stages to solve the singularly perturbed IVP of DDEs (5). Assume that $\varepsilon \neq 0$, an s-stages MIRK scheme for the solution of DDE (5) for any mesh $\Delta = \{t_o < t_1 < \ldots < t_N\}$ is defined by

$$y_{n+1} = y_n + h_n \sum_{r=1}^{s} b_r K^r_{n+1}, \tag{13}$$

with stages

$$K^r_{n+1} = \frac{1}{\varepsilon} f(t_n + c_r h_n, (1 - v_r)y_n + v_r y_{n+1} + h_n \sum_{j=1}^{r-1} x_{rj} K^j_{n+1}, y(t_n + c_r h_n - \tau)). \tag{14}$$

Here, in order to find the numerical solution at each mesh point, s-stages must be evaluated each time when the solution is advanced one step. An approximation to the delay term $y(t_n + c_r h_n - \tau)$ in equation (14) is needed at any point $t_n + c_r h_n - \tau$ which is not always a mesh point. That is why a continuous approximate solution for previous meshes must be available. Continuous approximate solution is obtained by the so called *continuous mono-implicit Runge-Kutta* (CMIRK) scheme.

[4]The local error in ODEs is the error that would be made in one step if the previous values were exact and if there were no roundoff errors.

Continuous mono-implicit RK schemes for DDEs

The discrete solution at the mesh points obtained by using discrete MIRK scheme is not enough to solve the DDEs. To get an approximation to the delay term $y(t-\tau)$ in (5), a continuous approximate solution between the mesh points must be available. This can be obtained using the CMIRK scheme defined by a similar tableau [38]

$$\begin{array}{c|c|cccccc} c_1 & v_1 & 0 & 0 & \dots & 0 & 0 \\ c_2 & v_2 & x_{2,1} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{s^*} & v_{s^*} & x_{s*,1} & x_{s*,2} & \dots & x_{s^*,s^*-1} & 0 \\ \hline & & b_1(\vartheta) & b_2(\vartheta) & b_3(\vartheta) & \dots & b_{s^*}(\vartheta) \end{array} \equiv \frac{\mathbf{C} \mid \mathbf{V} \mid \mathbf{X}}{\mid \mid \mathbf{b}^T(\vartheta)}.$$

The continuous approximate solution $\hat{y}(t)$ at any point between the two mesh points t_n and t_{n+1} is then obtained by

$$\hat{y}(t_n + \vartheta h_n) = y_n + h_n \sum_{r=1}^{s^*} b_r(\vartheta) K_{n+1}^r, \quad 0 \le \vartheta \le 1, \tag{15}$$

where $s^* \ge s$ is the number of stages for CMIRK scheme, $K_{n+1}^r, r = 1, ..., s^*$ are defined by the same equation (14) as in the discrete scheme. $b_r(\vartheta), r = 1, \dots, s^*$, are polynomials in ϑ, $0 \le \vartheta \le 1$, of degree $d \le p$ where p is the order of accuracy of the RK method [39]. The additional $s^* - s$ stages in the continuous extension formula (15) are assumed in order to raise its uniform order to the order of the discrete RK methods (13).

Definition 2 *(Continuous local order) The one-step interpolant (15) has uniform order q if q is the largest integer such that*

$$\max_{t_n \le t \le t_{n+1}} \|z_n(t) - \hat{y}(t)\| = O(h_n^{q+1}). \tag{16}$$

The uniform order q of the interpolant, which clearly does not exceed p, is also the uniform order of the CMIRK *method, such that*

$$\sum_i b_i(\vartheta) c_i^s = \frac{\vartheta^{s+1}}{s+1} \quad for \quad s = 0, \dots, q-1,$$

where $\hat{y}(t)$ is the continuous extension and z_n is the local solution in $[t_n, t_{n+1}]$; See [29].

Theorem 1 *(Order of convergence of DDEs methods) If the RK method (14) and (15) is applied to (5) with a mesh $\triangle$ which includes the discontinuity points ξ_k, the underlying discrete RK method has nodal order p and if the underlying interpolant has order $q \ge p-1$, then the continuous numerical solution $\hat{y}(t)$ is such that*

$$\max_{t_0 \le t \le T} \|y(t) - \hat{y}(t)\| = O(h^p), \qquad (h = \max_{1 \le n \le N} h_n).$$

The error estimate [29] of the global error $e(t) = y(t) - y_n(t)$ where $y(t)$ is the exact solution and $y_n(t)$ is the numerical approximation of $y(t)$ is related to the orders p and q as follows:

$$\|y(t) - y_n(t)\| = O(h^{\min(p,q)}). \tag{17}$$

The polynomials $b_r(\vartheta)$ and s^* are chosen to satisfy some continuity and order conditions (see [33, 39] for more details). In the following, we present the second, third and fourth order one-sided L-stable schemes. The one-sided schemes have the property that the degree of the numerator and denominator of the stability function $R(z)$ of the scheme are different (usually by one). Such a scheme can be used to introduce growth or damping in the numerical solution. To have a family of one-sided (damped) schemes, impose the one-sided condition by requiring the stability function of the family we derive to have a denominator whose degree is one greater than that of the numerator. This gives a family of one-sided (damped) scheme We then we deduce the weight polynomials $b_r(\vartheta)$, using the continuity conditions, from the corresponding $\mathbf{b}^T$ of the discrete scheme.

Second order one-sided schemes

Given the *second order* ($p = 2$) discrete MIRK scheme with $s = 2$ such that

$$\begin{array}{c|c|cc} 1 & 1 & 0 & 0 \\ \frac{2}{7} & \frac{24}{49} & -\frac{10}{49} & 0 \\ \hline & & \frac{3}{10} & \frac{7}{10} \end{array}, \text{ and } R(z) = \frac{\frac{5}{14}z + 1}{\frac{1}{7}z^2 - \frac{9}{14}z + 1}.$$

This scheme is L-stable [39]. The continuous scheme for the *second order*, with $s^* = 3$, has polynomials coefficients $b_r(\vartheta)$ of third degree satisfying the *continuity conditions*:

$$b_r(0) = 0; \quad b_r'(0) = \delta_{r1}; \quad b_r(1) = b_r; \quad b_r'(1) = \delta_{r2} \tag{18}$$

where δ_{ij} is the Kronecker delta. The general form of a third degree polynomial is then of the form

$$b_r(\vartheta) = c_1\vartheta^3 + c_2\vartheta^2 + c_3\vartheta + c_4.$$

Applying the third condition of (18) yields $c_4 = 0$ for all r. Put $r = 1$ (to find $b_1(\vartheta)$) and apply the second condition, we have $c_3 = 1$. The third and fourth conditions give $c_1 = 1, c_2 = -2$. Hence $b_1(\vartheta) = \vartheta(\vartheta - 1)^2$. When $r = 2$, using the second condition gives $c_3 = 0$. The third and fourth conditions give $c_1 = \frac{4}{10}, c_2 = -\frac{1}{10}$. Hence $b_2(\vartheta) = \frac{1}{10}\vartheta^2(4\vartheta - 1)$. To find $b_3(\vartheta)$, putting $r = 3$ and using the second condition of (18) yields $c_3 = 0$. The third and fourth conditions give $c_1 = -\frac{14}{10}, c_2 = \frac{21}{10}$. Hence $b_3(\vartheta) = -\frac{7}{10}\vartheta^2(2\vartheta - 3)$. We arrive at the following tableau

$$\begin{array}{c|c|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \frac{2}{7} & \frac{24}{49} & 0 & -\frac{10}{49} & 0 \\ \hline & & \vartheta(\vartheta-1)^2 & \frac{1}{10}\vartheta^2(4\vartheta-1) & -\frac{7}{10}\vartheta^2(2\vartheta-3) \end{array}$$

Third one-sided order schemes

The third order discrete MIRK scheme with $s = 4$ is defined by the tableau

$$\begin{array}{c|c|cccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \frac{3}{4} & \frac{27}{32} & \frac{3}{64} & -\frac{9}{64} & 0 & 0 \\ \frac{1}{5} & \frac{13}{125} & \frac{16}{125} & -\frac{4}{125} & 0 & 0 \\ \hline & & \frac{5}{9} & -\frac{3}{8} & \frac{128}{99} & -\frac{125}{264} \end{array}$$

In the same manner, we can deduce the CMIRK scheme from the discrete scheme (when $s^* = 4$) to have, after some manipulations, the following tableau

$$\begin{array}{c|c|cccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \frac{3}{4} & \frac{27}{32} & \frac{3}{64} & -\frac{9}{64} & 0 & 0 \\ \frac{1}{5} & \frac{13}{125} & \frac{16}{125} & -\frac{4}{125} & 0 & 0 \\ \hline & & -\frac{1}{9}\vartheta(\vartheta^2+3\vartheta-9) & \frac{1}{8}\vartheta^2(14\vartheta-17) & -\frac{128}{99}\vartheta^2(2\vartheta-3) & \frac{125}{264}\vartheta^2(2\vartheta-3) \end{array}$$

Fourth order one-sided schemes

The fourth order discrete MIRK scheme is obtained when $s = 5$ and defined by the tableau

$$\begin{array}{c|c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{20} & \frac{29}{4000} & \frac{361}{8000} & -\frac{19}{8000} & 0 & 0 & 0 \\ \frac{19}{20} & \frac{3971}{4000} & \frac{19}{8000} & -\frac{361}{8000} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{11}{16} & \frac{1}{32} & \frac{267}{608} & \frac{25}{684} & -\frac{25}{36} & 0 \\ \hline & & -\frac{43}{228} & -\frac{43}{228} & \frac{25}{57} & \frac{25}{57} & \frac{1}{2} \end{array}$$

The stability function for this scheme is $R(z) = \dfrac{1 + \frac{13}{32}z + \frac{5}{96}z^2}{1 - \frac{19}{32}z + \frac{7}{84}z^2 - \frac{1}{64}z^3}$, and the scheme is L-stable. We also employ the continuity conditions and order barriers conditions to obtain the weight polynomials of the CMIRK scheme (with $s^* = 5$) that takes the form

$$\begin{array}{c|c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{20} & \frac{29}{4000} & \frac{361}{8000} & -\frac{19}{8000} & 0 & 0 & 0 \\
\frac{19}{20} & \frac{3971}{4000} & \frac{19}{8000} & -\frac{361}{8000} & 0 & 0 & 0 \\
\frac{1}{2} & \frac{11}{16} & \frac{1}{32} & \frac{267}{608} & \frac{25}{684} & -\frac{25}{36} & 0 \\
\hline
 & & b_1(\vartheta) & b_2(\vartheta) & b_3(\vartheta) & b_4(\vartheta) & b_5(\vartheta)
\end{array}$$

where

$$\begin{aligned}
b_1(\vartheta) &= -\tfrac{1}{228}\vartheta(1200\vartheta^3 - 2714\vartheta^2 + 1785\vartheta - 228), \\
b_2(\vartheta) &= \tfrac{1}{228}\vartheta^2(1200\vartheta^2 - 2086\vartheta + 843), \\
b_3(\vartheta) &= \tfrac{25}{171}\vartheta^2(40\vartheta^2 - 86\vartheta + 49), \\
b_4(\vartheta) &= -\tfrac{25}{171}\vartheta^2(40\vartheta^2 - 74\vartheta + 31), \\
b_5(\vartheta) &= -\tfrac{1}{2}\vartheta^2(2\vartheta - 3).
\end{aligned} \tag{19}$$

Error/defect control

The numerical algorithm implemented in MIRK methods for solving DDEs (5) can be briefly summarized as follows:

1. Initial approximations to the solution at mesh points $\mathbf{Y} = \{y_n\}_{n=0}^{m}$, $\tau = mh_n$ are used to evaluate stages $\{K_{n+1}^r\}_{r=1}^{s}$ by (14).
2. The residual function $\mathbf{\Phi(Y)}$ whose n^{th} component is $\Phi_n = y_{n+1} - y_n - h_n \sum_{i=1}^{s} b_r K_{n+1}^r$ is evaluated at each mesh point. Then the system of equations $\mathbf{\Phi(Y)=0}$ is solved using Newton's iteration.
3. The Newton's iteration is terminated when the numerical solution $\mathbf{Y}$ causes $\mathbf{\Phi(Y)}$ to be within defined tolerance, otherwise steps 1 and 2 are repeated.
4. The CMIRK scheme is then used to obtain the dense output $\hat{y}(t)$, (15), for each step to have the solution at nonmesh points. Then an estimate to the *relative defect* function [2] at continuous approximation is defined by

$$\text{Defect}(t_i) = \frac{|\hat{y}'(t_i) - f(t_i, \hat{y}(t_i), \hat{y}(t_i - \tau))|}{1 + |f(t_i, \hat{y}(t_i), \hat{y}(t_i - \tau))|}. \tag{20}$$

5. An adaptive algorithm is used to control the step size in order to keep the defect (20) within a specified tolerance. If the defect is not within tolerance then the step size is halved and the iteration is repeated.

Stability of MIRK Schemes for DDEs

The stability of MIRK method is studied when applying the scheme defined by (13), (14) to the test problem

$$\varepsilon y'(t, \varepsilon) = \lambda y(t, \varepsilon) + \mu y(t - \tau, \varepsilon), \quad t \geq 0 \tag{21}$$

$$y(t, \varepsilon) = \psi(t), \quad -\tau \leq t \leq 0. \tag{22}$$

which results in

$$y_{n+1} = y_n + h_n \sum_{r=1}^{s} b_r K_{n+1}^r. \tag{23}$$

The delayed term is obtained using CMIRK defined by

$$u(t_n + \theta h_n) = y_n + h_n \sum_{r=1}^{s} b_r(\theta) K_{n+1}^r, \tag{24}$$

where

$$K_{n+1}^{r} = \lambda^{*}\{(1-v_r)y_n + v_r y_{n+1} + h_n \sum_{j=1}^{r-1} x_{rj} K_{n+1}^{j}\} + \mu^{*}\{y_{n-m} + h_{n-m} \sum_{i=1}^{s} b_i(c_r) K_{n-m+1}^{i}\}. \tag{25}$$

Here $\lambda^{*} = \lambda/\varepsilon$ and $\mu^{*} = \mu/\varepsilon$ ($\varepsilon \neq 0$).

Define the vectors $\mathbf{K}_{n+1} = [K_{n+1}^{1}, K_{n+1}^{2}, \ldots, K_{n+1}^{s}]^{T}$ and $\Psi_n = [y_n, h_n \mathbf{K}_n^{T}]^{T}$, then the previous equations can be arranged as

$$\Psi_{n+1} = \mathbf{P}(\alpha)\Psi_n + \beta \mathbf{Q}(\alpha)\Psi_{n-m+1} + \beta \mathbf{R}(\alpha)\Psi_{n-m} \tag{26}$$

where $\mathbf{P}(\alpha)$, $\mathbf{Q}(\alpha)$ and $\mathbf{R}(\alpha)$ are defined as

$$\mathbf{P}(\alpha) = \begin{bmatrix} 1+\alpha \mathbf{b^{T}Se} & 0 \\ \alpha \mathbf{Se} & O \end{bmatrix},\ \mathbf{Q}(\alpha) = \begin{bmatrix} 0 & \mathbf{b^{T}SB} \\ O & \mathbf{SB} \end{bmatrix} \text{and } \mathbf{R}(\alpha) = \begin{bmatrix} \mathbf{b^{T}Se} & 0 \\ \mathbf{Se} & O \end{bmatrix}. \tag{27}$$

For simplicity we assume h_n is fixed and $\tau = 1$. $\alpha = \lambda^{*}h_n$, $\beta = \mu^{*}h_n$, $\mathbf{O}$ is $s \times s$ zero matrix, $\mathbf{b}^{T} = [b_1, b_2, \ldots, b_s]$, $\mathbf{S} = (\mathbf{I} - \alpha(\mathbf{V}\mathbf{b}^{T} + \mathbf{X}))^{-1}$, $\mathbf{e} = [1, 1, \ldots, 1]^{T}$ is a unit vector of length s, $\mathbf{X} = [x_{ij}]_{i,j=1}^{s,i-1}$ and $\mathbf{B} = [b_j(c_i)]_{i,j=1}^{s}$ are $s \times s$ matrices. From equation (26) the characteristic equation takes the form

$$det[\xi^{m+1} I - \xi^{m}\mathbf{P}(\alpha) - \xi\beta\mathbf{Q}(\alpha) - \beta\mathbf{R}(\alpha)] = 0. \tag{28}$$

The characteristic equations (as examples) for second, third and fourth order MIRK method are then deduced as follows:

- The characteristic equation for second order MIRK is

$$\begin{aligned} \xi^{2m+1} &- \frac{1+\frac{5}{14}\alpha}{1-\frac{9}{14}\alpha+\frac{\alpha^2}{7}}\xi^{2m} - \frac{\frac{43}{98}-\frac{9}{49}\alpha}{1-\frac{9}{14}\alpha+\frac{\alpha^2}{7}}\beta\xi^{m+1} \\ &+ \left(\frac{\frac{2}{49}\alpha-1}{1-\frac{9}{14}\alpha+\frac{\alpha^2}{7}} + \frac{\frac{34}{98}-\frac{583}{1372}\alpha+\frac{53}{343}\alpha^2-\frac{1}{49}\alpha^3}{(1-\frac{9}{14}\alpha+\frac{\alpha^2}{7})^2}\right)\beta\xi^{m} \\ &+ \frac{\frac{2}{49}}{1-\frac{9}{14}\alpha+\frac{\alpha^2}{7}}\beta^2\xi - \frac{\frac{5}{49}}{1-\frac{9}{14}\alpha+\frac{\alpha^2}{7}}\beta^2 = 0. \end{aligned}$$

- The characteristic equation for third order MIRK is

$$\xi^{2m} - \frac{1+\frac{1}{3}\alpha}{1-\frac{2}{3}\alpha+\frac{\alpha^2}{6}}\xi^{2m-1} + \frac{-\frac{2}{3}+\frac{1}{3}\alpha}{1-\frac{2}{3}\alpha+\frac{\alpha^2}{6}}\beta\xi^{m} - \frac{\frac{1}{3}}{1-\frac{2}{3}\alpha+\frac{\alpha^2}{6}}\beta\xi^{m-1} + \frac{\frac{1}{6}}{1-\frac{2}{3}\alpha+\frac{\alpha^2}{6}}\beta^2 = 0.$$

- The characteristic equation for fourth order MIRK is

$$\begin{aligned} \xi^{3m+1} &- \frac{192+78\alpha+10\alpha^2}{192-114\alpha+28\alpha^2-3\alpha^3}\xi^{3m} - \frac{96-\frac{4719}{100}\alpha+\frac{6133}{800}\alpha^2-\frac{57}{1600}\alpha^3}{192-114\alpha+28\alpha^2-3\alpha^3}\beta\xi^{2m+1} \\ &- \frac{96+\frac{2919}{100}\alpha+\frac{1219}{800}\alpha^2-\frac{57}{1600}\alpha^3}{192-114\alpha+28\alpha^2-3\alpha^3}\beta\xi^{2m} + \frac{1919-\frac{5009}{8}\alpha+\frac{57}{8}\alpha^2}{100(192-114\alpha+28\alpha^2-3\alpha^3)}\beta^2\xi^{m+1} \\ &- \frac{1919+\frac{499}{16}\alpha-\frac{57}{8}\alpha^2}{100(192-114\alpha+28\alpha^2-3\alpha^3)}\beta^2\xi^{m} - \frac{\frac{319}{2}-\frac{57}{16}\alpha}{100(192-114\alpha+28\alpha^2-3\alpha^3)}\beta^3\xi \\ &- \frac{\frac{319}{2}-\frac{57}{16}\alpha}{100(192-114\alpha+28\alpha^2-3\alpha^3)}\beta^3 = 0. \end{aligned}$$

Stability regions

Regions of stability can be computed using the boundary locus (or grid-search) technique, in which one seeks the loci on which the characteristic polynomial has a zero of modulus unity ($|\xi| < 1$). Such a region can be obtained in the (λ,μ)-plane (see [40]). Fig.1 shows unbounded numerical stability region of fourth order MIRK method when $m = 25$ ($\tau = mh_n$), compared with the analytical asymptotically stability region. However, we get bounded stability regions for explicit schemes; See Figure 2. The bounded stability regions, to the explicit schemes, decrease as ε getting smaller.

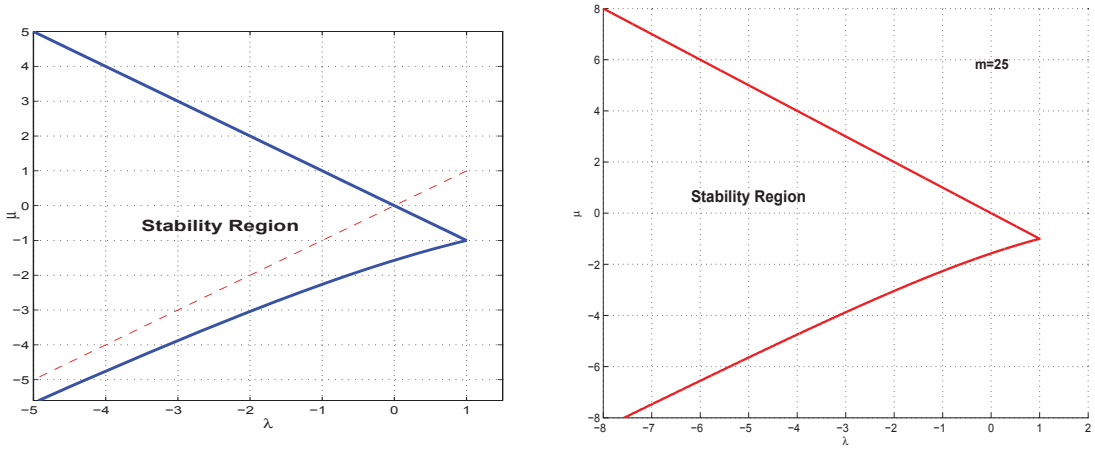

FIGURE 1. Left banner shows the analytical stability region for the test problem $\varepsilon y'(t) = \lambda y(t) + \mu y(t-1)$, while the right banner shows numerical stability regions for the 4^{th} order MIRK schemes, with m=25 ($h = \tau/m$), $\varepsilon = 1$.

Numerical Simulations

In this section we want to test our method on some simple stiff problems.

Example 1 *Consider the scalar case of singularly perturbed problem (21). For convenient suppose that the initial function $\psi(t)$ is smooth and $\lambda < 0$. Employing method of steps, for (21), when $\psi(t) = c$ yields*

$$y(t,\varepsilon) = \begin{cases} R_2 e^{\frac{\lambda}{\varepsilon}t} - cR_1, & 0 < t \le \tau, \\ R_2 e^{\frac{\lambda}{\varepsilon}t} + R_2\Big[\frac{\mu}{\varepsilon}(t-\tau) - R_1\Big]e^{\frac{\lambda}{\varepsilon}(t-\tau)} + cR_1^2, & \tau < t \le 2\tau, \\ R_2 e^{\frac{\lambda}{\varepsilon}t} + R_2\Big[\frac{\mu}{\varepsilon}(t-\tau) - R_1\Big]e^{\frac{\lambda}{\varepsilon}(t-\tau)} + \\ R_2\Big[\frac{\mu}{\varepsilon}\Big(\frac{\mu}{2\varepsilon}(t-2\tau)^2 - R_1(t-2\tau)\Big) + R_1^2\Big]e^{\frac{\lambda}{\varepsilon}(t-2\tau)} - cR_1^3, & 2\tau < t \le 3\tau, \\ R_2 e^{\frac{\lambda}{\varepsilon}t} + R_2\Big[\frac{\mu}{\varepsilon}(t-\tau) - R_1\Big]e^{\frac{\lambda}{\varepsilon}(t-\tau)} + \\ R_2\Big[\frac{\mu}{\varepsilon}\Big(\frac{\mu}{2\varepsilon}(t-2\tau)^2 - R_1(t-2\tau)\Big) + R_1^2\Big]e^{\frac{\lambda}{\varepsilon}(t-2\tau)} + \\ R_2\Big\{\frac{\mu}{\varepsilon}\Big[\frac{\mu}{2\varepsilon}\Big(\frac{\mu}{3\varepsilon}(t-3\tau)^3 - R_1(t-3\tau)^2\Big) + R_1^2(t-3\tau)\Big] - R_1^3\Big\}e^{\frac{\lambda}{\varepsilon}(t-3\tau)} + \\ cR_1^4, & 3\tau < t \le 4\tau, \\ \vdots & \vdots \\ R_2 e^{\frac{\lambda}{\varepsilon}t} + R_2 R_1^\natural e^{\frac{\lambda}{\varepsilon}(t-\tau)} + R_2 R_2^\natural e^{\frac{\lambda}{\varepsilon}(t-2\tau)} + R_2 R_3^\natural e^{\frac{\lambda}{\varepsilon}(t-3\tau)} + \\ R_2 R_4^\natural e^{\frac{\lambda}{\varepsilon}(t-4\tau)} + \ldots + R_2 R_{(n-1)}^\natural e^{\frac{\lambda}{\varepsilon}(t-(n-1)\tau)} + (-1)^n cR_1^n, & (n-1)\tau < t \le n\tau \end{cases} \tag{29}$$

where

$$R_n^\natural = \Big(\frac{\mu}{\varepsilon}\Big)^n \frac{1}{n!}(t-n\tau)^n - \Big(\frac{\mu}{\varepsilon}\Big)^{(n-1)} \frac{R_1}{(n-1)!}(t-n\tau)^{(n-1)} + \Big(\frac{\mu}{\varepsilon}\Big)^{(n-2)} \frac{R_1^2}{(n-2)!}(t-n\tau)^{(n-2)} + \ldots + (-1)^n R_1^n$$

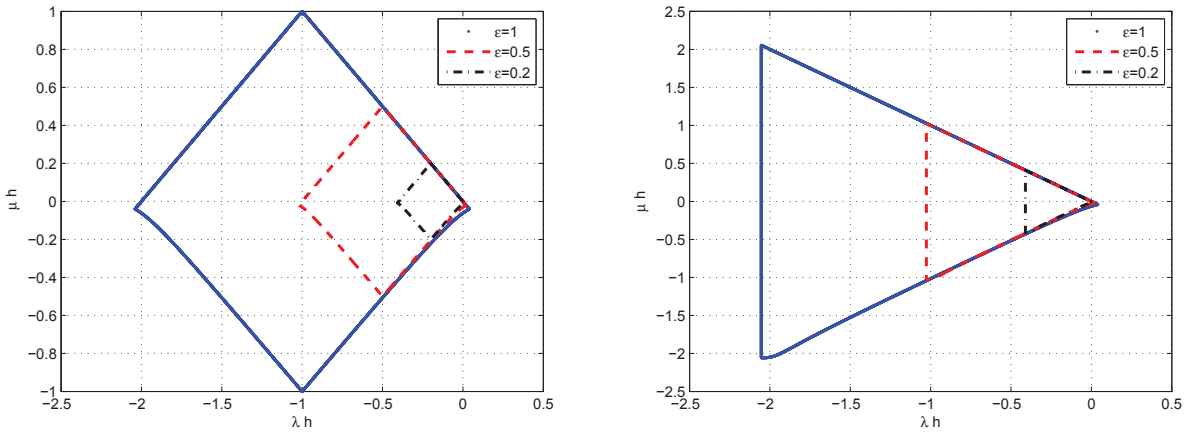

FIGURE 2. Numerical stability regions in $(\lambda h, \mu h)$-plane for explicit schemes of order one (left) and order two (right) of singular perturbation DDE $\varepsilon y'(t) = \lambda y(t) + \mu y(t-\tau)$, with different values of ε. The bounded stability region decreases as ε getting smaller.

and

$$R_1 = \frac{\mu}{\lambda}, \quad R_2 = c(1+R_1) \quad \textit{and} \quad \lambda < 0.$$

The degenerate equation (when $\varepsilon > 0$ is very small) is then the difference equation

$$\begin{aligned} y(t,0) &= -\frac{\mu}{\lambda} y(t-\tau,0), \quad t > 0, \\ y(t,0) &= c, \quad t \in [-\tau, 0], \end{aligned} \tag{30}$$

that has solution $y(t,0) = \left(-\frac{\mu}{\lambda}\right)^n c$ when $t \in [(n-1)\tau, n\tau]$, $n = 1, 2, 3, \ldots$ (rising if $|\mu| > |\lambda|$, descending if $|\mu| < |\lambda|$).

We see from (29) at each point $t = k\tau$, $k = 0, 1, 2, \ldots$ that

$$\lim_{t \to (k\tau)_-} \left(\lim_{\varepsilon \to 0} y(t,\varepsilon) \right) \neq \lim_{\varepsilon \to 0} \left(\lim_{t \to (k\tau)_+} y(t,\varepsilon) \right). \tag{31}$$

Such non-uniform behaviour occurs and propagates at $t = n\tau$ ($n = 0, 1, 2, \ldots$) whenever the initial function $\psi(t,\varepsilon)$ is not smooth in the initial points for the degenerate equation (30).

The degenerate solution $y(t,0)$ of (30) therefore provides a very close approximation to $y(t,\varepsilon)$ for sufficient small $\varepsilon > 0$. We also note from (29) and (30) that $\lim_{\varepsilon \to 0} y(t,\varepsilon) = y(t,0)$ only when $\lambda < 0$ and $\psi(t) = c$ (any constant). However, if $\lambda > 0$ $\lim_{\varepsilon \to 0} y(t,\varepsilon) = y(t,0)$ if and only if $\lambda = -\mu$. The true solution $y(t,\varepsilon)$ does not possess discontinuities for $t > 0$ and may smooth out, while the degenerate equation (30) only possesses a piecewise continuous solution. However, the true solution $y(t,\varepsilon)$ displays a boundary layer at the right side of each interval that gives rise large derivatives (in the time) of the solution for small ε.

The stability regions for explicit and trapezium rule method for the test problem (21) are displayed in Figure 2, and the exact solution of the problem compared with the solution of the degenerated equation is given in Figure 3, while the numerical solutions are displayed in Figure 4. The plots have a kink at $t = n\tau$ ($n = 1, 2, \ldots$) as a result of the existence of the delay term. The propagated 'boundary' layer is clearly visible and the large derivatives for small ε can introduce large inaccuracies in the numerical solutions.

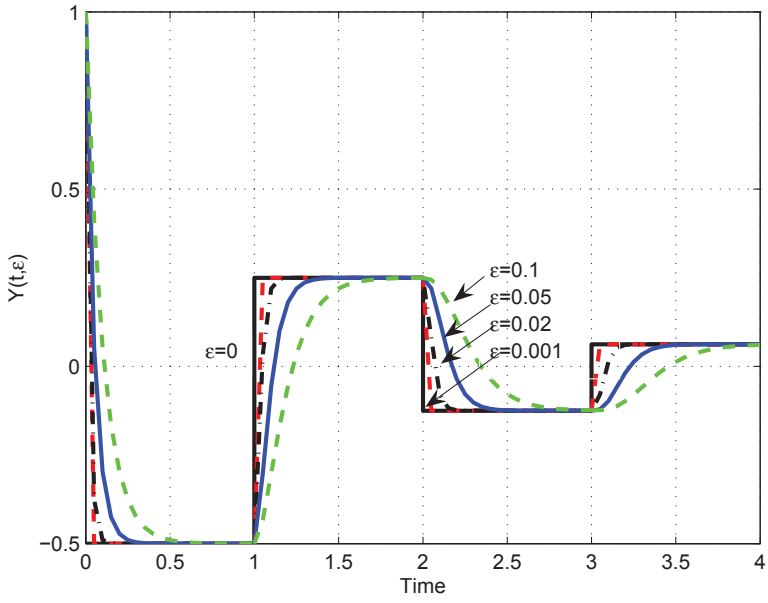

FIGURE 3. Exact solutions, $y(t,\varepsilon)$, of the singular perturbation DDE $\varepsilon y'(t,\varepsilon) = \lambda y(t,\varepsilon) + \mu y(t-\tau,\varepsilon),\ t \geq 0;\ y(t,\varepsilon) = 1$ when $t \leq 0$ for different values of ε compared with the degenerate solution $y(t,0)$.

Example 2 *Consider another stiff problem represented by DDE of the form*

$$\begin{aligned} \varepsilon y'(t) &= -\beta y(t) + \gamma y(t - \tfrac{3\pi}{2}) + \sin t, \qquad t \geq 0, \\ \phi(t) &= \exp(pt) + \sin t, \qquad t \leq 0, \end{aligned} \tag{32}$$

where $\varepsilon = (\exp(-3p\pi/2) - p)^{-1} \ll 1$, *for* $p \leq -1$, $\beta = 1$, $\gamma = \varepsilon$. *The parameter p is the stiffness parameter. The stiffness of the problem increases as p gets more negative. The analytical solution of this problem is* $y(t) = \exp(pt) + \sin t$. *The running time when solving (32) when* $p = -1$ *and* $p = -2$ *is* 43 *seconds when we use the explicit scheme of* DDE23 *package and* 7 *seconds only when one uses* CMIRK *of order 2 with the same conditions. Table 1 shows a comparison between the error in the numerical solution, at* $t_f = \frac{3\pi}{2}$ *and* $p = -2$, *when using* DDE23 *and* CMIRK *scheme of order 2. there is a big difference in the accuracy of the solution obtained by* DDE23 *and that obtained by* CMIRK *scheme. This is due to the fact that even if the explicit method gives a solution for the stiff problem, the computational cost of explicit methods is higher than implicit methods in case of a stiff problem as well as the accuracy is very low compared to those of implicit method. When the stiffness parameter p increases to* $p = -3$, DDE23 *code fails to get the solution. However,* CMIRK *scheme gives a good approximation to the exact solution even with higher values of the stiff parameter p. The graphs of Figure 5 show the numerical solutions obtained by* CMIRK *scheme and the corresponding error* $(y(t) - y_n)$ *when* $p = -10$. *However,* DDE23 *does not provide numerical solutions when* $p = -10$.

TABLE 1. Shows the error in the numerical solution of singularly perturbed DDE (32), using DDE23 and CMIRK scheme of order 2 with $p = -2$ at t_f.

TOL	Error (DDE23)	Error (CMIRK)
10^{-3}	$4.905266E-4$	$-2.382428E-12$
10^{-4}	$4.777792E-5$	$-7.327472E-15$
10^{-5}	$4.767974E-6$	$2.553513E-15$

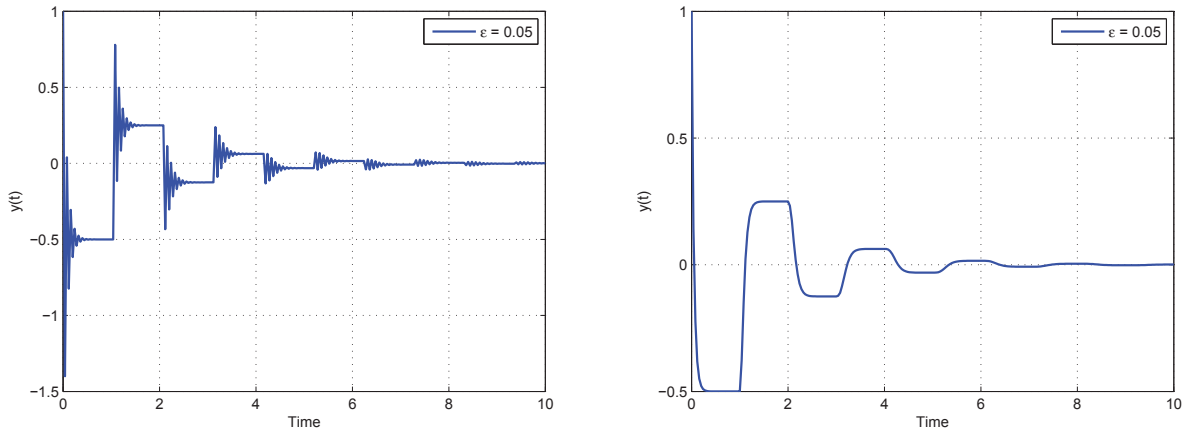

FIGURE 4. Numerical simulations, using explicit formula (left) of order 2 and CMIRK formula (right) for the same order, for $\varepsilon y'(t) = \lambda y(t) + \mu y(t-\tau),\ t \geq 0; y(t) = 1\ for\ t \leq 0$ and with $\tau = 1 = mh;\quad \lambda = -1,\ \mu = -0.5,\ m = 25.$ when $\varepsilon = 0.05$ & 0.02, respectively.

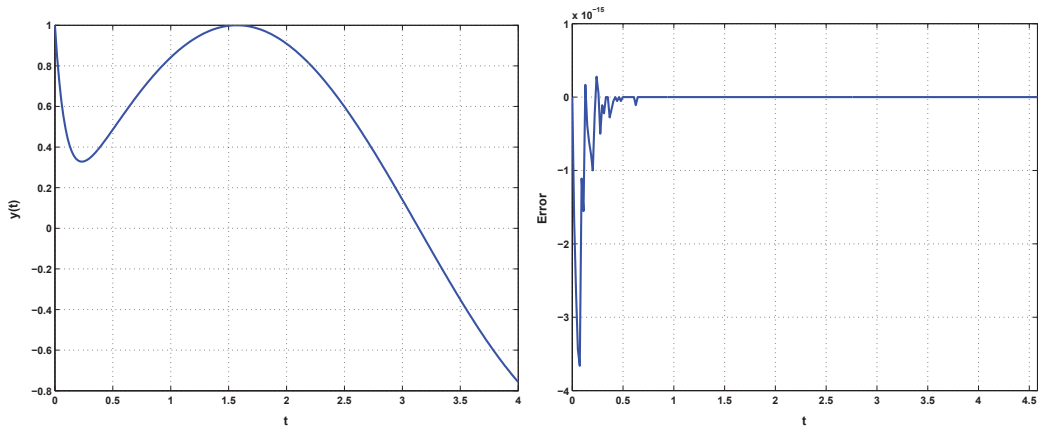

FIGURE 5. Numerical simulation of singularly perturbed problem (32) (with $p = -10$) obtained by CMIRK scheme (left), and the error (right).

Conclusion

It is important to be able to solve a stiff problems efficiently, without solving complicated nonlinear systems at each step. In this paper, we provided an unconditionally stable method based on mono-implicit RK schems for numerical treatments of stiff singularly perturbed delay differential equations. Mono-implicit RK method combines the accuracy of implicit methods and the efficient implementation of explicit methods which is suitable for both stiff and non-stiff IVPs. The numerical results obtained by CMIRK schemas of order 2 are better, compared with those obtained by DDE23 code, which is based on explicit schemes, when solving stiff problems. Even if the explicit method gives a solution for the stiff problem, the computational cost for explicit methods is higher than implicit methods in case of stiff problem while the accuracy is very low compared to those of implicit method. The explicit formula becomes unstable as ε decreases, however, the implicit formula is stable. Clearly a stiff solver is needed for a singularly perturbed DDEs. Mono-implicit schems are suitable for stiff and non-stiff IVPs. This work is extendable to solve stiff and non-stiff integro-delay differential equations.

Acknowledgement

This work is supported by funding received by the joint research project UAEU/SQU 2016.

REFERENCES

[1] C. F. Curtiss, J. O. Hirschfelder, Integration of stiff equations, Proc Natl Acad Sci U S A. 38 (3) (1952) 235–2343.
[2] W. Enright, P. Muir, A Runge Kutta boundary value ODE solver with defect control, Technical Report 267/93, Department of Computer Science, University of Toronto, 1993.
[3] K. Dekker, J. Verwer, Stability of Runge- Kutta Methods for Stiff Nonlinear Differential Equations, CWI Monographs 2, Amsterdam, 1984.
[4] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer-Verlag, Berlin, 1996.
[5] E. Hairer, G. Wanner, Stiff differential equations solved by Radau methods, J. Comput. Appl. Math. 111 (12) (1999) 93–111.
[6] G. A. Bocharov, G. I. Marchuk, A. A. Romanyukha, Numerical solution by LMMs of a stiff delay-differential system modelling an immune response, Numer. Math. 73 (1996) 131–148.
[7] N. Guglielmi, E. Hairer, Implementing Radau II-A methods for stiff delay differential equations, Computing 67 (2001) 1–12.
[8] N. Guglielmi, E. Hairer, Computing breaking points in implicit delay differential equations, Adv. Comput. Math. 29 (3) (2008) 229–247.
[9] K. Strehmel, R. Weiner, H. Claus, Stability analysis of linearly implicit one-step interpolation methods for stiff retarded differential equations, SIAM J. Numer. Anal. 26 (5) (1989) 1158–1174.
[10] C. T. H. Baker, C. A. H. Paul, Issues in the numerical solution of evolutionary delay differential equations, J. Adv. Comput. Math. 3 (1995) 171–196.
[11] V. B. Kolmanovskii, V. R. Nosov, Stability of Functional Differential Equations, Academic Press, New York, 1986.
[12] G. G. Kirlinger, Linear multistep methods applied to stiff initial value problems, J. Math. Comput. Mod. 40 (2004) 1181–1192.
[13] H. Podhaisky, R. Weiner, B. A. Schmitt, Rosenbrock-type Peer two-step methods, Appl. Numer. Math. 53 (2005) 409–420.
[14] H. S. Strogatz, Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering, Colorado Westview Press, Denver, 2000.
[15] H. Podhaisky, W. Marszalek, Bifurcations and synchronization of singularly perturbed oscillators: an application case study, Nonlinear Dynam. 69 (2012) 949–959.
[16] F. A. Rihan, Stability conditions for singularly perturbed delay differential equations (2008).
[17] G. A. Bocharov, F. A. Rihan, Numerical modelling in biosciences using delay differential equations, J. Comput. Appl. Math. 125 (2000) 183–199.
[18] F. A. Rihan, B. F. Rihan, Numerical modelling of biological systems with memory using delay differential equations, Appl. Math. & Inf. Sci. 9 (3) (2015) 1615–1658.
[19] G. Dahlquist, Error analysis for a class of methods for stiff nonlinear initial value problems, in: G.A. Watson (Ed.), Numerical Analysis (1975), in: Lecture Notes in Mathematics, vol. 506, Springer-Verlag, Berlin, 1975.
[20] A. Halanay, Differential Equations– Stability, Oscillations, Time Lags, Academic Press, New York, 1966.
[21] A. Longtin, J. Milton, Complex oscillations in the human pupil light reflex with mixed and delayed feedback, Math. Biosci. 90 (1988) 183199.
[22] M. C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (1977) 287–289.
[23] V. Y. Glizer, Asymptotic analysis and solution of a finitehorizon H_∞ control problem for singularly-perturbed linear systems with small state delay, J. Optim. Theor. Appl. 117 (2) (2003) 295–325.
[24] A. C. Fowler, G. P. Kalamangalam, G. Kember, A mathematical analysis of the grodin model of respiratory control, IMA J. Math. Appl. Med. Biol. 10 (1993) 249–280.
[25] K. Ikeda, Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system, Opt. Commun. 30 (1979) 257.

[26] H. M. Gibbs, F. Hopf, D.L.Kaplan, R. Shoemaker, Observation of chaos in optical bistability, Phys. Rev. Lett. 46 (1981) 474–477.
[27] C. T. H. Baker, Retarded differential equations, J. Comput. Appl. Math. 125 (2000) 309–355.
[28] A. Bellen, M. Zennaro, Numerical Methods for Delay Differential Equations, Oxford University Press, New York, 2003.
[29] F. A. Rihan, Numerical Treatment of Delay Differential Equations in Bioscience, PhD. Thesis, University of Manchester, UK, 2000.
[30] C. A. H. Paul, A user-guide to Archi – an explicit Runge-Kutta code for solving delay and neutral differential equations and parameter estimation problems, MCCM report 283, 1997, ISSN 1360-1375, University of Manchester.
[31] L. F. Shampine, S. Thompson, Solving DDEs in Matlab, Appl. Numer. Math. 37 (2001) 441–458.
[32] K. Burrage, A special family of Runge-Kutta methods for solving stiff differential equations, BIT 18 (1978) 22–41.
[33] J. Butcher, P. Chartier, A generalization of singly-implicit Runge-Kutta methods, J. Appl. Num. Math. 24 (1997) 343–350.
[34] S. Norsett, Semi-explicit Runge-Kutta methods, report, Math. and Comp. No. 6/74, Dept. of Math., University of Trondheim (1974).
[35] J. Cash, A. Singhal, Mono-implicit Runge-Kutta formula for the numerical integration of stiff differential systems, IMA J. Num. Anal. 2 (1982) 211–227.
[36] J. Butcher, The Numerical Analysis of Ordinary Differential Equations, Wiley London, 1987.
[37] K. Burrage, F. Chipman, P. Muir, Order results for mono-implicit Runge-Kutta methods (1994).
[38] F. A. Rihan, E. H. Doha, M. Hassan, N. M. Kamel, Numerical treatments for volterra delay integro-differential equations, Comput. Methods Appl. Math. 9 (3) (2009) 292–308.
[39] P. H. Muir, Optimal discrete and continuous mono-implicit Runge-Kutta schemes for BVODEs, J. Adv. Comput. Math. 10 (1999) 135–167.
[40] C. A. H. Paul, C. T. H. Baker, Stability boundaries revisited Runge-Kutta methods for delay differential equations, Rep.No. 205, University of Manchester 1991.