Two–step diagonally–implicit collocation based methods for Volterra Integral Equations

D. Conte, R. D'Ambrosio, B. Paternoster

Dipartimento di Matematica e Informatica
Università di Salerno
Via Ponte Don Melillo
I-84084 Fisciano (SA)
Italy

Abstract

We introduce a family of diagonally–implicit continuous methods for the numerical integration of Volterra Integral Equations. The derived methods are characterized by a lower triangular or diagonal coefficient matrix of the nonlinear system for the computation of the stages which, as it is known, can be exploited to get an efficient implementation. The constructed methods have an high uniform order of convergence together with strong stability properties (e.g. *A*-stability).

Keywords: Numerical methods for Volterra Integral Equations, Two–step collocation methods, Two–step Runge–Kutta methods, Diagonally–implicit methods.
2000 MSC: 65R20

1. Introduction

This paper concerns the construction of both efficient and highly stable numerical methods for Volterra Integral Equations (VIEs) of the form

$$y(t) = g(t) + \int_0^t k(t, \tau, y(\tau))d\tau, \quad t \in [0, T], \tag{1.1}$$

where the *forcing function* $g : \mathbb{R} \to \mathbb{R}^d$ and the *kernel* $k : \mathbb{R}^2 \times \mathbb{R}^d \to \mathbb{R}^d$ are assumed to be sufficiently smooth. Such equations arise in many models of evolutionary phenomena with memory and their classical numerical treatment has been widely described in [3, 4] and the related bibliography. It is well known that the best stability properties are reached by implicit numerical methods, with the disadvantage that they lead to nonlinear systems of equations to be solved at each time step. To this cost we have to add the cost arising from the computation of the lag term (containing the history of the phenomenon), which can be computed by means of fast methods developed in the literature for convolution kernels [6, 7, 20]. As regards the task of reducing the computational cost associated to the solution of the above nonlinear systems, a widespread strategy in the context of Ordinary Differential Equations (ODEs), consists in making the coefficient matrix have a structured shape. This strategy, in the field of Runge–Kutta methods for ODEs, leads to the raise of the famous classes of Diagonally Implicit and Singly Diagonally Implicit Runge-Kutta methods (DIRK and SDIRK), see [5, 17] and bibliography therein contained. Moreover, in the field of collocation-based methods for ODEs, an analogous strategy has been applied, obtaining a subclass of two-step Runge–Kutta methods (see [19]) having structured coefficient matrix [16].

In this paper we will adopt this strategy in the context of the numerical solution of VIEs (1.1) and, in particular, we will derive numerical methods involving nonlinear systems with lower triangular or diagonal coefficient matrices. The derived formulae belong to the class of Two-Step Almost Collocation (TSAC) methods, which have been

Email address: dajconte@unisa.it, rdambrosio@unisa.it, beapat@unisa.it (D. Conte, R. D'Ambrosio, B. Paternoster)

introduced in [8] in order to obtain high order and highly stable continuous methods for the problem (1.1). In fact, to integrate a system of d integral equations of the form (1.1), a collocation method generally requires the solution of md simultaneous nonlinear equations at each time step. A lower triangular matrix allows to solve the equations in m successive stages, with only a d-dimensional system to be solved at each stage. Moreover, if all the elements on the diagonal are equal, in solving the nonlinear systems by means of Newton-type iterations, one may hope to use repeatedly the stored LU factorization of the Jacobian. If the structure is diagonal, the problem reduces to the solution of m independent systems of dimension d, and can therefore be solved in a parallel environment.

The paper is structured as follows. In Section 2 we recall two-step almost collocation methods, together with the main results on continuous order conditions and convergence. Section 3 is devoted to the construction of diagonally implicit methods within this class, while Section 4 contains the analysis of the linear stability properties with respect to the basic test equation usually employed in the literature. Examples of $A-$stable methods are provided in Section 5 and some numerical experiments are given in Section 6. Some conclusions and future developements are remarked in Section 7.

2. Two-step collocation and almost collocation methods

Let $I_h = \{t_n := nh, n = 0, ..., N,\ \ h \geq 0, Nh = T\}$ be a uniform mesh, which constitutes the discrete counterpart of the interval $[0, T]$, and $c_1, \ldots, c_m$ be m collocation parameters, which identify m internal points $t_{nj} = t_n + c_j h$ inside the generic interval $[t_n, t_{n+1}]$.

Equation (1.1) can then be expressed in the following way

$$y(t) = F^{[n]}(t, y(\cdot)) + \Phi^{[n+1]}(t, y(\cdot)), \quad t \in [t_n, t_{n+1}],$$

where

$$F^{[n]}(t, y(\cdot)) := g(t) + \int_0^{t_n} k(t, \tau, y(\tau))d\tau, \quad \Phi^{[n+1]}(t, y(\cdot)) := \int_{t_n}^{t} k(t, \tau, y(\tau))d\tau$$

are the *lag term* and the *increment term* respectively.

TSAC methods for VIEs [8] provide a continuous approximation $P_n(t_n + sh)$, $s \in [0, 1]$, to the solution $y(t_n + sh)$ in the interval $[t_n, t_{n+1}]$, which employs the information about the equation on two consecutive steps and suitable sufficiently high order quadrature formulae $F_j^{[n]}$ and $\Phi_j^{[n+1]}$ for the discretization of $F^{[n]}(t_{nj}, P(\cdot))$ and $\Phi^{[n+1]}(t_{nj}, P(\cdot))$ respectively. The approximation $P(t)$ of the solution $y(t)$ of (1.1) on $[0, T]$ is then obtained by considering

$$P(t)|_{(t_n, t_{n+1}]} = P_n(t).$$

The method assumes the form

$$\begin{cases} P_n(t_n + sh) = \varphi_0(s)y_{n-1} + \varphi_1(s)y_n + \sum_{j=1}^{m} \chi_j(s)Y_j^{[n]} + \sum_{j=1}^{m} \psi_j(s)(F_j^{[n]} + \Phi_j^{[n+1]}) \\ y_{n+1} = P_n(t_{n+1}), \end{cases} \tag{2.1}$$

where $Y_j^{[n]} = P(t_{n-1,j})$. Thus the algebraic polynomial $P_n(t_n + sh)$ is expressed as linear combination of the basis functions $\varphi_0(s)$, $\varphi_1(s)$, $\chi_j(s)$ and $\psi_j(s)$, $j = 1, 2, \ldots, m$, which are determined from the continuous order conditions provided in [8]. These conditions arise from the analysis of the local truncation error

$$\begin{aligned} \eta(t_n + sh) &= y(t_n + sh) - \varphi_0(s)y(t_n - h) - \varphi_1(s)y(t_n) \\ &- \sum_{j=1}^{m} \Big(\chi_j(s)y(t_n + (c_j - 1)h) + \psi_j(s)y(t_n + c_j h)\Big). \end{aligned} \tag{2.2}$$

and are reported in the following result.

Theorem 2.1. *Assume that the kernel $k(t, \eta, y)$ and the function $g(t)$ in (1.1) are sufficiently smooth. Then the method (2.1) has uniform order p, i.e.,*

$$\eta(t_n + sh) = O(h^{p+1}), \quad h \to 0,$$

for $s \in [0,1]$, if the polynomials $\varphi_0(s)$, $\varphi_1(s)$, $\chi_j(s)$ and $\psi_j(s)$, $j = 1, 2, ..., m$ satisfy the system of equations

$$\begin{cases} 1 - \varphi_0(s) - \varphi_1(s) - \sum_{j=1}^{m} \chi_j(s) - \sum_{j=1}^{m} \psi_j(s) = 0, \\ s^k - (-1)^k \varphi_0(s) - \sum_{j=1}^{m} (c_j - 1)^k \chi_j(s) - \sum_{j=1}^{m} c_j^k \psi_j(s) = 0, \end{cases} \tag{2.3}$$

$s \in [0,1]$, $k = 1, 2, ..., p$.

Two-step collocation methods are obtained by solving the system of order conditions up to the maximum uniform attainable order $p = 2m + 1$, and, in this way, all the basis functions are determined as the unique solution of such system and have degree at most p. However, as observed in [8], it is not convenient to impose all the order conditions because it is not possible to achieve high stability properties (e.g. $A-$stability) without getting rid of some of them. Therefore, *almost* collocation methods are introduced by relaxing a specified number r of order conditions, i.e. by a priori appropriately fixing r basis functions, and determining the remaining ones as the unique solution of the system of order conditions up to $p = 2m + 1 - r$. If r fixed basis functions are polynomials of degree at most p, then the other ones, obtained in the described way, have the same degree. We have many possible different choices for the r fixed basis functions. In principle, they can be written as polynomials depending on their $p + 1$ coefficients, which can be considered as free parameters to be used in order to achieve A-stability. In fact, we are generally able to pursue this purpose by only using few degrees of freedom and, therefore, some coefficients are fixed by imposing some of the interpolation/collocation conditions (3.7)-(3.10), which we will further analyze later. Different choices can be done on the conditions to impose, each giving rise to a different family of TSAC methods, e.g. many $A-$stable methods have been constructed in [8] by imposing the collocation conditions (3.10), i.e. by fixing one or both of the polynomials $\varphi_0(s)$ and $\varphi_1(s)$ as

$$\begin{aligned} \varphi_0(s) &= \prod_{k=1}^{m} (s - c_k)(\alpha_0 + \alpha_1 s + \ldots + \alpha_{p-m} s^{p-m}), \\ \varphi_1(s) &= \prod_{k=1}^{m} (s - c_k)(\beta_0 + \beta_1 s + \ldots + \beta_{p-m} s^{p-m}), \end{aligned} \tag{2.4}$$

where α_j and β_j, $j = 0, 1, \ldots, p - m$, are free parameters. The choices we make in practice in the derivation of the new methods will be clear in Section 5, which regards the effective construction of the numerical methods.

The quadrature formulae in (2.1) are of the form

$$F_j^{[n]} = g(t_{nj}) + h \sum_{\nu=1}^{n} \Big(b_0 k(t_{nj}, t_{\nu-1}, y_{\nu-1}) + \sum_{l=1}^{m} b_l k(t_{nj}, t_{\nu-1,l}, Y_l^{[\nu]}) + b_{m+1} k(t_{nj}, t_\nu, y_\nu) \Big), \tag{2.5}$$

and

$$\Phi_j^{[n+1]} = h \Big(w_{j0} k(t_{nj}, t_n, y_n) + \sum_{l=1}^{m} w_{jl} k(t_{nj}, t_{nl}, Y_l^{[n+1]}) + w_{j,m+1} k(t_{nj}, t_{n+1}, y_{n+1}) \Big), \tag{2.6}$$

where $Y_i^{[n]} = P_{n-1}(t_{n-1,i})$ are the stage values and b_0, b_l, b_{m+1}, w_{j0}, w_{jl}, $w_{j,m+1}$ are given weights.

The polynomial $P(t_n + sh)$ is explicitly defined after solving, at each step, the following system of $(m+1)d$ nonlinear equations in the stage values $Y_i^{[n+1]}$ and y_{n+1}, obtained by computing (2.1) for $s = c_i$, $i = 1, 2, \ldots, m$, and $s = 1$:

$$\begin{cases} Y_i^{[n+1]} &= \varphi_0(c_i) y_{n-1} + \varphi_1(c_i) y_n + \sum_{j=1}^{m} \chi_j(c_i) Y_j^{[n]} + \sum_{j=1}^{m} \psi_j(c_i) \Big(F_j^{[n]} + \Phi_j^{[n+1]} \Big), \\ y_{n+1} &= \varphi_0(1) y_{n-1} + \varphi_1(1) y_n + \sum_{j=1}^{m} \chi_j(1) Y_j^{[n]} + \sum_{j=1}^{m} \psi_j(1) \Big(F_j^{[n]} + \Phi_j^{[n+1]} \Big), \end{cases} \tag{2.7}$$

$n = 1, 2, \ldots, N - 1$.

The starting values y_1 and $Y_i^{[1]}$, $i = 1, 2, \ldots, m$, are assumed to be prescribed and can be computed by using a one-step Runge–Kutta method of order at least p, according to the following theorem.

Theorem 2.2. *Let $e_h(t) := y(t) - P(t)$ be the global error of the TSAC method (2.1), and suppose the hypothesis of Theorem 2.1 are satisfied up to order p. Then, under suitable hypothesis of sufficient regularity on the kernel k (see [8]),*

$$\|e_h\|_{\infty,[t_0,T]} = O(h^{p^*}), \quad h \to 0.$$

i.e. the method has uniform order of convergence $p^* = \min\{s+1, q, p+1\}$, where s and q are such that:

i. *the starting error is $\|e_h\|_{\infty,[t_0,t_1]} = O(h^s)$;*

ii. *the lag-term and increment-term quadrature formulas (2.5)-(2.6) are of order $O(h^q)$.*

Remark 2.3. *In principle, in formula (2.1) we might take the piecewise polynomial of degree greater than p, however we did not follow this approach because first of all, by solving the system of order conditions (2.3) up to $p = 2m+1$, there exists an unique solution given by basis polynomials of degree at most equal to p. Moreover, when $p < 2m+1$, the powers $\{1, s, \dots, s^p\}$ are present in (2.3) and, therefore, it is natural for us to fix the relaxed basis functions of degree at most p in order to solve (2.3) with respect to the others. Moreover, the relaxation technique in practice leads to a decrease of p and, therefore, to a decrease of the degree of the basis functions and we observed that this improves the stability of the resulting methods. Higher degree polynomials, in our analysis, could deteriorate the stability properties of the corresponding methods.*

3. Two-step diagonally implicit almost collocation methods

This section is devoted to the construction of high order methods belonging to the class (2.1) such that the coefficient matrix of the nonlinear system (2.7) has a structured shape, leading to the solution of nonlinear systems of lower dimension d. First of all we consider $w_{j,m+1} = 0$, $j = 1, \dots, m$, in such a way that (2.7) becomes a nonlinear system of dimension md only depending on the stage values $Y_i^{[n+1]}$, $i = 1, \dots, m$, and assumes the following form

$$\begin{cases} Y_i^{[n+1]} - h \sum_{j=1}^{m} \sum_{l=1}^{m} \psi_j(c_i) w_{jl} k(t_{nj}, t_{nl}, Y_l^{[n+1]}) = B_i^{[n]}, \\ y_{n+1} = P_n(t_{n+1}), \end{cases} \tag{3.1}$$

where

$$B_i^{[n]} = \varphi_0(c_i) y_{n-1} + \varphi_1(c_i) y_n + \sum_{j=1}^{m} \chi_j(c_i) Y_j^{[n]} + \sum_{j=1}^{m} \psi_j(c_i) F_j^{[n]} + h \sum_{j=1}^{m} \psi_j(c_i) w_{j0} k(t_{nj}, t_n, y_n). \tag{3.2}$$

By defining

$$Y^{[n+1]} = \left[Y_1^{[n+1]}, Y_2^{[n+1]}, \dots, Y_m^{[n+1]} \right]^T, \quad B^{[n]} = \left[B_1^{[n]}, B_2^{[n]}, \dots, B_m^{[n]} \right]^T,$$

$$\Psi = (\psi_j(c_i))_{i,j=1}^{m}, \quad W = (w_{jl})_{j,l=1}^{m}, \quad K(t_{nc}, t_{nc}, Y^{[n+1]}) = \left(K(t_{ni}, t_{nj}, Y_j^{[n+1]}) \right)_{i,j=1}^{m},$$

the nonlinear system in (3.1) takes the form

$$Y^{[n+1]} - h\Psi(W \cdot K(t_{nc}, t_{nc}, Y^{[n+1]})) = B^{[n]}, \tag{3.3}$$

where $\cdot$ denotes the usual Hadamard product. The tensor form (3.3) clearly shows as the matrices which determine the structure of the nonlinear system (3.1) are Ψ and W. Therefore, in the following subsection we will describe how to choose the basis functions $\psi_j(s)$ and how to modify the quadrature formula (2.6) in order to obtain lower triangular or diagonal structures.

3.1. Determination of the basis functions $\psi_j(s)$

In order to achieve a lower triangular or diagonal structure for the matrix Ψ, the basis functions $\psi_j(s)$ must satisfy

$$\psi_j(c_i) = 0, \text{for } j > i \text{ or } j \neq i \text{ respectively,} \tag{3.4}$$

i.e. $\psi_j(s)$ assumes the form

$$\psi_j(s) = \prod_{k=1}^{j-1}(s - c_k)\bar{\omega}_j(s), \ j = 2, \dots, m \tag{3.5}$$

or

$$\psi_j(s) = \prod_{\substack{k=1 \\ k\neq j}}^{m}(s - c_k)\tilde{\omega}_j(s), \ j = 1, \dots, m \tag{3.6}$$

respectively, where $\bar{\omega}_j(s)$ is a polynomial of degree $p - j + 1$ and $\tilde{\omega}_j(s)$ is a polynomial of degree $p - m + 1$.

Imposing (3.5), the remaining $m + 3$ basis functions can be computed by solving the system of order conditions (2.3) and, as a consequence, the maximum attainable uniform order which can be achieved by the corresponding TSAC methods is $m+2$. On the other hand, by imposing (3.6), the corresponding TSAC methods would have uniform order at most equal to $m + 1$. However, we can follow a different strategy in order to obtain higher order methods. The idea is to impose the conditions (3.4) on less than m basis functions, e.g. one or two of them (generally $\varphi_0(s)$ and $\varphi_1(s)$), in such a way that the maximum attainable order is $p = 2m + r - 1$, with $r = 1, 2$.

Let us define the following sets of interpolation and collocation conditions (see [8]):

- *interpolation conditions in 0*

$$\varphi_0(0) = 0, \ \varphi_1(0) = 1, \ \chi_j(0) = 0, \ \psi_j(0) = 0, \ \forall j, \tag{3.7}$$

- *interpolation conditions in -1*

$$\varphi_0(-1) = 1, \ \varphi_1(-1) = 0, \ \chi_j(-1) = 0, \ \psi_j(-1) = 0, \ \forall j, \tag{3.8}$$

- *interpolation conditions in $c_i - 1$, $i \in \{1, \dots, m\}$*

$$\varphi_0(c_i - 1) = 0, \ \varphi_1(c_i - 1) = 0, \ \chi_j(c_i - 1) = \delta_{ij}, \ \psi_j(c_i - 1) = 0, \ \forall j. \tag{3.9}$$

- *collocation conditions in c_i, $i \in \{1, \dots, m\}$*

$$\varphi_0(c_i) = 0, \ \varphi_1(c_i) = 0, \ \chi_j(c_i) = 0, \ \psi_j(c_i) = \delta_{ij}, \ \forall j, \tag{3.10}$$

Remark 3.1. *The name of conditions (3.7)-(3.8)-(3.9)-(3.10) arises from the fact that they respectively ensure $P_n(t_n) = y_n$, $P_n(t_{n-1}) = y_{n-1}$, $P_n(t_{n-1,i}) = Y_i^{[n]}$, $P_n(t_{ni}) = F_i^{[n]} + \Phi_i^{[n+1]}$. In particular, the last one means that the collocation polynomial exactly satisfies the VIE (1.1) in the collocation point t_{ni}, except from the error associated to the quadrature formulas (2.5) and (2.6).*

Whatever condition from the sets (3.7)-(3.8)-(3.9)-(3.10) is inherited via order conditions, i.e. if we impose that some basis functions satisfy certain interpolation/collocation conditions from the sets (3.7)-(3.8)-(3.9)-(3.10) and derive all the other basis functions by solving the system of order conditions, the same interpolation/collocation conditions are also satisfied by the computed basis functions (and then the corresponding relation in Remark 3.1 is satisfied by the collocation polynomial), as proved in the following theorem.

Theorem 3.2. *Let us define $\xi_1 = -1$, $\xi_2 = 0$, $\xi_{2+j} = c_j - 1$, $\xi_{m+2+j} = c_j$, $j = 1, \dots, m$ and $\Gamma_1(s) = \varphi_0(s)$, $\Gamma_1(s) = \varphi_1(s)$, $\Gamma_{2+j}(s) = \chi_j(s)$, $\Gamma_{m+2+j}(s) = \psi_j(s)$, $j = 1, \dots, m$ and let $i \in \{1, 2, \dots, 2m + 2\}$ be a fixed integer. Then, supposing $\xi_i \neq \xi_j$, $i \neq j$,*

i. If $\Gamma_i(\xi_i) = 1$, then $\Gamma_j(\xi_i) = 0$ for all $j \neq i$;

ii. If $\Gamma_i(\xi_\ell) = 0$ *with* $\ell \neq i$, *then* $\Gamma_j(\xi_\ell) = \delta_{j\ell}$ *for* $j \neq i$.

Proof. The system of order conditions (2.3) can be rewritten in terms of ξ_j and $\Gamma_j(s)$ as

$$s^k - \sum_{k=1}^{2m+2} \xi_j^k \Gamma_j(s) = 0, \quad k = 0, 1, \ldots, p, \tag{3.11}$$

where we assume $\xi_2^0 = 1$. The thesis follows by evaluating (3.11) in $s = \xi_i$ and $s = \xi_\ell$, and by proving that *i.* and *ii.* respectively represent the unique solution of the resulting Vandermonde type linear system. Further details are reported in [11]. ■

Remark 3.3. *In some examples presented in Section 5, we will impose the collocation conditions (3.10) and relax the conditions (3.7)-(3.9), by fixing one or both of the polynomials* $\varphi_0(s)$ *and* $\varphi_1(s)$ *as in (2.4). Then, as a consequence of Theorem 3.2, all the conditions (3.10) are satisfied and* Ψ *reduces to the identity matrix of dimension m.*

3.2. Approximation of the increment term: modified quadrature formula

We observe that the quadrature formulae (2.5) and (2.6) can be obtained by applying the quadrature formulae

$$\int_0^1 f(s)ds \approx b_0 f(0) + \sum_{l=1}^{m} b_l f(c_l) + b_{m+1} f(1), \tag{3.12}$$

$$\int_0^{c_j} f(s)ds \approx w_{j0} f(0) + \sum_{l=1}^{m} w_{jl} f(c_l) + w_{j,m+1} f(1), \tag{3.13}$$

for the approximation of the integrals appearing in the right hand side of

$$\begin{aligned} F^{[n]}(t_{nj}, P(\cdot)) &= g(t_{nj}) + h \sum_{\nu=1}^{n} \int_0^1 k(t_{nj}, t_{\nu-1} + sh, P_{\nu-1}(t_{\nu-1} + sh))ds, \\ \Phi^{[n+1]}(t_{nj}, P(\cdot)) &= h \int_0^{c_j} k(t_{nj}, t_n + sh, P_n(t_n + sh))ds. \end{aligned} \tag{3.14}$$

We aim to derive a suitable modification of the quadrature formula (3.13) in such a way that the matrix W is lower triangular or diagonal and, with the purpose to preserve the order, we make use of some additional quadrature nodes, i.e. we consider quadrature formulae of the form

$$\int_0^{c_j} f(s)ds \approx w_{j0} f(0) + \sum_{l=1}^{m} \tilde{w}_{jl} f(c_l - 1) + \sum_{l=1}^{j} w_{jl} f(c_l), \tag{3.15}$$

where, in case of triangular structure, $\tilde{w}_{jl} = 0$, $l = 1, \ldots, j$ while, in case of diagonal structure, $\tilde{w}_{j1} = 0$ and $w_{jl} = 0$, $l = 1, \ldots, j-1$.

With the purpose of achieving the desired order (see Remark 2.2), quadrature formulae of the form (3.12) and (3.15) can be constructed by taking into account that the order of the corresponding lag term and increment term quadrature formulae is at least $O(h^q)$, if they are interpolatory quadrature formulae on q and $q-1$ nodes respectively [4].

Remark 3.4. *The quadrature formulae (3.12) and (3.15) can be further generalized if we need higher order by considering*

$$\int_0^1 f(s)ds \approx \sum_{l=0}^{\mu_1} b_l f(d_l),$$

$$\int_0^{c_j} f(s)ds \quad \approx \quad \sum_{l=0}^{\mu_0} \tilde{w}_{jl} f(d_{jl}-1) + \sum_{l=1}^{j} w_{jl} f(c_l),$$

where μ_0 and μ_1 depend on the desired order and (3.12) and (3.15) are special cases, obtained by setting $\mu_1 = m+1$, $d_0 = 0$, $d_l = c_l$, $l = 1, 2, \ldots, m$, $d_{m+1} = 1$, $\mu_0 = m$, $d_{j0} = 0$, $\tilde{w}_{j0} = 0$, $d_{jl} = c_l$, $j, l = 1, 2, \ldots, m$.

3.3. Form of the diagonally implicit TSAC methods

As a consequence of the choices reported in the previous subsections, we obtain what follows. If Ψ and W are lower triangular,

$$\Psi = \begin{bmatrix} \psi_{11} & & & \\ \psi_{21} & \psi_{22} & & \\ \vdots & & \ddots & \\ \psi_{m1} & \psi_{m2} & \ldots & \psi_{mm} \end{bmatrix}, \quad W = \begin{bmatrix} w_{11} & & & \\ w_{21} & w_{22} & & \\ \vdots & & \ddots & \\ w_{m1} & w_{m2} & \ldots & w_{mm} \end{bmatrix}, \tag{3.16}$$

the resulting method assumes the form

$$\begin{cases} Y_i^{[n+1]} - h\psi_i(c_i)w_{ii}k(t_{ni}, t_{ni}, Y_i^{[n+1]}) = B_i^{[n]} + \tilde{B}_i^{[n]} + h\sum_{l=1}^{i-1}\sum_{j=l}^{i} \psi_j(c_i)w_{jl}k(t_{nj}, t_{nl}, Y_l^{[n+1]}), \\ y_{n+1} = \varphi_0(1)y_{n-1} + \varphi_1(1)y_n + \sum_{j=1}^{m} \chi_j(1)Y_j^{[n]} + \sum_{j=1}^{m} \psi_j(1)\left(F_j^{[n]} + \Phi_j^{[n+1]}\right), \end{cases} \tag{3.17}$$

where $B_i^{[n]}$ is given by (3.2),

$$\tilde{B}_i^{[n]} = h\sum_{j=1}^{i}\sum_{l=1}^{m} \psi_j(c_i)\tilde{w}_{jl}k(t_{nj}, t_{n-1,l}Y_l^{[n]}), \tag{3.18}$$

and $F_j^{[n]}$, $\Phi_j^{[n+1]}$ are approximations of (3.14) by means of the quadrature formulae (3.12) and (3.15). The solution of the system (3.17) of dimension md can be obtained by solving m successive nonlinear systems of dimension d. Coherently with the case of ODEs, we denote the corresponding methods as diagonally implicit TSAC methods (DITSAC). If Ψ and W are lower triangular and, in addition, their product ΨW is one-point spectrum, i.e.

$$\Psi W = \begin{bmatrix} \lambda & & & \\ \mu_{21} & \lambda & & \\ \vdots & & \ddots & \\ \mu_{m1} & \mu_{m2} & \ldots & \lambda \end{bmatrix}, \tag{3.19}$$

where $\lambda = \psi_i(c_i)w_{ii}$, $i = 1, 2, \ldots, m$ then, in order to solve the system (3.17) by Newton-type iterations, we can repeatedly use the stored LU-factorization of the coefficient matrix

$$I - h\lambda\frac{\partial k}{\partial y}.$$

The related TSAC methods are then called singly diagonally implicit TSAC methods (SDITSAC).

If, in particular, Ψ and W are diagonal

$$\Psi = \begin{bmatrix} \psi_{11} & & & \\ & \psi_{22} & & \\ & & \ddots & \\ & & & \psi_{mm} \end{bmatrix}, \quad W = \begin{bmatrix} w_{11} & & & \\ & w_{22} & & \\ & & \ddots & \\ & & & w_{mm} \end{bmatrix}, \tag{3.20}$$

then the method (3.17) takes the form

$$Y_i^{[n+1]} - h\psi_i(c_i)w_{ii}k(t_{ni}, t_{ni}, Y_i^{[n+1]}) = B_i^{[n]} + \tilde{B}_i^{[n]}, \tag{3.21}$$

where $B_i^{[n]}$ and $\tilde{B}_i^{[n]}$ are given by (3.2) and (3.18), respectively. The nonlinear system (3.21) is then equivalent to m nonlinear systems of dimension d, which can be efficiently solved in a parallel environment. The corresponding methods are denominated diagonal TSAC methods (DTSAC). If ΨW is also one point spectrum, i.e.

$$\Psi W = \lambda I, \tag{3.22}$$

with $\lambda = \psi_i(c_i)w_{ii}$, it can be efficiently treated by means of Newton-type iterations, as in the case of SDITSAC methods: we denote these methods as singly diagonal TSAC methods (SDTSAC).

4. Linear stability analysis

We now focus our attention on the linear stability properties of TSAC methods (3.17) with respect to the basic test equation

$$y(t) = 1 + \lambda \int_0^t y(\tau)d\tau, \quad t \geq 0, \quad \mathrm{Re}(\lambda) \leq 0, \tag{4.1}$$

usually employed in the literature for the stability analysis of numerical methods for VIEs (see [2, 4, 8, 9]). The following result holds.

Theorem 4.1. *The stability matrix associated to the two-step collocation method (3.17) takes the form*

$$R(z) = Q^{-1}(z)M(z), \tag{4.2}$$

where

$$Q(z) = \begin{bmatrix} 1 & -z\psi^T(1)W & -\psi^T(1) & 0 \\ 0 & I - z\Psi W & -\Psi & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

is an invertible matrix for $z < \dfrac{1}{\|\Psi W\|}$ *(for some matrix norm) and*

$$M(z) = \begin{bmatrix} \varphi_1(1) + z\psi^T(1)w_0 & \chi^T(1) + z\psi^T(1)\tilde{W} & 0 & \varphi_0(1) \\ \varphi_1(c) + z\Psi w_0 & A + z\Psi\tilde{W} & 0 & \varphi_0(c) \\ zb_{m+1}u & zub^T & I & zb_0u \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4.4}$$

Proof. The method (3.17) applied to the test problem (4.1) assumes the form

$$\begin{array}{ll} y_{n+1} & = \varphi_0(1)y_{n-1} + (\varphi_1(1) + z\psi^T(1)w_0)y_n + (\chi^T(1) + z\psi^T(1)\tilde{W})Y^{[n]} \\ & \quad +\psi^T(1)F^{[n]} + z\psi^T(1)WY^{[n+1]}, \\ Y^{[n+1]} & = \varphi_0(c)y_{n-1} + (\varphi_1(c) + z\Psi w_0)y_n + (A + z\Psi\tilde{W})Y^{[n]} + \Psi F^{[n]} + z\Psi WY^{[n+1]}, \end{array} \tag{4.5}$$

where we define the column vectors $\psi(1) = \left(\psi_j(1)\right)_{j=1}^m$, $\chi(1) = \left(\chi_j(1)\right)_{j=1}^m$, $w_0 = \left(\tilde{w}_{j0}\right)_{j=1}^m$, $\varphi_0(c) = \left(\varphi_0(c_j)\right)_{j=1}^m$ and $\varphi_1(c) = \left(\varphi_1(c_j)\right)_{j=1}^m$ and the matrix $\tilde{W} = \left(\tilde{w}_{ij}\right)_{i,j=1}^m$. The lag term satisfies the following recurrence relation

$$F^{[n]} = F^{[n-1]} + zb_{m+1}uy_n + zub^TY^{[n]} + zb_0uy_{n-1}, \tag{4.6}$$

with $b = [b_1, b_2, \ldots, b_m]^T$ and $u = [1, \ldots, 1]^T \in \mathbb{R}^m$. By defining

$$v_n = \left[y_n, Y^{[n]}, F^{[n-1]}, y_{n-1}\right]^T,$$

from (4.5) and (4.6), we obtain the following recurrence relation

$$Q(z)v_{n+1} = M(z)v_n, \tag{4.7}$$

where $Q(z)$ and $M(z)$ take the form (4.3) and (4.4), respectively. The proof will be completed by showing the invertibility of the matrix $Q(z)$, by means of some algebraic tools, based on the Schür complement. It is well known (see [1]) that, given a block matrix of the type

$$\left[\begin{array}{c|c} Q_{11} & Q_{12} \\ \hline 0 & Q_{22} \end{array}\right]$$

with invertible blocks Q_{11} and Q_{22}, the inverse assumes the following form:

$$\left[\begin{array}{c|c} Q_{11} & Q_{12} \\ \hline 0 & Q_{22} \end{array}\right]^{-1} = \left[\begin{array}{c|c} Q_{11}^{-1} & -Q_{11}^{-1}Q_{12}Q_{22}^{-1} \\ \hline 0 & Q_{22}^{-1} \end{array}\right].$$

According to this result, the invertibility of the matrix $Q(z)$ follows from the invertibility of the matrix $I - z\Psi W$ thus, if $z\|\Psi W\| < 1$ for some matrix norm (see [1], p. 492), the inverse of $Q(z)$ can be computed as

$$Q^{-1}(z) = \left[\begin{array}{cccc} 1 & z\psi^T(1)W(I - z\Psi W)^{-1} & -\psi^T(1)(I + zW(I - z\Psi W)^{-1}) & 0 \\ 0 & (I - z\Psi W)^{-1} & -(I - z\Psi W)^{-1}\Psi & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]. \tag{4.8}$$

■

The proof of Theorem 4.1 also provides the tools for an efficient inversion of the matrix $Q(z)$. In fact, using the Schür complement, we have reduced the problem of the inversion of $Q(z)$, i.e. a matrix of dimension $2m + 2$, to the inversion of the lower triangular matrix $I - z\Psi W$, of dimension m, as results from (4.8). This allows an efficient computation of the inverse of $Q(z)$.

5. Derivation of *A*-stable (S)DITSAC and (S)DTSAC methods

The strategy we carry out in the construction of A-stable methods can be summarized as follows.

First of all we set the quadrature formulae (3.12) and (3.15) in such a way to enforce the desired structure on the matrix W and the order p of convergence. Moreover, we fix the polynomial $\varphi_0(s)$ and, possibly, also $\varphi_1(s)$, satisfying some of the interpolation/collocation conditions (3.7)-(3.8)-(3.9)-(3.10), e.g. the ones in (2.4). As a consequence some free parameters are available to be spent in order to enforce the desired structure on the matrix Ψ and to achieve A-stability. We next derive the remaining basis functions by solving the system of order conditions (2.3) up to p: as stated by Theorem 3.2, the same fixed interpolation/collocation conditions are inherited by the other basis functions. We next compute the stability polynomial $p(\omega, z)$ of the obtained methods, i.e. the characteristic polynomial of the stability matrix (4.2), which depends on the matrices $Q(z)$ in (4.3) and $M(z)$ in (4.4). In particular, in the construction process it is useful to consider the expession (4.8) of the inverse of the matrix $Q(z)$, which provides considerable simplifications in the computations. We next analyze the stability properties of the polynomial $p(\omega, z)$: in particular, we aim to derive methods which are A-stable, i.e. the roots $\omega_1, \omega_2, \ldots, \omega_{2m+2}$ of the polynomial $p(\omega, z)$ lie in the unit circle, for all $z \in \mathbb{C}$ such that $\mathrm{Re}(z) \leq 0$. We investigate A-stability using the Schur criterion [21], similarly as it has already been done in [8, 11, 12, 13, 14, 15, 18].

Consider the polynomial

$$\eta(w) = d_k w^k + d_{k-1} w^{k-1} + \cdots + d_1 w + d_0,$$

where d_i are complex coefficients, $d_k \neq 0$ and $d_0 \neq 0$. $\eta(w)$ is said to be a Schur polynomial if all its roots w_i, $i = 1, 2, \ldots, k$, are inside of the unit circle. Define

$$\hat{\eta}(w) = \bar{d}_0 w^k + \bar{d}_1 w^{k-1} + \cdots + \bar{d}_{k-1} w + \bar{d}_k,$$

where $\bar{d}_i$ is the complex conjugate of d_i. Define also the polynomial

$$\eta_1(w) = \frac{1}{w}\Big(\hat{\eta}(0)\eta(w) - \eta(0)\hat{\eta}(w)\Big)$$

of degree at most $k - 1$. We have the following theorem.

Theorem 5.1. *(Schur [21]). $\eta(w)$ is a Schur polynomial if and only if*

$$|\hat{\eta}(0)| > |\eta(0)|$$

and $\eta_1(w)$ is a Schur polynomial.

Roughly speaking, the Schur criterion allows us to investigate the stability properties of a k^{th} degree polynomial, looking at the roots of a polynomial of lower degree (i.e. $k-1$). Iterating this process, the last step consists in the investigation of the root of a linear polynomial, plus some additional conditions.

5.1. Examples of methods with $m = 2$ with Ψ and W lower triangular

We first show the construction of highly stable two-stage DITSAC methods (3.17), i.e. we require that the matrices Ψ and W are lower triangular. As a first attempt, we have derived and analyzed the stability properties of (3.17) with $m = 2$ and order $p = 2m + 1 = 5$, and discovered that no A-stable methods within this class exist (for further details, see [11]). Therefore, we next relax one order condition ($r = 1$), and consider DITSAC methods (3.17) with $m = 2$ and order $p = 2m = 4$. We compute the weights of the quadrature formulae (3.12) and (3.15) according to the desired order $p = 4$, obtaining

$$\begin{aligned}
b_0 &= -\frac{-6c_2c_1 + 2c_1 + 2c_2 - 1}{12c_1c_2}, \quad b = \begin{bmatrix} -\frac{1-2c_2}{12(c_1-1)c_1(c_1-c_2)} & \frac{2c_1-1}{12(c_2-1)c_2(c_2-c_1)} \end{bmatrix}^T, \\
b_3 &= -\frac{-6c_2c_1 + 4c_1 + 4c_2 - 3}{12(c_1-1)(c_2-1)}, \quad w_0 = \left[-\frac{c_1(c_1 - 3c_2 + 3)}{6(c_2-1)} - \frac{c_2^2 - 3c_1c_2}{6c_1} \right], \\
W &= \begin{bmatrix} \frac{c_1(2c_1-3c_2+3)}{6(c_1-c_2+1)} & 0 \\ -\frac{c_2^3}{6c_1(c_1-c_2)} & -\frac{2c_2^2-3c_1c_2}{6(c_1-c_2)} \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & \frac{c_1^3}{6(c_1-c_2+1)(c_2-1)} \\ 0 & 0 \end{bmatrix}.
\end{aligned}$$

As in Remark , we assume

$$\varphi_0(s) = s(s - c_1)(s - c_2)(\alpha_0 + \alpha_1 s)$$

and, as a consequence, the matrix $\Psi = I$. Imposing such a factorization on the polynomial $\varphi_0(s)$ implies that it satisfies the interpolation condition $\varphi_0(0) = 0$ and the collocation conditions $\varphi_0(c_1) = \varphi_0(c_2) = 0$. We impose the condition $\alpha_0 = -\alpha_1$, in order to derive methods which do not depend on y_{n-1}: this choice, as also in the case of two-step Runge–Kutta methods for ODEs, is particularly suitable in order to improve the stability properties of the resulting methods (compare with [18, 19]). We next determine the remaining basis functions $\varphi_1(s), \chi_1(s), \chi_2(s), \psi_1(s), \psi_2(s)$ by imposing the system of order conditions (2.3), which result to be

$$\begin{aligned}
\varphi_1(s) &= \frac{(c_1-s)(s-c_2)\big(c_2(c_2+1)\alpha_1(s-1)sc_1^2+(s+c_2((c_2+1)\alpha_1(s-1)s-1)+1)c_1+(c_2-s-1)(s+1)\big)}{(c_1-1)c_1(c_2-1)c_2}, \\
\chi_1(s) &= \frac{(c_1-s)(c_2-s)(c_2((c_1+1)(c_2+1)\alpha_1(s-1)+1)-s-1)s}{(c_1-1)(c_1-c_2-1)(c_1-c_2)}, \\
\chi_2(s) &= \frac{(c_1-s)(c_2-s)(c_1((c_1+1)(c_2+1)\alpha_1(s-1)+1)-s-1)s}{(c_1-c_2)(c_1-c_2+1)(c_2-1)}, \\
\psi_1(s) &= \frac{(c_2-s)s\big(-c_2(c_2+1)\alpha_1(s-1)c_1^2+(s+c_2((c_2+1)\alpha_1(s-1)s-1)+1)c_1+(c_2-s-1)(s+1)\big)}{c_1(c_1-c_2)(c_1-c_2+1)}, \\
\psi_2(s) &= \frac{(c_1-s)s\big(-c_2\alpha_1(c_2-s)(s-1)c_1^2+(c_2(-\alpha_1(c_2-s)(s-1)-1)+s+1)c_1+(c_2-s-1)(s+1)\big)}{(c_1-c_2-1)(c_1-c_2)c_2}.
\end{aligned}$$

The determined quadrature weights and basis functions now depend on the parameters α_1, c_1 and c_2, which can be regarded as degrees of freedom in order to enforce strong stability properties for the corresponding methods, such as A-stability. We also observe that, in force of Theorem 3.2, the interpolation/collocation conditions imposed on $\varphi_0(s)$ are automatically inherited by all the other basis functions and, *a fortiori*, on the whole collocation polynomial (2.1).

We next derive the stability polynomial $p(\omega, z)$ of order $2m + 2 = 6$ with respect to the variable ω. In force of the choices we have made, it takes the form

$$p(\omega, z) = \omega(p_0(z) + p_1(z)\omega + p_2(z)\omega^2 + p_3(z)\omega^3 + p_4(z)\omega^4 + p_5(z)\omega^5), \tag{5.1}$$

where $p_j(z)$, $j = 0, 1, \ldots, 5$, are rational functions with respect to z, which do not depend on the value of the parameter α_1, but only the abscissae c_1 and c_2. In order to investigate on the stability properties of the polynomial (5.1), it is sufficient to consider the polynomial

$$\tilde{p}(\omega, z) = p_0(z) + p_1(z)\omega + p_2(z)\omega^2 + p_3(z)\omega^3 + p_4(z)\omega^4 + p_5(z)\omega^5,$$

of degree 5 with respect to ω. We apply the Schür criterion on the polynomial $\tilde{p}(\omega, z)$, in order to determine the values of the free parameters c_1 and c_2 corresponding to A-stable methods. The result of this analysis is reported in Figure 1.

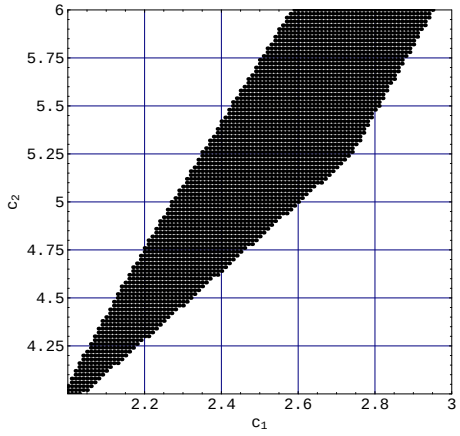

Figure 1: Region of A-stability in the parameter space (c_1, c_2) for DITSAC methods (3.17), with $m = 2$ and $p = 4$, for any value of the parameter α_1

We next derive A-stable two-stage SDITSAC methods within the class (3.17). In this case, by using the Schür criterion, we did not find A-stable methods with $m = 2$ and $p = 4$ and, therefore, we focus our attention on methods with $m = 2$ and $p = 3$, by relaxing two order conditions. We determine the weights of the quadrature formulae (3.12) and (3.15) corresponding to the lower triangular case, obtaining

$$\begin{aligned} b_0 &= -\frac{1-3c_2}{6c_2}, \quad b = \begin{bmatrix} 0 & -\frac{1}{6(c_2-1)c_2} \end{bmatrix}^T, \quad b_3 = -\frac{2-3c_2}{6(c_2-1)}, \\ w_0 &= \begin{bmatrix} \frac{c_1}{2} & \frac{c_2}{2} \end{bmatrix}^T, \quad W = \begin{bmatrix} \frac{c_1}{2} & 0 \\ 0 & \frac{c_2}{2} \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

We next impose

$$\begin{aligned} \varphi_0(s) &= s(\alpha_0 + \alpha_1 s)(s - \alpha_2), \\ \varphi_1(s) &= s(\beta_0 + \beta_1 s)(s - \beta_2) \end{aligned}$$

and, as a consequence, the interpolation condition $\varphi_0(0) = \varphi_1(0) = 0$ holds. In addition, we also set $\alpha_2 = 1$ in order to enforce the independency on y_{n-1}. We next determine the remaining basis functions $\chi_1(s), \chi_2(s), \psi_1(s), \psi_2(s)$ by imposing the system of order conditions (2.3) and which inherit the interpolation condition in 0. The computed basis functions and quadrature weights now depend on the free parameters $\alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2, c_1$ and c_2. First of all, we spend α_0 and β_0 in order to enforce ΨW being lower triangular and one point spectrum. We next enforce some further simplifying assumptions on the basis functions, using the values of α_1, α_2 and β_1, obtaining a three-parameter family of methods to be investigated. We next derive the stability polynomial $p(\omega, z)$ of order $2m + 2 = 6$ with respect to the variable ω, which assumes the same form (5.1), where now $p_j(z)$, $j = 0, 1, \ldots, 5$, depend on the free parameters c_1, c_2 and β_2. As in the previous case, we focus our attention on a polynomial $\tilde{p}(\omega, z)$ of degree 5 with respect to ω. We apply the Schür criterion on $\tilde{p}(\omega, z)$, in order to determine the values of the free parameters c_1, c_2 and β_2 achieving A-stability. The results are shown in Figure 2.

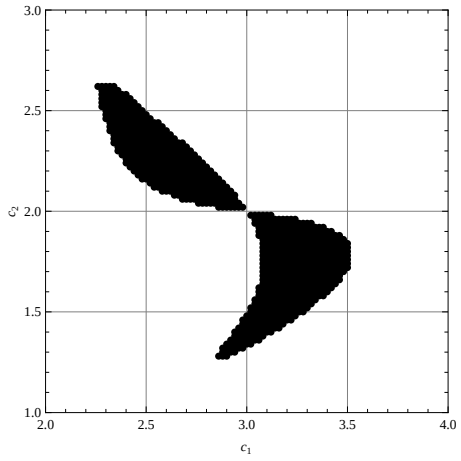

Figure 2: Region of A-stability in the parameter space (c_1, c_2) for SDITSAC methods (3.17), with $m = 2$, $p = 3$ and $\beta_2 = \frac{1}{4}$

5.2. Examples of methods with $m = 2$ with Ψ and W diagonal

We now present the construction of highly stable two-stage DTSAC methods (3.21), i.e. we require that the matrices Ψ and W are diagonal. We first observe that, among the examples of A-stable methods provided in [8], the one reported in Figure 5 belongs to the class of DTSAC methods with $m = 2$ and $p = 3$. In this paper, we present examples of two-stage A-stable SDTSAC methods, requiring that the matrix ΨW is diagonal and one-point spectrum. First of all, we did not find A-stable SDTSAC methods with $m = 2$ and $p = 4, 5$ exist and, therefore, we relax two order conditions ($r = 2$), and consider SDTSAC methods (3.21) with $m = 2$ and order $p = 3$. We compute the weights of the quadrature formulae (3.12) and (3.15) corresponding to the diagonal case, obtaining

$$\begin{aligned} b_0 &= -\frac{1-3c_2}{6c_2}, \quad b = \begin{bmatrix} 0 & -\dfrac{1}{6(c_2-1)c_2} \end{bmatrix}^T, \quad b_3 = -\frac{2-3c_2}{6(c_2-1)}, \\ w_0 &= \begin{bmatrix} \frac{c_1}{2} & \frac{c_2}{2} \end{bmatrix}^T, \quad W = \begin{bmatrix} \frac{c_1}{2} & 0 \\ 0 & \frac{c_2}{2} \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

We next impose

$$\begin{aligned} \varphi_0(s) &= s(\alpha_0 + \alpha_1 s)(s - \alpha_2), \\ \varphi_1(s) &= s(\beta_0 + \beta_1 s)(s - \beta_2) \end{aligned}$$

and, as a consequence, the interpolation condition $\varphi_0(0) = \varphi_1(0) = 0$ holds. In addition, we also set $\alpha_2 = 1$ in order to enforce the independency on y_{n-1}. We next determine the remaining basis functions $\chi_1(s), \chi_2(s), \psi_1(s), \psi_2(s)$ by imposing the system of order conditions (2.3) up to $p = 3$, transferring to them the interpolation condition in 0. Then, At this point, everything depends on the values of $\alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2, c_1$ and c_2. We spend α_0, α_1 and β_0 in order to obtain ΨW being diagonal and one point spectrum. We next enforce some further simplifying assumptions on the basis functions, using the values of α_2 and β_1, obtaining a three-parameter family of methods, depending on c_1, c_2 and β_2. We next derive the stability polynomial $p(\omega, z)$ of order 6 with respect to the variable ω, which assumes the form (5.1), where now $p_j(z)$, $j = 0, 1, \ldots, 5$, are rational functions with respect to z depending on c_1, c_2 and α_2. We apply the Schür criterion, in order to determine the values of the free parameters c_1, c_2 and β_2 achieving A-stability. The results are shown in Figure 3.

6. Numerical experiments

In the numerical experiments we tested the performances of the new diagonally implicit TSAC methods (2.7) in terms of order of convergence and computational cost, compared with the fully implicit TSAC methods (3.17). We

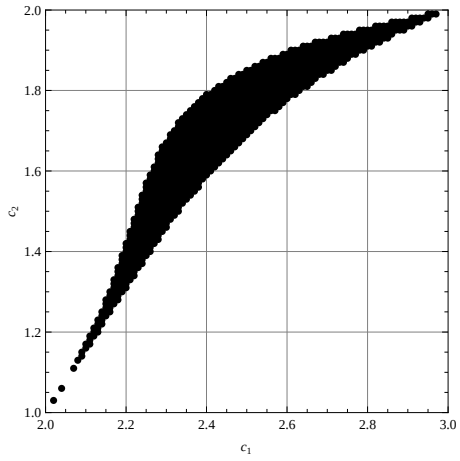

Figure 3: Region of A-stability in the parameter space (c_1, c_2) for SDTSAC methods (3.21), with $m = 2$, $p = 3$ and $\beta_2 = \frac{1}{4}$

considered DITSAC methods leading to a lower triangular coefficient matrix, as these methods are are appropriate to be used in a serial computing environment. As regards the methods with diagonal structure, their advantage can be appreciated in a parallel environment.

We expect a decrease in the computational cost needed for the solution of the nonlinear systems (3.17) and (2.7), while the computational cost of the lag term remains the same, as its reduction is outside the scope of this paper. However a further improvement can be reached by applying fast algorithms also for the lag term computation (see for example [6, 7] in the case of Runge-Kutta and one step collocation methods).

We report here the results obtained on three test problems (taken from [4]) by applying the following A-stable methods having the same order of convergence $p = 4$ and the same abscissa vector $c = [2, 4]^T$, i.e.

- DITSAC4: the lower triangular DITSAC method with $m = 2$ and $p = 4$, provided in Section 5.1, with $\alpha_1 = -1$;

- TSAC4: the fully implicit TSAC method with $m = 2$ and $p = 4$ reported in Section 6 of paper [8].

In the implementation of both methods we have solved the nonlinear systems (3.17) and (2.7) in the stages $Y_i^{[n+1]}$, $i = 1, 2, ..., m$, by using Newton iterations.

The test problems are of the form (1.1), where the kernel k and the forcing function g are reported in Table 6.1 and $T = 2$.

Problem	$g(t)$	$k(t, s, y)$
1	e^t	$2cos(t - s)y$
2	e^{-t}	$e^{s-t}(y + e^{-y})$
3	$2 - \cos(t)$	$-3\sin(ty - s)$

Table 6.1: Test problems.

We compute a numerical estimation of the order of convergence of the methods with the formula

$$p(h) = \frac{cd(h) - cd(2h)}{log_{10}2} \tag{6.1}$$

for a fixed h, where cd is the number of correct significant digits at the end point (the maximal absolute end point error is written as 10^{-cd}). In order to show the improvement in the efficiency of DITSAC4 with respect to TSAC4, we compare the number of kernel and jacobian evaluations needed for the solution of the nonlinar systems (3.17) and (2.7).

The meaning of the headers in the following tables is:

	DITSAC4				TSAC4					
N	cd	p	n_{KEV}	n_{JEV}	cd	p	n_{KEV}	n_{JEV}	r_k	r_J
16	3.7		120	60	3.1		150	120	1.25	2.00
32	4.8	3.6	248	124	4.4	4.0	310	248	1.25	2.00
64	5.9	3.7	504	252	5.6	4.0	630	504	1.25	2.00
128	7.1	3.8	1016	508	6.8	4.0	1270	1016	1.25	2.00
256	8.2	3.9	2040	1020	8.0	4.0	2550	2040	1.25	2.00
512	9.4	4.0	4088	2044	9.2	4.0	5110	4088	1.25	2.00

Table 6.2: Numerical results for DITSAC4 method and TSAC4 method for Problem 1 in Table 6.1

	DITSAC4				TSAC4					
N	cd	p	n_{KEV}	n_{JEV}	cd	p	n_{KEV}	n_{JEV}	r_k	r_J
16	3.6		180	120	3.4		270	240	1.50	2.00
32	4.8	4.0	372	248	4.6	4.0	558	496	1.50	2.00
64	6.0	4.0	693	441	5.8	4.0	1134	1008	1.64	2.29
128	7.1	4.0	1270	762	7.0	4.0	1778	1524	1.40	2.00
256	8.3	4.0	2550	1530	8.1	4.0	3570	3060	1.40	2.00
512	9.5	4.0	5110	3066	9.4	4.0	7154	6132	1.40	2.00

Table 6.3: Numerical results for DITSAC4 method and TSAC4 method for Problem 2 in Table 6.1

- $cd(T/N)$: number of correct significant digits at the end point for $h = T/N$;
- $p(T/N)$: estimated order obtained with the formula (6.1) for $h = T/N$;
- n_{KEV}: number of kernel evaluations;
- n_{JEV}: number of jacobian evaluations;
- r_K: ratio between the number of kernel evaluations needed by the methods TSAC4 and DITSAC4.
- r_J: ratio between the number of jacobian evaluations needed by the methods TSAC4 and DITSAC4.

The tables clearly show as the methods exhibit the same accuracy, confirming the theoretical order of convergence $p = 4$. Moreover the method DITSAC4 needs less kernel and jacobian evaluations than TSAC4. The improvement is much more visible when the problem is nonlinear, as for linear problems Newton method requires just one iteration.

7. Conclusions

We have developed a family of highly-stable two-step almost collocation methods depending on structured coefficient matrices, for the numerical integration of VIEs (1.1). These methods possess uniform order of convergence on the whole integration interval. We have provided examples of A-stable two-stage methods (3.17), where Ψ and W are lower triangular and/or diagonal and, possibly, such that their product is one-point spectrum. Numerical experiments showing the achieved improvement in the computational cost have been reported. Future works will address the construction of highly stable methods (3.17) depending on more stages and their implementation, in order to exploit their properties to get an efficient variable stepsize-variable order implementation and, possibly, in a parallel environment.

	DITSAC4				TSAC4					
N	cd	p	n_{KEV}	n_{JEV}	cd	p	n_{KEV}	n_{JEV}	r_k	r_J
16	4.6		120	60	4.3		258	228	2.15	3.80
32	5.7	3.6	248	124	5.5	3.7	434	372	1.75	3.00
64	6.8	3.8	504	252	6.6	3.8	882	756	1.75	3.00
128	8.0	3.9	1010	502	7.8	3.9	1758	1504	1.74	3.00
256	9.2	4.0	1976	956	9.0	4.0	3082	2572	1.56	2.69
512	10.4	4.0	3066	1022	10.2	4.0	5110	4088	1.67	4.00

Table 6.4: Numerical results for DITSAC4 method and TSAC4 method for Problem 3 in Table 6.1

References

[1] K.E. Atkinson, Introduction to Numerical Analysis, Wiley, New York, 1989.

[2] A. Bellen, Z. Jackiewicz, R. Vermiglio, M. Zennaro, *Stability Analysis of Runge–Kutta methods for Volterra Integral Equations of second kind*, IMA J. Num. Anal. 10, 103–118 (1990).

[3] H. Brunner, *Collocation Methods for Volterra Integral and Related Functional Equations*, Cambridge University Press, 2004.

[4] H. Brunner and P. J. van der Houwen, *The Numerical Solution of Volterra Equations*, CWI Monographs, 3, North-Holland, Amsterdam, 1986.

[5] Butcher, J. C., *Numerical Methods for Ordinary Differential Equations*, 2nd Edition, John Wiley & Sons, Chichester, 2008.

[6] G. Capobianco, D. Conte, I. Del Prete, E. Russo *Fast Runge–Kutta Methods for nonlinear convolution systems of Volterra Integral Equations*, BIT 47/2, 259-275 (2007).

[7] D. Conte, I. Del Prete *Fast collocation methods for Volterra integral equations of convolution type*, J. Comput. Appl. Math. 196/2, 652–663 (2006).

[8] Conte, D., Jackiewicz, Z., and Paternoster, B., Two-step almost collocation methods for Volterra integral equations, *Appl. Math. Comput.* **204** (2008), 839–853.

[9] Conte, D. and Paternoster, B., Multistep collocation methods for Volterra Integral Equations, *Appl. Numer. Math.* **59** (2009), 1721–1736.

[10] D. Conte, B. Paternoster, *A Family of Multistep Collocation Methods for Volterra Integral Equations*, in: AIP Conference Proceedings, Numerical Analysis and Applied Mathematics, T.E.Simos, G. Psihoyios, Ch. Tsitouras (Eds.), vol. 936, pp. 128-131, Springer (2007).

[11] R. D'Ambrosio, *Highly Stable Multistage Numerical Methods for Functional Equations: Theory and Implementation Issues*, Ph.D. Thesis, University of Salerno, in preparation.

[12] D'Ambrosio, R., Ferro, M., Jackiewicz, Z., and Paternoster, B., Two-step almost collocation methods for ordinary differential equations, *Numer. Algorithms* (doi: 10.1007/s11075-009-9280-5).

[13] D'Ambrosio, R. and Jackiewicz Z., Continuous Two–Step Runge–Kutta Methods for Ordinary Differential Equations, *Numer. Algorithms* (doi: 10.1007/s11075-009-9280-5).

[14] D'Ambrosio, R. and Jackiewicz Z., Construction and implementation of highly stable two-step collocation methods, *Numer. Algorithms* (doi: 10.1007/s11075-009-9329-5).

[15] D'Ambrosio, R., Izzo G. and Jackiewicz Z., Search for highly stable two-step Runge–Kutta methods for ordinary differential equations, submitted.

[16] R. D'Ambrosio, B. Paternoster, *Two-step modified collocation methods with structured coefficient matrices for ordinary differential equations*, submitted.

[17] Hairer E., Wanner G., *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, Springer, Berlin, 1991.

[18] Z. Jackiewicz, *General linear methods for ordinary differential equations*, Wiley-Interscience, John Wiley & Sons, to appear.

[19] Jackiewicz, Z. and Tracogna, S., A general class of two-step Runge-Kutta methods for ordinary differential equations, *SIAM J. Numer. Anal.* **32** (1995), 1390–1427.

[20] A. Schädle, M. López-Fernández, Ch. Lubich, *Fast and oblivious convolution quadrature*, SIAM J. Sci. Comput. 28, no. 2, 421–438 (2006).

[21] J. Schur, Über Potenzreihen die im Innern des Einheitskreises beschränkt sind, J. Reine Angew. Math. 147, 205–232 (1916).