Comparative study of numerical methods for a nonlinear weakly singular Volterra integral equation

M. Teresa Diogo, Pedro M. Lima
Departamento de Matemática
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
tdiogo@math.ist.utl.pt, plima@math.ist.utl.pt

Magda S. Rebelo
Departamento de Matemática
Faculdade de Ciências e Tecnologia
Monte da Caparica, 2829-516 Caparica, Portugal
msjr@fct.unl.pt

Abstract

This work is concerned with the numerical solution of a nonlinear weakly singular Volterra integral equation. We investigate the application of product integration methods and a detailed analysis of the Trapezoidal method is given. In order to improve the numerical results we consider extrapolation procedures and collocation methods based on graded meshes. Several examples are presented illustrating the performance of the methods.

AMS subject classification: 65R20
Keywords: nonlinear integral equation; Abel type kernel; trapezoidal method; extrapolation procedure.

1 Introduction

This work is concerned with the following nonlinear Volterra integral equation

\[ y(t) = 1 - \frac{\sqrt{3}}{\pi} \int_0^t s^{\frac{1}{3}} y(s)^4 \frac{ds}{(t-s)^{\frac{2}{3}}} , \quad t \in [0,1], \]  

(1.1)

which arises in connection with a temperature distribution problem studied by Lighthill in [21] (see also [15], [14]). Equation (1.1) has been first studied in [10] and we note that it has an Abel type kernel of the form \( p(t, s, y(s))(t-s)^{-\alpha} \), with \( \alpha = 2/3 \) and \( p(t, s, y) = s^{1/3} y^4 \). Unlike most Abel equations so far studied, the function \( p \) is not differentiable with respect to \( s \) (at \( s = 0 \)); consequently, the classical theory regarding regularity properties and
convergence analysis of numerical methods is not applicable. Therefore the present work complements the existing bibliography on Abel type equations.

Regularity properties and numerical methods for Abel equations with a sufficiently smooth \( p(t, s, y) \) have been considered for example in ([28], [25]). As it is known the solution of these equations is typically nonsmooth and if one wants a high order method, this singular behaviour has to be taken into account in some way. Numerical methods that have been employed include: product integration ([24], [18], [9]), collocation with graded meshes ([5], [8]), transformation of variables ([29], [30], [11], [16], [32]); see also different approaches in ([1], [2], [20], [31], [19], [3]). Related results for Volterra integro-differential equations were obtained in [33], [34]. For comprehensive studies and bibliographic surveys we refer to [17], [7], [6]. A different kind of weakly singular kernel has been considered in [22], [23], where extrapolation methods were used in conjunction with Euler’s method in order to improve the accuracy of the results.

We complete this section with some regularity results for equation (1.1). It is straightforward to demonstrate that (1.1) has a unique continuous solution \( y(t) \) for \( t \in [0, 1] \) (see e.g. [27]). In [10] a series representation for \( y \) was obtained:

\[
y(t) = 1 - 1.461t^{2/3} + 7.252t^{4/3} - 46.460t^2 + 332.9t^{8/3} + \ldots
\]

(1.2)

for values of \( t \) satisfying \( 0 \leq t < R^{3/2} \), where \( R \approx 0.106 \). We thus see that the derivative \( y'(t) \) behaves like \( t^{-1/3} \) near the origin. Moreover, using a result from [8], the behaviour of \( y(t) \) away from the origin has been analysed in [10].

Lemma 1.1 The solution of equation (1.1) is such that \( y \in C^{1,2/3}(\epsilon, 1) \) and \( y \in C^{2,5/3}(\epsilon, 1) \), where \( 0 < \epsilon < R^{3/2} \). That is, \( y(t) \in C^2[\epsilon, 1] \) and for \( t \in (\epsilon, 1] \) we have

\[
|y'(t)| \leq B_y(t - \epsilon)^{-2/3}
\]

(1.3)

\[
|y''(t)| \leq C_y(t - \epsilon)^{-5/3},
\]

(1.4)

for some constants \( B_y > 0 \) and \( C_y > 0 \).

In this work we investigate the application of several numerical methods to equation (1.1). The singular behaviour of the solution near the origin is expected to cause a drop in the global convergence orders. Recently, in [10], we have proved that the explicit product Euler’s method is convergent of order \( 1/3 \) and, for \( t \) away from the origin, obtained order one. In the present paper we give a detailed convergence analysis of the Trapezoidal method and present numerical results illustrating the performance of several methods. A product integration method based on the repeated Simpson’s method is considered as well as spline collocation methods based on uniform and graded meshes. Finally, in order to accelerate the convergence of the numerical methods the use of extrapolation algorithms is also investigated.
2 Two numerical methods

Here we describe the explicit product Euler’s method and the product Trapezoidal method. We introduce on $I = [0, 1]$ the uniform grid $X_h = \{t_i = ih, \ 0 \leq i \leq N\}$, with stepsize $h = 1/N$.

In the Euler’s method we approximate $s^{1/3}y(s)^4$ by a piecewise constant function, that is, for $j = 0, 1, ..., N - 1$,

$$s^{1/3}y(s)^4 \approx t_j^{1/3}y(t_j)^4, \ s \in [t_j, t_{j+1}].$$

This yields the algorithm

$$y_i = 1 - \sqrt{3} \pi \frac{1}{h^2} \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} \frac{ds}{(t_i - s)^{2/3}} t_j^{1/3} y_j^4, \ i = 1, 2, ..., N, \quad (2.1)$$

where $y_i$ denotes an approximation to $y(t_i)$.

In the Trapezoidal method we consider a piecewise linear approximation, that is, on each subinterval $[t_j, t_{j+1}]$,

$$s^{1/3}y^4(s) \approx \frac{(s - t_j)}{h} t_{j+1}^{1/3} y^4(t_{j+1}) + \frac{(t_{j+1} - s)}{h} t_j^{1/3} y^4(t_j). \quad (2.2)$$

We obtain the algorithm

$$y_i = 1 - \frac{\sqrt{3}}{\pi} h \sum_{j=0}^{i} w_{ij} t_j^{1/3} y_j^4, \quad (2.3)$$

with $1 \leq i \leq N$, and where the weights $w_{ij}$ are given by

$$w_{0j} = \frac{1}{h^2} \int_0^{t_1} \frac{(t_1 - s)}{(t_i - s)^{2/3}} ds$$

$$w_{ij} = \frac{1}{h^2} \left( \int_{t_j}^{t_{j+1}} \frac{(t_{j+1} - s)}{(t_i - s)^{2/3}} ds + \int_{t_{j-1}}^{t_j} \frac{(s - t_{j-1})}{(t_i - s)^{2/3}} ds \right)$$

$$1 \leq j \leq i - 1, \ 2 \leq i \leq N;$$

$$w_{ii} = \frac{1}{h^2} \int_{t_{i-1}}^{t_i} \frac{(s - t_{i-1})}{(t_i - s)^{2/3}} ds, \quad 2 \leq i \leq N.$$

It can be shown that there exists $M > 0$, independent of $h$, such that, for $1 \leq i \leq N$,

$$0 < w_{ii} < M h^{-2/3}, \ 0 < w_{ij} < M \frac{h^{-2/3}}{(i - j)^{2/3}} \quad (j < i). \quad (2.4)$$

Taking $y_0 = y(0) = 1$ as a starting value, the above algorithms yield approximate values of $y(t_i)$. In the numerical examples, the non-linear equation (2.3) was solved for $y_i$ by Newton iteration, using $y_{i-1}$ as the initial guess.
3 Convergence results

In [10] we have proved the following theorem.

**Theorem 3.1** Let \( y(t) \) be the solution of (1.1) and \( y_i \) an approximation to \( y(t_i) \) at \( t = t_i \), obtained with the product Euler’s method (2.1). Then the error \( e_i = y(t_i) - y_i \) satisfies

\[
|e_i| \leq D h^{1/3} \\
|e_i| \leq D_1 \left( \frac{h^{4/3}}{t_i^{2/3}} + h \right), \quad i = 1, 2, \ldots, N,
\]

where \( D, D_1 \) are positive constants independent of \( h \).

Therefore we can conclude that the error of the explicit product Euler’s method for equation (1.1) is of order \( O(h^{1/3}) \). However, at points \( t_i \) away from the origin, first order of convergence may be obtained.

In this work we study the product Trapezoidal method and give a summary of its convergence analysis. The total error \( e_i = y(t_i) - y_i \) of the approximate solution obtained with the Trapezoidal method (2.3) satisfies, at \( t = t_i \)

\[
e_i = \frac{\sqrt{3}}{\pi} \frac{1}{i} \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} \frac{s^{1/3}y(s)^4}{(t_i - s)^{2/3}} ds - h \sum_{j=0}^{i} w_{ij} t_j^{1/3} y_j^4
\]

\[
= \frac{\sqrt{3}}{\pi} h \left( \sum_{j=0}^{i} \left( t_j^{1/3} y(t_j)^4 - t_j^{1/3} y_j^4 \right) w_{ij} \right) + T_i,
\]

where \( T_i \) is the quadrature error at \( t = t_i \), given by:

\[
T_i = \frac{\sqrt{3}}{\pi} \frac{1}{i} \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} \left( s^{1/3}y(s)^4 - P_j(s) \right) \frac{1}{(t_i - s)^{2/3}} ds,
\]

\[
i = 1, \ldots, N,
\]

where, for \( s \in [t_j, t_{j+1}] \), \( P_j(s) \) is the interpolating polynomial (cf. (2.2))

\[
P_j(s) = \frac{(t_{j+1} - s)}{h} t_j^{1/3} y(t_j) + \frac{(s - t_j)}{h} t_{j+1}^{1/3} y(t_{j+1}).
\]

Using (2.4) in (3.3) and the fact that the function \( f(y) = y^4 \) satisfies a Lipschitz condition, we obtain

\[
|e_i| \leq |T_i| + M'h^{1/3} \sum_{j=0}^{i} \frac{|e_j|}{(i - j)^{2/3}}.
\]
Therefore, provided \( M' h^{1/3} < 1 \), we arrive at the following Gronwall type inequality
\[
|e_i| \leq C_1|T_i| + M'' h^{1/3} \sum_{j=0}^{i-1} \frac{|e_j|}{(i-j)^{2/3}}. \tag{3.5}
\]

**Lemma 3.1** The quadrature error, \( T_i \), defined by (3.3), satisfies
\[
|T_i| \leq C_3 h^{1/3}, \quad i = 1, 2, \ldots, N, \tag{3.6}
\]
where \( C_3 \) is a positive constant independent of \( h \).

**Proof:** In [10] it was proved that the solution \( y \) to equation (1.1) satisfies the inequality
\[
|y(z) - y(z')| \leq C|z - z'|^{1/3}, \quad \forall z, z' \in [0, 1], \tag{3.7}
\]
where \( C \) is a positive constant that does not depend on \( z \) or \( z' \). Defining
\[
\psi_j(s) = |\left(\frac{t_{j+1} - s}{h}\right)^{1/3} y^4(s) - y^4(t_j)| + |\left(\frac{s - t_j}{h}\right)^{1/3} y^4(s) - y^4(t_{j+1})|,
\]
and
\[
\phi_j(s) = y^4(s) \left| s^{1/3} - \left(\frac{t_{j+1} - s}{h}\right)^{1/3} t_j^{1/3} + \frac{s - t_j}{h} t_{j+1}^{1/3} \right|,
\]
we have
\[
|s^{1/3} y^4(s) - P_j(s)| \leq \psi_j(s) + \phi_j(s), \quad s \in [t_j, t_{j+1}], \tag{3.8}
\]
for \( j = 0, 1, \ldots, i - 1 \). Then, making use of (3.7), gives
\[
\psi_j(s) \leq |y^4(s) - y^4(t_j)| + |y^4(s) - y^4(t_{j+1})| \leq LC(s-t_j)^{1/3} + |s - t_j|^{1/3} = 2LCh^{1/3},
\]
where \( L \) is the Lipschitz constant of the function \( f(y) = y^4 \). On the other hand, it can be shown that, for \( s \in (t_j, t_{j+1}], \ j = 0, 1, \ldots, N - 1, \)
\[
|s^{1/3} - \left(\frac{t_{j+1} - s}{h}\right)^{1/3} t_j^{1/3} + \frac{s - t_j}{h} t_{j+1}^{1/3}| \leq C_2 h^{1/3}.
\]
Let \( M_4 = \max_{s \in [0,1]} |y^4(s)| \). Using the above inequalities to bound (3.8), it follows from (3.3)
\[
|T_i| \leq \max\{2LC + M_4C_2\} h^{1/3} \int_0^{t_i} \frac{1}{(t_i - s)^{2/3}} ds \leq C_4 h^{1/3}. \quad \Diamond
\]

Using Lemma 3.1 in (3.5) and applying a standard weakly singular Gronwall lemma (see e.g. [13]) leads to the following theorem.
Theorem 3.2 Let $y(t)$ be the solution of (1.1) and $y_i$ an approximation to $y(t)$ at $t = t_i$ obtained with the Trapezoidal method (2.3). Then, the error $e_i = y(t_i) - y_i$ satisfies:

$$|e_i| \leq C_4 h^{1/3}, \quad i = 1, \ldots, N,$$

where $C_4$ is a constant independent of $h$.

By a detailed analysis of the quadrature error, as it was done in [10] for the product Euler’s method, we have the following result.

Lemma 3.2 The quadrature error, $T_i$, defined by (3.3), satisfies

$$|T_i| \leq C_5 \frac{h^{4/3}}{t_i^{2/3}} + C_6 h^2, \quad i = 1, 2, \ldots, N,$$

where $C_5$, $C_6$ are positive constants independent of $h$.

Using (3.10) in (3.5) we obtain

$$|e_i| \leq C_7 \frac{h^{4/3}}{t_i^{2/3}} + C_8 h^2 + M'' h^{1/3} \sum_{j=0}^{i-1} \frac{|e_j|}{(i-j)^{2/3}}.$$

Then a generalized discrete Gronwall lemma from [12] can be applied to yield the following convergence result.

Theorem 3.3 Let $y(t)$ be the solution of (1.1) and $y_i$ an approximation to $y(t)$ at $t = t_i$ obtained with the product Trapezoidal method (2.3). Then the error $e_i = y(t_i) - y_i$ satisfies

$$|e_i| = |y(t_i) - y_i| \leq C_9 \left( \frac{h^{4/3}}{t_i^{2/3}} + h^2 \right),$$

where $C_9$ is a positive constant independent of $h$.

From the above theorem we conclude that the order of the error at the fixed point $t_i$ away from the origin is $4/3$. 

6
4 Numerical results

In this section we present some numerical results obtained with the product Euler and Trapezoidal methods, while in Section 5 several other methods are considered.

As an estimate of the convergence rates, the following quantity has been used in all the experiments:

$$p = \frac{\log \left( \frac{y^{h/2} - y^h}{y^{h/4} - y^{h/2}} \right)}{\log 2},$$  \hspace{1cm} (4.1)

where $y^h$, $y^{h/2}$ and $y^{h/4}$ denote approximations to $y(t)$ using the step sizes $h$, $h/2$ and $h/4$, respectively. In Tables 1,3,5,6 two sets of values of $h = 1/N$ have been considered, with $N = 80, 160, 320$ and $N = 160, 320, 640$, respectively. In order to obtain estimates for the errors $|e^h(t_i)| = |y(t_i) - y^h_i|$, we have used the formula

$$|e^h(t_i)| \approx \left| y^{h/2}_i - y^h_i \right| \left( 1 - \frac{1}{2} \right)^p.$$  \hspace{1cm} (4.2)

The results of Table 1 are in agreement with Theorems 3.1 and 3.2, confirming the predicted first order of convergence for the Euler’s method and 4/3 for the Trapezoidal method. The absolute errors corresponding to the values of $h = 1/160, 1/320, 1/640$ are shown in Figures 1,2.

For the error estimates, we have used (4.2) with $p = 1$ for Euler’s method and $p = 1.3$ for the Trapezoidal method, according to the estimate (4.1). The computed error norms, given by:

$$\|e^h\|_{\infty} = \max_{1 \leq i \leq N} |y(t_i) - y^h_i|$$

are displayed in Table 2. The results show an expected drop in the global convergence orders, due to the nonsmooth behaviour of the solution near the origin (cf. Theorems 3.1 and 3.2).
5 Other methods

In this section we consider a product integration method based on the repeated Simpson’s rule as well as collocation methods using polynomial splines of degrees 0, 1, 2. As an attempt to recover the optimal convergence orders, the use of graded meshes is also tested. Finally, extrapolation algorithms are employed in conjunction with the Euler and Trapezoidal methods, in order to improve the accuracy of the results.

5.1 Higher order product quadrature rules

Consider the discretised form of (1.1) at $t = t_i$

$$y(t_i) = 1 - \frac{\sqrt{3}}{\pi} \int_0^{t_i} \frac{s^4 y(s)^4}{(t_i - s)^2} ds, \quad 0 \leq i \leq N. \quad (5.1)$$
Table 2: Error norms for several values of $h$

<table>
<thead>
<tr>
<th>$h$</th>
<th>Euler's method</th>
<th>Trapezoidal method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{40}$</td>
<td>0.1782</td>
<td>0.1600 $\times 10^{-1}$</td>
</tr>
<tr>
<td>$\frac{1}{80}$</td>
<td>0.1122</td>
<td>0.1117 $\times 10^{-1}$</td>
</tr>
<tr>
<td>$\frac{1}{160}$</td>
<td>0.7071 $\times 10^{-1}$</td>
<td>0.7548 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$\frac{1}{320}$</td>
<td>0.4454 $\times 10^{-1}$</td>
<td>0.4981 $\times 10^{-2}$</td>
</tr>
<tr>
<td>$\frac{1}{640}$</td>
<td>0.2806 $\times 10^{-1}$</td>
<td>0.3234 $\times 10^{-2}$</td>
</tr>
</tbody>
</table>

In the product Simpson’s method we approximate the integral in (5.1) by the product Simpson’s rule used repeatedly over $[0, t_i]$ if $i$ is even. When $i$ is odd, we use the product Simpson’s rule over $[0, t_i-3]$ and the product three-eights rule over $[t_i-3, t_i]$. Then the product Simpson’s method for equation (1.1) is defined by

$$y_{2r} = 1 - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{j=0}^{r-1} \sum_{k=0}^{2} t_{2j+k}^{1/3} y_{2j+k}^4 b_k(2j, 2r)$$

$$y_{2r+1} = 1 - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{j=0}^{r-2} \sum_{k=0}^{2} t_{2j+k}^{1/3} y_{2j+k}^4 b_k(2j, 2r + 1) - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{k=0}^{3} t_{2r-2+k}^{1/3} y_{2r-2+k}^4 d_k,$$

with

$$b_0(j, i) = \frac{1}{2} \int_0^2 \frac{(v - 2)(v - 1)}{(i - j - v)^{2/3}} dv$$

$$b_1(j, i) = -\int_0^2 \frac{v(v - 2)}{(i - j - v)^{2/3}} dv$$

$$b_2(j, i) = \frac{1}{2} \int_0^2 \frac{v(v - 1)}{(i - j - v)^{2/3}} dv$$

and

$$d_0 = -\frac{1}{6} \int_0^3 \frac{(v - 2)(v - 1)(v - 3)}{(3 - v)^{2/3}} dv$$
\[
d_1 = \frac{1}{2} \int_0^3 \frac{v(v-2)(v-3)}{(3-v)^{2/3}} dv \\
d_2 = -\frac{1}{2} \int_0^3 \frac{v(v-1)(v-3)}{(3-v)^{2/3}} dv \\
d_3 = \frac{1}{6} \int_0^3 \frac{v(v-1)(v-2)}{(3-v)^{2/3}} dv.
\]

Like in the Trapezoidal method, the nonlinear equations (5.2) and (5.3) were solved by Newton’s method and we have used the series (1.2) to compute the starting value \( y_1 \approx y(t_1) \).

Again the numerical results show evidence of a drop in the global convergence order (cf. Table 4). The results of Table 3 suggest that, for points away from the origin, the product Simpson’s method exhibits the same convergence order, 4/3, as the Trapezoidal method. We believe that similar conclusions are to be expected for product integration methods based on the application of higher order repeated rules (as main rules).

<table>
<thead>
<tr>
<th>( t_i )</th>
<th>( 1 )</th>
<th>( 1/160 )</th>
<th>( 1/320 )</th>
<th>( 1 )</th>
<th>( 1/160 )</th>
<th>( 1/320 )</th>
<th>( 1 )</th>
<th>( 1/640 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>1.379</td>
<td>1.385</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>1.388</td>
<td>1.387</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1.389</td>
<td>1.387</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>1.390</td>
<td>1.388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1.390</td>
<td>1.388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.391</td>
<td>1.388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**5.2 Collocation methods with uniform meshes**

Here we present numerical results generated by the application of collocation methods with uniform meshes. The following notation and methods were introduced in [7]. Let \( \Pi_N : 0 = t_0 < t_1 < \ldots < t_N = 1 \) be a partition of the interval \([0,1]\). Consider the subintervals \( \sigma_0 := [t_0, t_1] \), \( \sigma_n := (t_n, t_{n+1}] \), \( 1 \leq n \leq N - 1 \), and define \( Z_N := \{ t_n : n = 1, \ldots, N - 1 \} \). The collocation methods use elements of the polynomial spline space \( S_{m-1}^{(d)}(Z_N) \), that is, functions \( u \in C^{(d)}([0,1]) : u|\sigma_n = u_n \in \pi_{m-1} \), \( 1 \leq n \leq N - 1 \). Here \( \pi_{m-1} \) is the set of polynomials of degree not exceeding \( m - 1 \) (with \( m \geq 1 \)). In the case \( d = -1 \), no continuity conditions are imposed at the mesh points and \( u \) will in general possess jump discontinuities at the knots \( Z_N \). The desired approximation to the solution of equation
Table 4: Error norms for Simpson’s method

| \( h \) | \( ||e^h||_\infty \) |
|-------|----------------|
| \( \frac{1}{40} \) | \( 0.4953 \times 10^{-2} \) |
| \( \frac{1}{80} \) | \( 0.3500 \times 10^{-2} \) |
| \( \frac{1}{160} \) | \( 0.2377 \times 10^{-2} \) |
| \( \frac{1}{320} \) | \( 0.1578 \times 10^{-2} \) |
| \( \frac{1}{640} \) | \( 0.1030 \times 10^{-2} \) |

(1.1) is an element \( u \in S^{(d)}_{m-1}(Z_N) \) satisfying

\[
   u(t) = 1 - \frac{\sqrt{3}}{\pi} \int_0^t \frac{s^{1/3}u^4(s)}{(t-s)^{2/3}} ds, \quad t \in X(N),
\]

where \( X_N = \bigcup_{n=0}^{N-1} X_n \) with

\[
   X_n = \left\{ t_{nj} = t_n + c_j h : 0 \leq c_1 < \ldots < c_m \leq 1, \quad h = \frac{1}{N} \right\}.
\]

The collocation equation (5.4) has the form

\[
   u_i(t_{ik}) = 1 - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{j=0}^{i-1} \int_0^1 \frac{(t_j + sh)^{1/3}u_j^4(t_j + sh)}{(i + c_k - j - s)^{2/3}} ds
   - \frac{\sqrt{3}}{\pi} h^{1/3} \int_0^{c_k} (t_i + sh)^{1/3}u_i^4(t_i + sh) \frac{(c_k - s)^{2/3}}{(c_k - s)^{2/3}} ds.
\]

Approximating the integrals in (5.5) by product integration formulae, we obtain the following discretised version of (5.4)

\[
   u_{ik} = 1 - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{j=0}^{i-1} \sum_{l=1}^{m} w_{kl}^{(ij)} (t_j + c_l h)^{1/3} u_{jl}^4
   - \frac{\sqrt{3}}{\pi} h^{1/3} \sum_{l=1}^{m} w_{kl} (t_i + c_l c_k h)^{1/3} \sum_{r=1}^{m} L_r(c_k c_l) u_{lr}^4.
\]

(5.6)
for $k = 1, \ldots, m$ and $0 \leq i \leq N - 1$. Above the $L_r$, $r = 1, \ldots, m$, are the Lagrange polynomials associated with $c_r$. The quadrature weights in (5.6) are given by

$$w_{kl}^{(ij)} = \int_0^1 \lambda_l(s) \frac{(i + c_k - j - s)^{2/3}}{(c_k - s)^{2/3}} ds$$

$$w_{kl} = \int_0^{c_k} \lambda_{kl}(s) \frac{(c_k - s)^{2/3}}{(c_k - s)^{2/3}} ds,$$

where

$$\lambda_l(s) = \prod_{j=1, j \neq l}^m \frac{(s - c_j)}{(c_l - c_j)}$$

and

$$\lambda_{kl}(s) = \prod_{j=1, j \neq l}^m \frac{(s - c_k c_j)}{(c_k c_l - c_j c_k)}.$$

In order to approximate the solution of equation (1.1) we have considered several choices of $m$ and of the collocation parameters:

1. $m = 2$ ($u \in S_1^{-1}(Z_N)$), with collocation parameters
   
   - $c_1 = 1/2$ and $c_2 = 1$,
   - $c_1 = 2/3$ and $c_2 = 1$,

2. $m = 3$ ($u \in S_2^{-1}(Z_N)$), with collocation parameters
   
   - Gauss points:
     
     $c_1 = \frac{3-\sqrt{3}}{6}, c_2 = \frac{3+\sqrt{3}}{6}, c_3 = 1$;
   
   - Radau II points:
     
     $c_1 = \frac{4-\sqrt{6}}{10}, c_2 = \frac{4+\sqrt{6}}{10}, c_3 = 1$;

In Tables 5 and 6 we show the computed experimental rates of convergence, obtained with (4.1), using different step sizes $h = 1/N$. The results suggest that the convergence order of the collocation methods is approximately 1.3 and, similarly to product integration methods, we see a reduction in the global orders of convergence (cf. Table 7). Let us compare, for example, the error norms of Table 7 (1st column) with Table 2 (2nd column) and the ones in Table 7 (2nd column) with Table 4. It would appear that collocation methods give more accurate approximations than the ones obtained with product integration methods using quadrature rules of the same orders.
5.3 Collocation methods with graded meshes

One way to recover the optimal convergence rates of collocation methods for weakly singular equations, when the solution is not smooth, is to use a graded mesh ([5], [8]). We have considered collocation in the polynomial spline space $S_{1}^{(-1)}(Z_{N})$, using the grid:

$$\Delta N = \left\{ t_{i} \in [0, 1] : t_{i} = (i/N)^{3}, \; i = 0, 1, ..., N \right\}.$$  

The numerical results of Table 8 indicate first order of convergence for the piecewise constant approximation, that is, the optimal order seems to have been recovered.
Table 7: Error norms for several values of $h$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$S_1^{(-1)}(Z_N)$</th>
<th>$S_2^{(-1)}(Z_N)$</th>
<th>Gauss points</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{40}$</td>
<td>$0.1177 \times 10^{-2}$</td>
<td>$0.2947 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{1}{80}$</td>
<td>$0.8296 \times 10^{-3}$</td>
<td>$0.2013 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{1}{160}$</td>
<td>$0.5648 \times 10^{-3}$</td>
<td>$0.1342 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{1}{320}$</td>
<td>$0.3745 \times 10^{-3}$</td>
<td>$0.8787 \times 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{1}{640}$</td>
<td>$0.2437 \times 10^{-3}$</td>
<td>$0.5684 \times 10^{-4}$</td>
<td></td>
</tr>
</tbody>
</table>

5.4 Extrapolation

An extensive survey on extrapolation methods is given in [4]; see also [26] for applications to various types of functional equations.

Here we present some results concerning the use of extrapolation procedures in order to accelerate the convergence of the numerical results obtained with the two methods considered in Section 2. In the case of the explicit product Euler’s method, we have assumed that the approximate solution $y^{h_n}$ has an asymptotic error expansion with the form

$$y_{j}^{h_n} - y(t_j) = a_1 h_n + a_2 h_n^2 + O(h^3),$$

for $t = t_j$ away from the origin. Using an algorithm based on Richardson’s extrapolation, we started with an initial approximation $E_0^{(n)} = y_{j}^{h_n}$ and the new approximations were computed recursively by

$$E_{kj}^{(n)} = \frac{h_{n+k} E_{k-1j}^{(n)} - h_n E_{k-1j}^{(n+1)}}{h_{n+k} - h_n}, \quad k = 1, 2; \quad n = 0, 1, 2, 3$$

The results of Tables 9 and 10 illustrate this extrapolation process at $t = 0.3$ and $t = 0.5$, where we have taken $E_2^{(1)}$ as the exact solution of equation (1.1) at those points. We see that the convergence is accelerated only in the first step of the process, that is, there is an improvement in the accuracy from the first to the second column but not to the third column. This seems to confirm the $O(h)$ order of the first term of the error expansion, but no conclusions can be drawn about the order of the next term.
Table 8: Error norms for collocation with graded meshes

<table>
<thead>
<tr>
<th>( h )</th>
<th>( |e|_\infty )</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{10} )</td>
<td>( 0.2135 \times 10^{-1} )</td>
<td>–</td>
</tr>
<tr>
<td>( \frac{1}{20} )</td>
<td>( 0.7969 \times 10^{-2} )</td>
<td>1.422</td>
</tr>
<tr>
<td>( \frac{1}{40} )</td>
<td>( 0.3352 \times 10^{-2} )</td>
<td>1.250</td>
</tr>
<tr>
<td>( \frac{1}{80} )</td>
<td>( 0.1502 \times 10^{-2} )</td>
<td>1.158</td>
</tr>
<tr>
<td>( \frac{1}{160} )</td>
<td>( 0.6935 \times 10^{-3} )</td>
<td>1.115</td>
</tr>
<tr>
<td>( \frac{1}{320} )</td>
<td>( 0.3269 \times 10^{-3} )</td>
<td>1.085</td>
</tr>
<tr>
<td>( \frac{1}{640} )</td>
<td>( 0.1563 \times 10^{-3} )</td>
<td>1.065</td>
</tr>
</tbody>
</table>

Table 9: Absolute error of the entries of the E-array (\( t = 0.3 \)) for Euler’s method

| \( n \) | \( |E_0^{(n)} - y(0.3)|\) | \( |E_1^{(n)} - y(0.3)|\) | \( |E_2^{(n)} - y(0.3)|\) |
|---------|-----------------|-----------------|-----------------|
| 0       | \( 0.2697 \times 10^{-3} \) | \( 0.3337 \times 10^{-3} \) | \( 0.1427 \times 10^{-3} \) |
| 1       | \( 0.1032 \times 10^{-3} \) | \( 0.1904 \times 10^{-5} \) | \( 0.1904 \times 10^{-3} \) |
| 2       | \( 0.5064 \times 10^{-4} \) | \( 0.4761 \times 10^{-6} \) | \( 0.4761 \times 10^{-4} \) |
| 3       | \( 0.2508 \times 10^{-4} \) | \( 0.2508 \times 10^{-4} \) | \( 0.2508 \times 10^{-4} \) |

For the product Trapezoidal method we have assumed that the approximate solution \( y^{h_n} \) has an asymptotic error expansion with the form

\[
y_j^{h_n} - y(t_j) = a_1 h_n^{4/3} + a_2 h_n^2 + O(h^3),
\]

for points \( t = t_j \) away from the origin. In this case we have used the E-algorithm (see e.g. [4]) in order to accelerate the convergence of the Trapezoidal method. The computation of \( E_k^{(n)} \) begins with

\[
E_0^{(n)} = y_j^{h_n}, \quad n = 0, 1, 2
\]

\[
g_0^{(n)} = g_1(n), \quad i = 1, 2; n = 0, 1, 2
\]

where \( g_1(n) = h_n^{4/3} \) and \( g_2(n) = h_n^2 \).
Table 10: Absolute error of the entries of the E-array \((t = 0.5)\) for Euler’s method

| \(n\) | \(|E_0^{(n)} - y(0.5)|\) | \(|E_1^{(n)} - y(0.5)|\) | \(|E_2^{(n)} - y(0.5)|\) |
|-------|-----------------|-----------------|-----------------|
| 0     | 0.8704 × 10^{-4}| 0.1737 × 10^{-5}| 0.7749 × 10^{-6}|
| 1     | 0.4265 × 10^{-4}| 0.1015 × 10^{-5}|                   |
| 2     | 0.2882 × 10^{-4}| 0.2539 × 10^{-6}|                   |
| 3     | 0.1028 × 10^{-4}|                   |                   |

For \(k = 1, 2, 3\) and \(n = 0, 1, ..., 3 - k\) the new approximations are computed recursively by

\[
E_k^{(n)} = \frac{E_k^{(n)} - E_k^{(n+1)}}{g_k^{(n)} - g_k^{(n+1)}}
\]

\[
g_{k,i}^{(n)} = \frac{g_k^{(n)} g_{k-1}^{(n+1)} - g_k^{(n+1)} g_{k-1}^{(n)}}{g_k^{(n)} - g_k^{(n+1)}}, \quad i = k + 1, k + 2, ...
\]

We have taken \(E_2^{(1)}\) as the exact solution of equation (1.1) and the absolute errors, for \(t = 0.3\) and \(t = 0.5\), of the entries of the E-array are displayed in Tables 11 and 12. The results show that the accuracy is improved in each step of the extrapolation process. This seems to confirm the \(O(h^{4/3})\) order of the first term of the error expansion and the \(O(h^2)\) order of the second term.

Table 11: Absolute error of the entries of the E-array \((t = 0.3)\) for the Trapezoidal method

| \(n\) | \(|E_0^{(n)} - y(0.3)|\) | \(|E_1^{(n)} - y(0.3)|\) | \(|E_2^{(n)} - y(0.3)|\) |
|-------|-----------------|-----------------|-----------------|
| 0     | 0.9875 × 10^{-4}| 0.7041 × 10^{-4}| 0.4993 × 10^{-5}|
| 1     | 0.3961 × 10^{-4}| 0.2135 × 10^{-4}|                   |
| 2     | 0.1585 × 10^{-4}| 0.5337 × 10^{-7}|                   |
| 3     | 0.6322 × 10^{-5}|                   |                   |
Table 12: Absolute error of the entries of the E-array (t = 0.5) for the Trapezoidal method

| n | |E_0^n| - y(0.5)| |E_1^n| - y(0.5)| |E_2^n| - y(0.5)| |
|---|---|---|---|---|
| 0 | 0.5608 × 10^{-4} | 0.4869 × 10^{-6} | 0.2736 × 10^{-7} |
| 1 | 0.2255 × 10^{-4} | 0.1423 × 10^{-6} |
| 2 | 0.9034 × 10^{-5} | 0.3556 × 10^{-7} |
| 3 | 0.3606 × 10^{-5} |

6 Conclusions

This work has been concerned with the numerical analysis of the nonlinear Volterra integral equation (1.1), which has a weakly singular kernel of the form s^{1/3}y(s)^4(t - s)^{-2/3}. The derivative y'(t) of the solution of this equation behaves like t^{-1/3} near the origin which causes the loss of the optimal (global) convergence orders of product integration and collocation methods. This was shown theoretically for the explicit product Euler’s method in [10], and now in Section 2 for the product Trapezoidal method; while their global order is 1/3, for points t away from the origin the convergence order is one for Euler’s method and 4/3 for the Trapezoidal method. These results were confirmed by some numerical examples (cf. Section 4). We have also implemented a product integration method based on Simpson’s rule as well as collocation methods using polynomial splines of degrees 0,1,2. The numerical experiments of Section 5 seem to support the conjecture that general product integration and collocation methods applied to equation (1.1) have 1/3 global order of convergence, independently of the degree of the approximating polynomials used; as t increases the errors seem to be of order 4/3, except in the case of Euler’s method, which has order one.

On the other hand, the use of collocation methods based on graded meshes suggests that the optimal orders can be recovered. Finally, Richardson’s extrapolation procedure was used in conjunction with the product Euler’s method and some improvement in the accuracy was observed. In order to accelerate the convergence of the Trapezoidal method the E-algorithm was applied, indicating that the first and the second terms of the error expansion are of orders O(h^{4/3}) and O(h^2), respectively. Hence in this case the extrapolation procedure has yielded a more significant increase in the accuracy of numerical solutions.

7 Acknowledgements

This research work was supported by Fundação para a Ciência e Tecnologia (Project POCTI/MAT/45700/2002).
References


