ISSN: 1017-060X (Print)

ISSN: 1735-8515 (Online)

Bulletin of the Iranian Mathematical Society

Vol. 43 (2017), No. 5, pp. 1245–1258

Title:

Efficient quadrature rules for a class of cordial Volterra integral equations: A comparative study

Author(s):

H. Majidian

Published by the Iranian Mathematical Society
http://bims.ims.ir

Bull. Iranian Math. Soc.
Vol. 43 (2017), No. 5, pp. 1245–1258
Online ISSN: 1735-8515

EFFICIENT QUADRATURE RULES FOR A CLASS OF CORDIAL VOLTERRA INTEGRAL EQUATIONS: A COMPARATIVE STUDY

H. MAJIDIAN

(Communicated by Touraj Nikazad)

Abstract. A natural algorithm with an optimal order of convergence is proposed for numerical solution of a class of cordial weakly singular Volterra integral equations. The equations of this class appear in heat conduction problems with mixed boundary conditions. The algorithm is based on a representation of the solution and compound Gaussian quadrature rules with graded meshes. A comparative study is carried out, which points out that the proposed method is the most efficient one among other existing methods. In fact, the results of this paper introduce a most-efficient decisive-choice for computing the solution of the heat conduction model.
Keywords: Cordial Volterra integral equation, heat conduction problem, mixed-type boundary condition, compound quadrature rule, graded mesh.
MSC(2010): 65R20.

1. Introduction

In this paper we are concerned with the numerical solution of the linear weakly singular Volterra integral equation

$$y(t) = \int_0^t \frac{s^{\mu-1}}{t^{\mu}} y(s)\,\mathrm{d}s + g(t), \quad t \in (0, T], \tag{1.1}$$

where μ is a positive parameter, and g is a given function of certain class. equation (1.1) has been the subject of many recent researches (see, e.g. [5,6,14, 15] and references therein). Most of these researches are devoted to numerical methods.

Article electronically published on 31 October, 2017.
Received: 19 December 2015, Accepted: 20 May 2016.

Some authors considered a more general class of equations in the form of

$$y(t) = \int_0^t a(t,s)\frac{s^{\mu-1}}{t^{\mu}} y(s)\,\mathrm{d}s + g(t), \quad t \in (0,T], \tag{1.2}$$

where $a(.,.)$ is a smooth function [1, 9]. equations of the form (1.2) lie in turn in a wider class of the so-called *cordial Volterra integral equations* (see [16, 17, 19]). Let $\varphi \in L^1(0,1)$, $a(.,.)$ be a continuous function on $\Delta_T := \{(t,s) : 0 \le s \le t \le T\}$, and $g \in C(\Delta_T \times \mathcal{D})$, where $\mathcal{D} \subset \mathbb{R}$ or $\mathcal{D} \subset \mathbb{C}$ is an open set. The corresponding linear and nonlinear cordial Volterra integral equations are introduced by the operators $V_{\varphi,a}$ and $\mathcal{V}_{\varphi,g}$, respectively, defined on $C[0,T]$ by

$$(V_{\varphi,a}u)(t) = \int_0^t t^{-1}\varphi(t^{-1}s)a(t,s)u(s)\,\mathrm{d}s, \quad 0 \le t \le T,$$

$$(\mathcal{V}_{\varphi,g}u)(t) = \int_0^t t^{-1}\varphi(t^{-1}s)g(t,s,u(s))\,\mathrm{d}s, \quad 0 \le t \le T.$$

Numerical solution of any linear cordial equation, including (1.1), is no longer a challenging problem since we have now rich information about the spectrum and eigenfunctions of the corresponding integral operator (see [16]). Among linear cordial equations, however, the class of equations (1.1) with $\varphi(x) = x^{\mu-1}$ is of practical importance, and despite several proposed methods, the desire for higher accuracy is still needed specially for smaller μ.

In practice, equation (1.1) appears in a heat conduction problem with certain mixed boundary conditions (cf. [2, 13]). To be precise, consider the heat conduction problem

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{a^2}\frac{\partial u}{\partial t}, \quad 0 \le x \le l, \tag{1.3}$$

with the initial condition

$$u(x,0) = 0, \quad 0 \le x \le l, \tag{1.4}$$

and the mixed-type boundary conditions

$$\frac{\partial u}{\partial x}(0,t) - u(0,t) = \phi_1(t), \tag{1.5}$$

$$-\frac{\partial u}{\partial x}(l,t) - u(l,t) = \phi_2(t). \tag{1.6}$$

If we express the solution in terms of a specific single layer ansatz and then apply some suitable transformations, then the above heat problem will be reduced to equation (1.1) (see [13] and references therein).

Among all the existing methods for the case $\mu \le 1$, the quadrature rule [14] and the spline collocation-interpolation method [19] have the optimal convergence rate. Also, Euler's method with extrapolation [13] provides highly accurate approximations at a rather low cost. In this paper, we implement efficient

composite quadrature rules [11], and introduce a new algorithm that is applicable to equation (1.1) with any $\mu > 0$. The main achievement is that the resulting algorithm is the most efficient one among all the existing methods; the convergence order is optimal, it is the most accurate and the most stable, the computational cost is minimal, the assumptions under which the method is applicable to equation (1.1) is also the weakest. Indeed the algorithm is superior to other methods in all aspects.

The main idea of the proposed algorithm comes from [14], where certain representations of the C^1-solution of equation (1.1) are considered. For computing such a solution, these representations are usually ignored by the authors since they are in the form of singular integrals with an algebraic singularity at 0. Even if the m-point Gauss-Christoffel quadrature rule is implemented, one does not achieve a convergence order higher than $\mathcal{O}(m^{-\mu})$. In this paper, we follow the idea of our previous work [14], where certain class of composite quadrature rules on uniform meshes is considered. If suitable order-preserving meshes (cf. [10, 11]) are considered instead, one achieves an algorithm with an optimal order of convergence under weaker assumptions.

Although, at first look it seems that we have just implemented some *well-known* quadrature rules on the problem, and this is a trivial task, the paper have something more to say: It unveils that choosing this strategy in practice leads to the most efficient algorithm for numerical solution of the problem, a practically important fact that had not been discovered by the earlier authors.

The remainder of this paper is organized as follows. After introducing the algorithm in Section 2, we carry out a comparative study in Section 3, which points out that our proposed method is superior to other existing methods from various aspects. In section 4, we give further numerical experiments, which illustrate the capability and efficiency of the method. Finally, we give some conclusions.

2. An algorithm with optimal rate of convergence

The following representation theorem is due to [7] and [9] and plays a fundamental role in the algorithm presented in this paper.

Theorem 2.1. *(a) If $\mu > 1$ and the function g belongs to $C^r[0,T]$ for some positive integer r, then equation (1.1) possesses a unique solution $u \in C^r[0,T]$, that can be represented as*

$$u(t) = g(t) + t^{1-\mu} \int_0^t s^{\mu-2} g(s)\, \mathrm{d}s. \tag{2.1}$$

(b) If $\mu = 1$ and $g \in C^r[0,T]$, $r \geq 1$, then equation (1.1) has a family of solutions in $C[0,T]$ of which only one has C^1 continuity. The unique C^1 solution

of equation (1.1) *can be represented explicitly as*

$$u(t) = g(t) + \int_0^t s^{-1} g(s)\,\mathrm{d}s. \tag{2.2}$$

(c) If $0 < \mu < 1$ *and* $g \in C^r[0,T]$, $r \geq 1$, *then equation* (1.1) *has a family of solutions in* $C[0,T]$ *of which only one has* C^1 *continuity. The unique* C^1 *solution of equation* (1.1) *can be represented explicitly as*

$$u(t) = g(t) + \frac{g(0)}{\mu - 1} + t^{1-\mu} \int_0^t s^{\mu-2}(g(s) - g(0))\,\mathrm{d}s. \tag{2.3}$$

While other authors focused on efficient projection or discretization methods for the Volterra operator of (1.1), the idea of the author is to consider the representations proposed by Theorem 2.1 and compute the involved integrals efficiently [14]. The originality of the present paper is that we enhance the latter task by employing more efficient compound Gaussian quadrature rules [10,11].

By the variable transformation $s = xt$, we have

$$\int_0^1 x^{\mu-2} f_t(x)\,\mathrm{d}x = \begin{cases} t^{1-\mu} \int_0^t s^{\mu-2} g(s)\,\mathrm{d}s, & \text{if } \mu > 1, \\ t^{1-\mu} \int_0^t s^{\mu-2}(g(s) - g(0))\,\mathrm{d}s, & \text{if } 0 < \mu \leq 1, \end{cases} \tag{2.4}$$

where $f_t(x) = g(xt)$ if $\mu > 1$, and $f_t(x) = g(xt) - g(0)$ if $0 < \mu \leq 1$. Note that $x^{-1}(g(xt) - g(0))$, as a function of x, is always continuous on $[0,T]$ due to the continuity of g'. Thus, considering the representations (2.1) or (2.3), the problem of approximating the solution of (1.1) may be reduced to that of approximating (possibly singular) integrals of the form

$$\int_0^1 x^{\alpha-1} f(x)\,\mathrm{d}x, \quad \alpha > 0. \tag{2.5}$$

Köhler [11] and Kaneko et al. [10], independently and almost simultaneously, noticed that the compound Gaussian quadrature rules can compute weakly singular integrals as accurate as the regular ones if the integration interval is partitioned by suitable graded meshes. It is difficult to say any words about the primacy since neither of the papers cited the other one. The authors in [10] considered only the Gaussian rule, while the author of [11] considered any quadrature rules in general. In the following, we give an overview of these two papers and their main results.

Consider the integral

$$I[f] := \int_a^b w(x) f(x)\,\mathrm{d}x, \tag{2.6}$$

where f and/or the weight function w are allowed to have a singularity at the endpoint a.

2.1. **A review to the paper of Köhler (1994).** Let

$$a = z_0 < z_1 < \cdots < z_n = b \tag{2.7}$$

be a partition of the integration interval, and let

$$0 \le x_1 < \cdots < x_m \le 1$$

be nodes of an elementary quadrature formula on $[0,1]$. The nodes of the compound quadrature formula are then $x_{i,j} = z_{i-1} + x_j h_i$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, where $h_i := z_i - z_{i-1}$. The compound quadrature rule is then defined by

$$Q_n[f] = \sum_{i=1}^{n} Q^i[f], \quad \text{and} \quad Q^i[f] = \sum_{j=1}^{m} a_{i,j} f(x_{i,j}), \tag{2.8}$$

where

$$a_{i,j} = \int_{z_{i-1}}^{z_i} w(x) l_j \left(\frac{x - z_{i-1}}{h_i} \right) \mathrm{d}x \tag{2.9}$$

are the weights corresponding to the elementary quadrature rule in each panel. Here, $l_j(x) = \prod_{\nu=1,\nu\neq j}^{m} (x - x_\nu)/(x_j - x_\nu)$. Indeed, $f(x)$ is replaced in each panel by an interpolation polynomial of degree $m-1$. If $x_{1,1} = a$, and f has an unbounded singularity at a, then the first summand in (2.8) is omitted.

In [11], it has been shown that the order of the selected elementary quadrature rule can be preserved in the presence of singularities if one uses the compound quadrature rule with an appropriate partition (2.7). More precisely, we have the following result.

Without loss of generality, let $a = 0$ and $b = 1$. Assume that the quadrature rule, applied in each panel, is exact for any polynomial p of degree less than $k := m + \mu$ for some $\mu \ge 0$.

Theorem 2.2. *Assume that f is k times differentiable on $(0,1]$, and*

$$\begin{aligned}
\left| w^{(i)}(x) \right| &\le c_1 x^{\alpha_1 - i} \left| \ln d_1 x \right|^{\alpha_2}, \quad x \in (0,1], \quad i = 0, \ldots, \mu, \\
\left| f^{(i)}(x) \right| &\le c_2 x^{\beta_1 - i} \left| \ln d_1 x \right|^{\beta_2}, \quad x \in (0,1], \quad i = m, \ldots, k, \\
|f(x)| &\le c_2 x^{\beta_1} \left| \ln d_1 x \right|^{\beta_2}, \quad x \in (0,1],
\end{aligned}$$

where $\alpha_1 > -1$, $\alpha_2 \in \mathbb{R}$, $\beta_1 > -1$, $\beta_2 \in \mathbb{R}$, and $0 < d_1 < 1$. Let the mesh points z_i be defined by $z_i := (i/n)^q$, where

$$q > \frac{k}{1 + \alpha_1 + \beta_1}.$$

Then the absolute error of the compound quadrature rule is $\mathcal{O}(n^{-k})$.

2.2. **A review to the paper of Kaneko et al. (1994).** Similar results can be found in [10], but with less generality. Indeed, the authors of [10] has been considered only the Gaussian quadrature rule, while the results of [11] apply to any kind of interpolatory quadrature rule.

Let $S \subset [0,1]$ contains a finite number of points. Define

$$\omega_S(x) = \inf\{|x-t| : t \in S\},$$

measuring the distance of x from the set S.

For $\alpha > -1$ and an integer $k \geq 0$, a real-valued function f is said to be of Type(α, k, S) if

$$\left|f^{(k)}(x)\right| \leq C\,[\omega_S(x)]^{\alpha-k}, \quad x \notin S,$$

and $f \in C^k([0,1] \setminus S)$.

Now consider the integral (2.6). At first, assume that f is allowed to has a bounded singularity at a. Then, in the subinterval $[a, z_1]$, we replace $f(x)$ by its linear interpolant at the points a and z_1. In other panels, we use the m-point Gaussian quadrature rule w.r.t. the weight function $w(x)$. Then we have the following result (see [10]).

Theorem 2.3. *Let $w \in L_1[a,b]$, and*

$$f \in \text{Type}(\alpha, 2m, \{a\}) \cap C[a,b]$$

with $\alpha > 0$. Consider the graded mesh $z_i = (i/n)^q$ with $q = 2m/\alpha$. Then the absolute error of the compound m-point Gaussian quadrature rule is $\mathcal{O}(n^{-2m+1})$.

In a more general case, when f is allowed to has an unbounded integrable singularity at a, we replace $f(x)$ simply by zero in $[a, z_1]$. Then we have the following result (see [10]).

Theorem 2.4. *Let $w \in L_\infty[a,b]$, and*

$$f \in \text{Type}\,(\alpha, 2m, \{a\}) \cap C[a,b]$$

with $\alpha > 0$. Consider the graded mesh $z_i = (i/n)^q$ with $q = (2m+1)/(\alpha+1)$. Then the absolute error of the compound m-point Gaussian quadrature rule is $\mathcal{O}(n^{-2m})$.

2.3. **Connections.** In order to see the analogy between Theorem 2.2 and Theorem 2.4, we let $w \equiv 1$ and let $x_1, \ldots, x_m$ be nodes of m-point Gaussian rule in $[-1,1]$. Then, Theorem 2.2 is simplified as follows.

Corollary 2.5. *Assume that f is $2m$ times differentiable on $(0,1]$, and*

$$\left|f^{(i)}(x)\right| \leq c_2 x^{\beta_1 - i} \left|\ln d_1 x\right|^{\beta_2}, \quad x \in (0,1], \quad i = m, \ldots, 2m,$$

$$|f(x)| \leq c_2 x^{\beta_1} \left|\ln d_1 x\right|^{\beta_2}, \quad x \in (0,1],$$

where $\beta_1 > -1$, $\beta_2 \in \mathbb{R}$, $0 < d_1 < 1$. *Let the mesh points* z_i *be defined by* $z_i := (i/n)^q$, *where*

$$q > \frac{2m}{1+\beta_1}.$$

Then the absolute error of the compound m*-point Gaussian quadrature rule is* $\mathcal{O}(n^{-2m})$.

In Corollary 2.5, if we let $\beta_2 = 0$, then we arrive at a result that is very similar to Theorem 2.4, though the former is still more general than the latter. Indeed, in Theorem 2.4, the grading exponent q takes one value which depends on α, the degree of regularity of f, while in Corollary 2.5, q takes any value being large enough.

2.4. **The algorithm.** Considering the representations (2.1)-(2.3) and Corollary 2.5, our proposed algorithm for local approximation of the unique C^1 solution of equation (1.1) can be summarized as follows.

For any given $t > 0$, define the function g_μ on $(0, 1]$ as

$$g_\mu(x) := \begin{cases} x^{\mu-2} g(xt), & \text{if } \mu \geq 1, \\ x^{\mu-2}(g(xt) - g(0)), & \text{if } 0 < \mu < 1. \end{cases} \tag{2.10}$$

Then, the approximated solution at t, $u_{n,m}(t)$, is described by

$$u_{n,m}(t) = \begin{cases} g(t) + \sum_{i=1}^{n} Q^{(G;i)}[g_\mu], & \text{if } \mu \geq 1, \\ g(t) + \dfrac{g(0)}{\mu - 1} + \sum_{i=1}^{n} Q^{(G;i)}[g_\mu], & \text{if } 0 < \mu < 1, \end{cases} \tag{2.11}$$

where $Q^{(G;i)}[g_\mu]$ is the m-point Gauss-Legendre quadrature rule on the subinterval $[z_{i-1}, z_i]$, as defined in Corollary 2.5.

Moreover, Corollary 2.5 implies that the above algorithm converges with an optimal rate.

3. Comparisons

In this section we compare from various aspects the proposed algorithm (2.11) with some efficient existing methods for equation (1.1) with $\mu < 1$.

3.1. **Euler's method with extrapolation.** In the Euler's method for equation (1.1) (cf. [12, 13]), a uniform grid $0 = t_0 < \cdots < t_n = T$ with stepsize $h := T/n$ is considered. The local approximate solution $u_i^h \simeq u(t_i)$, where u is the unique solution specified by Theorem 2.1, is computed recursively by

$$u_i^h = g(t_i) + \frac{1}{t_i^\mu} \sum_{j=0}^{i-1} D_j u_j^h, \quad i = 1, \ldots, n, \tag{3.1}$$

with a known $u_0^h = u(0)$ and

$$D_j := \frac{t_{j+1}^{\mu} - t_j^{\mu}}{\mu}, \quad j = 0, \dots, n-1. \tag{3.2}$$

If $t^{-\alpha} g(t) \in C^2[0,T]$, for some $\alpha \in \mathbb{R}$, then some error estimates have been obtained in [13], which allow an application of a general extrapolation procedures, e.g. the E-algorithm [3]. The numerical results are very satisfactory even for small μ. However, the algorithm (2.11) is still preferred in many essential aspects.

Limitations: The method of [13] is applicable when $\bar{g}(t) := t^{-\alpha} g(t)$, for some $\alpha \in \mathbb{R}$, is at least two times differentiable on $[0,T]$, and $\bar{g}(0) \neq 0$. The method (2.11), however, does not need any smoothness of $g(x)$ at the endpoint $x = 0$. For an example, the numerical results in Section 4 show the algorithm (2.11) is successfully applicable to the function $g(t) = t^{1.5}(1 + \ln t)$, $t \in [0,1]$; for any α, however, the derivative of $t^{-\alpha} g(t)$ at $t = 0$ either vanishes or does not exist.

Complexity: In the Euler's method (3.1) for approximating n instances of the solution at $t_1, \dots, t_n$, one needs evaluation of g at n distinct points, evaluation of the power function t^{μ} at n distinct points, $n(n+3)/2-1$ multiplications, and $n(n+3)/2-1$ additions. If one decides to apply the E-algorithm, the method should be performed for some increasing n, which increase the complexity. If n is doubled in each row of the E-algorithm, the complexity reduces since the coarser grid is included in the finer one, and then g as well as the power function needs to be evaluated only on the coarsest grid.

It seems the method (2.11) provides an approximation of the solution only at one instance point t, and if one needs an approximation at another point, the process should be performed from the beginning. The following result shows that only one performance of the method can provide an approximation on the whole graded mesh.

Corollary 3.1. *Let all assumptions of Corollary 2.5 hold. Then the absolute error of the compound m-point Gaussian quadrature rule, when applied to the integral*

$$\int_0^{z_k} w(x) f(x)\, \mathrm{d}x, \tag{3.3}$$

is of order $\mathcal{O}(k^{-2m})$, for $k = 2, \dots, n$.

Proof. Apply to (3.3) the change of variables $x = z_k t$ and consider the partition $t_i := (i/k)^q$, $i = 0, \dots, k$, where q is the same. Now, it is enough to apply Corollary 2.5 with z_i replaced by t_i. Then the result follows since $z_i = t_i z_k$. □

Without loss of generality, consider equation (1.1) on $[0,1]$. If $\mu < 1$, then the desired solution at $t = 1$, represented by (2.3), is approximated by using

the formula (2.8):

$$u_n^n := g(1) + \frac{g(0)}{\mu - 1} + \sum_{i=1}^{n} Q^i[g_\mu], \tag{3.4}$$

with $g_\mu(x) = x^{\mu-2}(g(x) - g(0))$. If one compute $Q^i[g_\mu]$, for $i = 1, \ldots, n$, then the approximation at z_k, $k = 2, \ldots, n-1$, is also available:

$$u_k^n := g(z_k) + \frac{g(0)}{\mu - 1} + \sum_{i=1}^{k} Q^i[g_\mu]. \tag{3.5}$$

Thus, $u_2^n, \ldots, u_n^n$ can be computed recursively. Since computing $Q^i[g_\mu]$ for each i requires m multiplications and $m-1$ additions, the total process requires at most mn function evaluations, $mn+1$ multiplications, and $m(n+1)$ additions. In practice one takes m to be a fixed and usually small (i.e. $m \ll n$) integer and let n increases until the desired accuracy is achieved. There are many tables in the literature for abscissas and weights of the Gauss-Legendre quadrature rule (e.g. [4]), so one does not need to pay any cost for computing these values (one needs only m multiplications and n additions for the linear transformation of the abscissas into the panels and m multiplications for computing the new weights, $hw_j/2$). Thus, the complexity of the method (2.11) is considerably less than that of Euler's method, which has been shown above to be $\mathcal{O}(n^2)$. The same discussion holds for the case when $\mu \geq 1$.

A numerical experiment: As a numerical comparison, we have solved the following example of (1.1) by using the method (2.11) and compare it with the solution given in [13]:

$$\begin{cases} g(t) = t^{0.3}(1+t), \\ \mu = 0.5, \end{cases}$$

with the exact solution $u(t) = -4t^{0.3} + 2.25t^{1.3}$. If the Euler's method with $n = 40 \times 2^j$, for $j = 0, \ldots, 5$, is employed, and then the E-algorithm, as described in [13], is applied, the absolute error is reduced to 1.01×10^{-10} (cf. [13, Table 3]). According to the above discussion on the complexity of the Euler's method, this procedure takes $2,560$ function evaluations, at least $1,095,780$ multiplications, and at least $1,095,780$ additions.

The representation (2.3) is valid when g is at least in $C^1[0,1]$, so the method (2.11) can not directly be applied to the above example. However, it is easy to see that $t^{0.3}\bar{u}(t)$, $\bar{u} \in C^1[0,1]$, is the solution to equation (1.1) iff $\bar{u}(t)$ is its unique solution with $\bar{g}(t) = 1 + t$. Then by the method (2.11) with $m = 9, n = 128$, we have achieved a higher accuracy with the absolute error 5.73×10^{-11}. This costs only 1152 function evaluations, 1171 multiplications, and 1289 additions.

3.2. **Modified Euler's method on graded meshes.** In Euler's method, $u(s)$ is approximated on each panel by $u(t_j)$, while in the modified Euler's method, introduced in [15], it is approximated by $(u(t_{j+1})+u(t_j))/2$. This increases the complexity of the Euler's method even more, though the accuracy enhances. Our numerical experiences, however, show that the direct method of this paper is still far more accurate. For example, if we take $g(t) = t^{1.5} + t + 1$ and $\mu = 0.5$, then the absolute error of the modified Euler's method on a graded mesh is $6.98E$-7, when the number of panels n is 20480 (see [15, Table 3]). However, the absolute error by the method (2.11) with $n = 32$ and $m = 5$ will be $5.52E$-8. This costs only 160 function evaluations, 171 multiplications, and 197 additions.

3.3. **Composite quadrature rules on uniform meshes.** The first direct method based on the representations (2.1)–(2.3) of the solution is the composite quadrature rule [14], which is applied on uniform meshes. The complexity and the accuracy of the method are comparable to those of the direct method (2.11). However, some extra convergence conditions (cf. [14, Theorems 6 and 8]) should be imposed on it, which in turn impose limitations on its applicability. For example, let $g(t) = (1-2t)^{7/3}/(1+t^2)$, $\mu = 0.1$, and $T = 1.55$. The absolute error of the method of [14] with $n = 160, m = 5$ is 39.15 (cf. [14, Table 4]), while it is near $2.97E$-6 by the method of this paper with $n = 8, m = 5$.

Even if the convergence conditions of the direct method of [14] are satisfied, our new method is still far more accurate, since its convergence is of optimal order under very weak conditions. For example, let $g(t) = t^3((21/31)\ln t + 100/961)$, $\mu = 0.1$, and $T = 1.55$. The absolute error of the method of [14] with $n = 160, m = 5$ is $4.48E$-5 (cf. [14, Table 3]), while it is near $1.26E$-10 by the method of this paper with $n = 8, m = 5$.

3.4. **Spline collocation method.** Spline collocation methods have been developed for linear and nonlinear cordial Volterra integral equations (see [18,19]), so they can be employed to solve equation (1.1) for any $\mu > 0$. Although, convergence of the method is of optimal order (cf. [18, Theorem 4.2]), its complexity is not comparable to that of the rule (2.11). If $g \in C^m[0,T]$, then the method consists of recurrent solving of n linear systems, each one of order m, where $n = T/h$ and $h > 0$ is a small step size. In this case, the error decays as $\mathcal{O}(h^m)$, for decreasing h. The $(i+1)$-th system $(i = 0, \dots, n-1)$ corresponds to the matrix $I_m - D_i$, where I_m is the identity matrix of order m, and D_i is an $m \times m$-matrix with the entries

$$D_i = \left(d_{j,k}^{i,i}\right)_{j,k=1}^{m}.$$

Each entry $d_{j,k}^{i,i}$ is a definite integral and, in general, should be computed by quadrature rules. The right-hand-side of the system is also determined by $m^2 i$ similar integrals, $d_{j,k}^{i,l}$, $l = 0, \dots, i-1$, $j = 1, \dots, m$, $k = 1, \dots, m$. If calculation

of each $d_{j,k}^{i,l}$ requires r flops, then we need $m^2(i+1)r$ flops for constructing the i-th system; so in total we need $m^2n(r(n+3)/2+1)$ flops. After constructing the systems, additional $\mathcal{O}(m^2n)$ flops are needed for solving them.

Despite the method is rather costly, its accuracy is considerably less than that of the rule (2.11). For example, consider equation (1.1) with $g(t) = t^{9/2}$, $\mu = 0.1$, and $T = 1$. Since $g \in C^4[0,1]$, one can use the spline collocation method with $m = 4$. By choosing four Lobatto points as collocation parameters, the absolute errors at $t = 1$ will be $1.3E$-3, $1.9E$-4, $9.0E$-6 for $n = 8, 16, 32$, respectively. This is while the error of the proposed rule (2.11) with $(m, n) = (4, 4)$ will be $7.1E$-10.

Note also that conditions under which the rule (2.11) converges with an optimal order is weaker than the corresponding conditions for the spline collocation method. In the latter method, the parameter m depends on the smoothness degree of g in $[0, T]$, while in our proposed rule, it does not depend on the differentiability of g at 0 (see Corollary 2.5). For instance, in the above example with $g(t) = t^{9/2}$, one can employ the rule with any integer $m > 0$, while in the spline collocation method m can not exceed 4.

The applicability of the spline collocation method, i.e. the solvability of all of the systems $I_m - D_i$ $(i = 0, \dots, n-1)$, is another important issue. According to [18], the applicability conditions reduce to

$$\det(I_m - D_i) \neq 0, \tag{3.6}$$

for only some small values of i. It has been proved that the piecewise linear collocation method ($m = 2$) is always applicable (cf. [8, Theorem 5]). Also the piecewise quadratic collocation method is applicable when 0 is chosen as the first collocation parameter (cf. [8, Theorem 8]). For larger m, there is a conjecture (not proved yet) that the spline collocation method with any choice of collocation parameters is always applicable to equation (1.1).

Remark 3.2. The first tree methods mentioned above are devoted to equation (1.1), while the spline collocation method has been extended to wider class of cordial Volterra equations, either linear or nonlinear (cf. [19]). The above discussion shows that to specifically solve the special equation (1.1), our proposed algorithm is superior to the spline collocation method. Certainly, for other equations of the large class of cordial Volterra equations, the proposed algorithm is unapplicable, but this is not an important matter in the real world problems; yet, we do not know any applications of these equations in engineering, physics, etc.

4. **A numerical example**

Consider the C^1 solutions at $t = 1$ of equation (1.1) with $g(t) = t^{1.5}(1 + \ln t)$ on $[0, 1]$, for $\mu = 0.1$ and $\mu = 0.01$. In order to illustrate the theoretical results of this paper, we apply the algorithm (2.11) to this example.

TABLE 1. Absolute errors and numerical orders of convergence for the sample equation with $g(t) = t^{1.5}(1 + \ln t)$ and $\mu = 0.1$.

n	$m = 2$	$m = 3$	$m = 4$	$m = 5$
10	1.56×10^{-2}	3.82×10^{-3}	1.58×10^{-3}	9.51×10^{-4}
20	1.81×10^{-3}	1.42×10^{-4}	1.94×10^{-5}	4.08×10^{-6}
40	1.83×10^{-4}	4.12×10^{-6}	1.64×10^{-7}	1.01×10^{-8}
80	1.69×10^{-5}	1.04×10^{-7}	1.13×10^{-9}	1.91×10^{-11}
160	1.47×10^{-6}	2.40×10^{-9}	6.88×10^{-12}	3.08×10^{-14}
320	1.22×10^{-7}	5.19×10^{-11}	3.88×10^{-14}	4.51×10^{-17}
640	9.73×10^{-9}	1.07×10^{-12}	2.07×10^{-16}	6.17×10^{-20}
1280	7.58×10^{-10}	2.15×10^{-14}	1.06×10^{-18}	8.06×10^{-23}
2560	5.78×10^{-11}	4.19×10^{-16}	5.25×10^{-21}	1.01×10^{-25}
Rate	3.7	5.7	7.7	9.6

TABLE 2. Absolute errors and numerical orders of convergence for the sample equation with $g(t) = t^{1.5}(1 + \ln t)$ and $\mu = 0.01$.

n	$m = 2$	$m = 3$	$m = 4$	$m = 5$
10	4.23×10^{-2}	1.38×10^{-2}	7.27×10^{-3}	5.40×10^{-3}
20	5.30×10^{-3}	5.72×10^{-4}	1.06×10^{-4}	2.97×10^{-5}
40	5.61×10^{-4}	1.78×10^{-5}	9.79×10^{-7}	8.31×10^{-8}
80	5.35×10^{-5}	4.67×10^{-7}	7.11×10^{-9}	1.67×10^{-10}
160	4.74×10^{-6}	1.11×10^{-8}	4.49×10^{-11}	2.81×10^{-13}
320	3.99×10^{-7}	2.45×10^{-10}	2.59×10^{-13}	4.23×10^{-16}
640	3.24×10^{-8}	5.14×10^{-12}	1.40×10^{-15}	5.91×10^{-19}
1280	2.55×10^{-9}	1.04×10^{-13}	7.28×10^{-18}	7.82×10^{-22}
2560	1.96×10^{-10}	2.05×10^{-15}	3.64×10^{-20}	9.95×10^{-25}
Rate	3.7	5.7	7.6	9.6

By (2.10),

$$g_\mu(x) = x^{\mu-0.5}(1 + \ln x), \tag{4.1}$$

for $x \in (0, 1]$. Now, Corollary 2.5 should be applied to g_μ, as suggested by the algorithm.

If $\mu = 0.1$, then $g_\mu(t)$ satisfies conditions of Corollary 2.5 with $\beta_1 = -0.40$. Since g_μ is m times differentiable on $(0, 1]$ for each $m \geq 1$, one should take $q > 2m/(1 - 0.40)$ to achieve the convergence rate of $\mathcal{O}(n^{-2m})$. For $\mu = 0.01$, we have the same discussion with $\beta_1 = -0.49$. Then, we should take $q > 2m/(1-0.49)$. Table 1 shows the results. As it is seen, the rates of convergence (see the last row) are in accordance with the theoretical order of convergence $\mathcal{O}(n^{-2m})$, for some choices of m. As it is seen, despite low smoothness degree of $g(t)$ and smallness of μ, the accuracy of the method is rather high. By

the discussion given in Section 3 and the results of Table 1, we can certainly claim that the algorithm (2.11) is the most efficient algorithm proposed in the literature for numerical solution of equation (1.1).

5. Conclusions

We have suggested a simple and highly accurate algorithm for numerical solution of equation (1.1) even when the parameter μ is very small. The method is based on a representation of the solution that contains an integral term with one algebraic singularity at zero. If traditional quadrature methods are applied to this integral, then we obtain a low rate of convergence for smaller μ. However, if the compound Gaussian quadrature rule with a suitable graded mesh is applied, one obtains an optimal rate of convergence under some rather weak conditions. The complexity of the algorithm is rather low, while its accuracy is higher than all other existing methods. Moreover, the rate of convergence is optimal under rather weak conditions on g. These properties make the algorithm as the most-efficient decisive-choice for computing the solution of the heat conduction model (1.1).

References

[1] P. Baratella, A Nyström interpolant for some weakly singular linear Volterra integral equations, *J. Comput. Appl. Math.* **231** (2009), no. 2, 725–734.

[2] M.A. Bartoshevich, On a heat conduction problem, *Inž h.-Fiz. Žh.* **28** (2008), no. 2, 340–346.

[3] C. Brezinski, A general extrapolation algorithm, *Numer. Math.* **35** (1980), no. 2, 175–187.

[4] P. Davis and P. Rabinowitz, Abscissas and weights for Gaussian quadratures of high order, *J. Res. Nat. Bur. Standards* **56** (1956), no. 1, 35–37.

[5] T. Diogo, Collocation and iterated collocation methods for a class of weakly singular Volterra integral equations, *J. Comput. Appl. Math.* **229** (2009), no. 2, 363–372.

[6] T. Diogo and P. Lima, Superconvergence of collocation methods for a class of weakly singular Volterra integral equations, *J. Comput. Appl. Math.* **218** (2008), no. 2, 307–316.

[7] T. Diogo, S. McKee and T. Tang, A Hermite-type collocation method for the solution of an integral equation with a certain weakly singular kernel, *IMA J. Numer. Anal.* **11** (1991), no. 4, 595–605.

[8] T. Diogo and G. Vainikko, Applicability of spline collocation to cordial Volterra equations, *Math. Model. Anal.* **18** (2013), no. 1, 1–21.

[9] W. Han, Existence, uniqueness and smoothness results for second-kind Volterra equations with weakly singular kernels, *J. Integral Equations Appl.* **6** (1994), no. 3, 365–384.

[10] H. Kaneko and Y. Xu, Gauss-type quadratures for weakly singular integrals and their application to Fredholm integral equations of the second kind, *Math. Comput.* **62** (1994), no. 206, 739–753.

[11] P. Köhler, Order-preserving mesh spacing for compound quadrature formulas and functions with endpoint singularities, *SIAM J. Numer. Anal.* **32** (1995), no. 2, 671–686.

[12] P. Lima and T. Diogo, An extrapolation method for a Volterra integral equation with weakly singular kernel, *Appl. Numer. Math.* **24** (1997), no. 2, 131–148.

[13] P. Lima and T. Diogo, Numerical solution of a nonuniquely solvable Volterra integral equation using extrapolation methods, *J. Comput. Appl. Math.* **140** (2002), no. 1, 537–557.
[14] H. Majidian, Composite quadrature rules for a class of weakly singular Volterra integral equations with noncompact kernels, *Appl. Numer. Math.* **83** (2014) 1–11.
[15] H. Majidian, Modified Euler's method with a graded mesh for a class of Volterra integral equations with weakly singular kernel, *Numer. Algor.* **67** (2014), no. 2, 405–422.
[16] G. Vainikko, Cordial Volterra integral equations 1, *Numer. Funct. Anal. Optim.* **30** (2009), no. 9-10, 1145–1172.
[17] G. Vainikko, Cordial Volterra integral equations 2, *Numer. Funct. Anal. Optim.* **31** (2010), no. 2, 191–219.
[18] G. Vainikko, Spline collocation for cordial Volterra integral equations, *Numer. Funct. Anal. Optim.* **31** (2010), no. 3, 313–338.
[19] G. Vainikko, Spline collocation-interpolation method for linear and nonlinear cordial Volterra integral equations, *Numer. Funct. Anal. Optim.* **32** (2011), no. 1, 83–109.

(Hassan Majidian) Department of Basic Sciences, Iranian Institute for Encyclopedia Research, P.O. Box 14655-478, Tehran, Iran.
E-mail address: majidian@iecf.ir