A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations

Hermann Brunner

1. INTRODUCTION

In this paper we present a survey of methods for the numerical treatment of various types of Volterra functional equations. The equations subsequently discussed are characterized by compact integral operators (in the space of continuous functions with uniform norm) of the form

\[(L_\alpha f)(t) := \int_0^t (t-s)^{-\alpha} G(t,s)f(s)ds, \quad t \in I := [t_0,T] \text{ (compact), } 0 \leq \alpha < 1 \text{ (linear case),} \quad (1.1)\]

and

\[(V_\alpha f)(t) := \int_0^t (t-s)^{-\alpha} K[t,s,f(s)]ds, \quad t \in I, 0 \leq \alpha < 1 \text{ (nonlinear case),} \quad (1.2)\]

Here, the kernels G and K are assumed to be (at least) continuous on \(S := \{(t,s) : t_0 \leq s \leq t \leq T\}\) and on \(S \times \mathbb{R}\), respectively. Three main classes of functional equations will be considered:

- \(y(t) = g(t) + (V_\alpha y)(t), \quad t \in I \) (Volterra integral equation of the second kind); \( (1.3) \)
- \( (L_\alpha y)(t) = g(t), \quad t \in I \) (linear Volterra integral equation of the first kind); \( (1.4) \)
- \( y'(t) = F[t,y(t)] + (V_\alpha Y)(t), \quad t \in I \) (first-order Volterra integro-differential equation). \( (1.5) \)

An equation of the above type whose kernel contains a weak singularity (i.e. \(0 \leq \alpha < 1\)) is often referred to as an Abel integral or integro-differential equation.

In addition to these types of functional equations an important class of singular (first-kind) integral equations which occur in numerous physical modelling processes and whose underlying integral operator is not compact will be briefly mentioned in section 6.

A brief glimpse at the books [A11], [A10], the surveys [A26], [A27] and [A14], and the list of recent theses [B1] – [B26] clearly reveals that the last fifteen years have seen a rapid development of three aspects of Volterra equations:

(i) the qualitative theory (i.e. the asymptotic behavior of solutions) of equations of the form (1.3) and (1.5);
(ii) the numerical approximation of solutions of (1.3), (1.4) and (1.5); and
(iii) the application of functional equations of the above forms in the modelling of a wide variety of problems in the physical and biological sciences.

The present survey will deal with the second of these developments; references [A14] and especially [A10] reflect the state of the art regarding the first of these three aspects, while [A15] and [A9] have been added as summary-type references for the classical theory of Volterra equations. As an introduction to the application of such equations the reader may wish to consult [A28], [A2], as well as the references listed under subheading H of the bibliography (we should like to point out [H1], [H9], [H10], [H13], [H14], [H26], [H27], [H30], [H35]); many of these references contain extensive bibliographies.

A comprehensive listing of books and articles on methods for the approximate solution of Volterra equations (and on the application of these equations) up to 1971 was given by Noble in [A12]. Our survey thus includes articles and monographs published before 1971 only if the seem, to the author, to be of special historical interest or if they represent contributions which were particularly important for the further development of numerical methods. The bibliography which complements the subsequent discussion classifies the references under various subheadings to facilitate the use of this survey. It has as its principal aim to serve as a guide to further reading (due to limitation of space many of the numerical methods will not be described in detail); it is, therefore, represent-
In order to discretize a given Volterra equation we associate with the compact interval I = [t_0, T] the (uniform) partition characterized by \( t_n := t_0 + nh \), \( n = 0(1)n \) (\( n \geq 1 \)), with \( n = T \) and we denote by \( y_n \) any approximation to the exact solution \( y(t) \) at the point \( t = t_n \). We discuss the discretization of (1.3) with \( \alpha = 0 \) and \( 0 < \alpha < 1 \) separately: since an equation of the form

\[
y(t) = g(t) + (V_0y)(t),
\]

may be regarded as a generalization and \( 0 < \alpha < 1 \) separately: since an equation of the form

\[
y(t) = g(t) + (V_0y)(t),
\]

may be regarded as a generalization of the initial value problem

\[
y'(t) = f[t, y(t)], \quad t \in I, \quad y(t_0) = y_0
\]

(2.1)

(note that its integrated form,

\[
y(t) = y_0 + \int_{t_0}^{t} f[s, y(s)] ds, \quad t \in I,
\]

(2.2)

has a kernel and a nonhomogeneous term not depending on \( t \), many of the numerical methods for such equations are modelled after methods for (2.1). However, most of these methods cannot be applied to (1.3) with \( 0 < \alpha < 1 \) due to the occurrence of the weak singularity in the kernel.

2.1. The case (1.3) with \( \alpha = 0 \)

An important (and perhaps the most natural) class of numerical methods for approximating the solution of (1.3), \( \alpha = 0 \), is obtained by discretizing the integral term, \( (V_0y)(t) \), by an appropriate family of quadrature formulas associated with the mesh points \( \{ t_n : n = r(1)N \} \) for some integer \( r > 1 \), i.e., we replace the equation

\[
y'(t_n) = g(t_n)(V_0y)(t_n)
\]

by the triangular algebraic system

\[
y_n = g_n + h \sum_{j=0}^{n} w_{n,j} K(t_n, t_j, y_j)(n = r(1)N),
\]

(2.3)

with given starting values \( \{ y_0 = g(t_0), y_1, \ldots, y_{r-1} \} \).

Thus (2.3) represents a recursive process for generating the approximations \( \{ y_1, \ldots, y_N \} \).

A general discussion of these so-called (direct) quadrature methods may be found in [A4] (pp. 759-849); compare also [C29], [C46] (containing a generalization of the Dahlquist-Ortiz theory for linear multistep methods for (2.1), [C60], [C64] (both dealing with the use of the Gregory quadrature formulas), [A24], [C40], [C56], [C65], and [C35]. The required \( (r-1) \) starting values in (2.3) are computed either by specially designed starting methods ([C25]), also [A4], pp. 835-849), or by Runge-Kutta methods (or more general one-step methods based on piecewise polynomial collocation) whose order of accuracy is consistent with that of the quadrature method; these latter methods are described below.

If the kernel \( K(t, s, y) \) is nonlinear with respect to \( y \), and if \( w_{n,n} \neq 0 \), then (2.3) represents a sequence of (\( N-r+1 \)) nonlinear algebraic equations for the values \( \{ y_r, \ldots, y_N \} \); i.e., we have

\[
y_n = h w_{n,n} K(t_n, t_n, y_n) + \Phi_n \quad (n = r(1)N),
\]

(2.4)

with \( \Phi_n := g(t_n) + h \sum_{j=0}^{n-1} w_{n,j} K(t_n, t_j, y_j) \) not depending on \( y_n \). If the given kernel is such that its Lipschitz constant with respect to \( y \) is not large, these equations are usually solved by predictor-corrector techniques analogous to those used in linear multistep methods for (2.1). For details we refer to [C31]; see also [C23] for a discussion of such methods based on certain rational approximation techniques. Predictor-corrector schemes using variable stepizes are treated in the thesis [B25].

Among the direct quadrature methods for Volterra integral equations of the second kind a subclass related to linear multistep methods for the initial-value problem (2.1) has recently been analyzed in detail, motivated largely by investigations of numerical stability. If a linear \( k \)-step method \((\rho, \sigma)\) defined by the characteristic polynomials \( \rho(z) := \sum_{j=0}^{k} \alpha_j z^j \) and \( \sigma(z) := \sum_{j=0}^{k} \beta_j z^j \) \( (k \geq 1) \)

is applied to (2.1) we obtain the recurrence relation

\[
\sum_{j=0}^{k} \alpha_j y_n + h \sum_{j=0}^{k} \beta_j f(t_n+j, y_{n+j}) = (n = 0(1)N-k; y_0, \ldots, y_{n-1} \text{ given})
\]

(2.5)

which, for all \( n \), depends on the same fixed number of values \( \{ y_n, \ldots, y_{n+k} \} \). This is in sharp contrast to the recurrence relation (2.3) which exhibits a (lower) triangular structure. Consider now the special Volterra equation (2.2) and its corresponding discretization (2.3): it is natural to ask under which conditions on the quadrature weights \( \{ w_{n,j} \} \) this discretization can be reduced (by taking appropriate linear combinations of certain (fixed) number of consecutive relations) to a fixed-term recurrence relation of the form (2.5). A direct qua-
The paper [C67] contains the details of the structure of this relationship and gives a number of specific examples (where methods reducible to backward differentiation methods are of particular interest). In addition, an efficient stable procedure for generating these weights \( \{w_{n,j}\} \) is derived. (Compare also [C43] for a detailed analysis of backward differentiation methods for Volterra equations.)

A disadvantage of nontrivial \((\rho, \sigma)\)-reducible quadrature methods is that their implementation (i.e. the generation of the quadrature weights \( \{w_{n,j}\} \) for each integration step \( n \)) or, alternately, the use of imbedding techniques (compare [C69]) results in considerable overhead costs. Recently Wolkenfelt [C68] has proposed so-called multilag methods (based on a linear multistep method \((\rho, \sigma)\) and a family of efficient quadrature formulas) which permit a more efficient implementation; we shall return to these methods briefly when dealing with numerical stability.

Before turning to one-step methods (particularly Runge-Kutta methods) for Volterra integral equations of the second kind we add this remark. It follows from the analysis of backward differentiation methods for Volterra equations.

The general Runge-Kutta method for (2.7) (in the following called VRK method) is then given by

\[
y_{n+1} = y_{n} + h \sum_{i=1}^{m} b_{i} K_{n} + c_{i} h, Y_{n}^{(i)}.
\]

Thus for each value of \( n \) the number of kernel evaluations is \( m(m+1)/2 \) (for the implicit case, \( m(m+1) \)), while \( K_{n}(\cdot) \) depends on the vector \( \theta_{i}(\cdot) \) and an appropriate set of quadrature formulas for the intervals \([t_{n}, t_{n+1}]\) whose selection will be governed by the order of the Runge-Kutta part of the method and certain efficiency criteria like minimization of the number of kernel evaluations. The scheme (2.9a) is called explicit if \( m(i) = i-1 \); for \( m(i) = i \) and \( m(i) = m \) \((i = 1, 2, \ldots, m)\) we have methods which are semi-explicit and implicit, respectively.

The general formulation (2.9) contains two important special cases (compare the original papers [C62] and [C9]):

A. Runge-Kutta methods of Pouzet type (PRK methods): These methods are characterized by

\[
y_{n+1} = y_{n} + h \sum_{i=1}^{m} b_{i} K_{n} + c_{i} h, Y_{n}^{(i)}.
\]

Thus for each value of \( n \) the number of kernel evaluations in the Runge-Kutta part is \( m(m+1)/2 \) for the explicit case, and \( m(m+1) \) for the implicit case. (Note that, for the implicit case, kernel values have to be supplied for \( s > t \).)
B. Runge-Kutta methods of Bel’tyukov type (BRK methods):

Here we have dij = ej (< cj, in order to avoid defining K for s > t), 0 i= c i :

\[ Y_n^{(i)} = \hat{F}_{n}^{(i)}(t_n + c_i h) + h \sum_{j=1}^{m(i)} b_j K(t_n + c_j h, t_n + c_j h, Y_n^{(j)}) \quad (i = 1(1)m), \]

\[ y_{n+1} = \hat{F}_{n+1}^{(1)}(t_{n+1} + h) + h \sum_{j=1}^{m} b_j K(t_{n+1} + c_j h, t_{n+1} + c_j h, Y_n^{(j)}), \]  

\[ (2.11a) \]

\[ (2.11b) \]

For these methods the number of kernel evaluations in the Runge-Kutta part is given by m, independent of whether they are explicit or implicit.

Since a Runge-Kutta method for the initial-value problem (2.1) has the form

\[ k_{n}^{(i)} = Y_n + h \sum_{j=1}^{m(i)} b_j f(t_n + c_j h, k_{n}^{(j)}) \quad (i = 1(1)m), \]

\[ y_{n+1} = \hat{F}_{n+1}^{(1)}(t_{n+1}) + h \sum_{j=1}^{m} b_j f(t_{n+1} + c_j h, k_{n}^{(j)}), \]  

\[ (2.12a) \]

\[ (2.12b) \]

the Runge-Kutta part of a Pouzet-type method is completely characterized by the parameters of the differential equation method (2.12) (we note in passing that, for kernels K(t, s, y) satisfying \( \partial K/\partial t = 0 \), the methods (2.10) and (2.11) coincide). This observation can be used to establish results on the attainable order of both explicit and implicit methods. The following table summarizes results on the order of VRK methods of the above forms. Details may be found in [C20] which deals with the general theory of VRK methods.

**TABLE 1. Attainable order of VRK methods**

<table>
<thead>
<tr>
<th>m</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit PRK: p</td>
<td>( \leq 1 )</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>implicit PRK: p</td>
<td>( \leq 2 )</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>explicit BRK: p</td>
<td>( \leq 1 )</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>implicit BRK: p</td>
<td>( \leq 2 )</td>
<td>2</td>
<td>2</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

There are essentially two ways for choosing the discretization \( \hat{F}_{n}^{(i)}(t) \) in (2.9), (2.10), and (2.11) of the exact history term \( F_{n}^{(i)}(t) \) in (2.7): \( \hat{F}_{n}^{(i)}(t) \) can either be obtained by using, on each subinterval \([t_k, t_{k+1}^{+}] \) of \([t_0, t_n] \), a quadrature formula based on the appropriate Runge-Kutta parameters \( (b_j) \), \( (c_j) \) of (2.9b), (2.10b), (2.11b), or, more efficiently (with regard to the number of kernel evaluations) be relying on suitable quadrature formulas using as abscissas only the mesh points \( \{t_0, \ldots, t_n\} \).

In the former case we speak of an extended VRK method while in the latter case we have a mixed (quadrature) VRK method; here, the class of \( (\rho, \sigma) \)-reducible quadrature formulas play an important role. Details on the selection of these approximation procedures may be found in [A4] (pp. 849-864), [C40], [C3], [C42], and [C1]. In section 2.3 (Numerical stability) a third technique for approximating the history term \( F_{n}^{(i)}(t) \) in (2.7) (leading to so-called \( \gamma \)-modified VRK methods) will be mentioned.

A subset of the class of implicit PRK methods can be viewed as collocation methods in certain piecewise polynomial spaces; this interpretation can also be used to establish general results on the attainable order of such methods (viz. the second line in Table 1). Let \( Z_n := \{t_n : n = 1(1)N\} \), and set \( \sigma_0 := \{0, t_1\} \), \( \sigma_n := \{t_n, t_{n+1}\} (n = 1(1)N-1) \). We denote by \( S_m^{(d)}(Z_n) := \{u \in C(I) : u|_{\sigma_n} =: u_n \in \Pi_m, n = 0(1)N-1\} \) the linear space of piecewise polynomials of degree \( m \geq 0 \) which have \( Z_n \) as their knots and which lie in the continuity class \( C^d(I) \), \(-1 \leq d \leq m-1\). For \( d = m-1 \) we have the classical polynomial splines of degree \( m \). However, for the numerical solution of Volterra equations the cases \( d = 0 \) (continuous piecewise polynomials) and \( d = -1 \) (piecewise polynomials possessing finite jumps at their knots) are the most important ones. Observe that the dimension of \( S_m^{(d)}(Z_n) \) is given by \( \dim S_m^{(d)}(Z_n) = (N-d)+d+1 \); hence we have \( \dim S_m^{(d-1)}(Z_n) = N+m, \) \( \dim S_m^{(0)}(Z_n) = N+m+1 \), and \( \dim S_m^{(-1)}(Z_n) = N(m+1) \).

Collocation in a given space \( S_m^{(d)}(Z_n) \) now consists in determining an element \( u \) in this space which satisfies \( (1.3) \) on a finite subset of \( I \) whose cardinality is consistent with the dimension of the space. i.e. we replace (2.7) by

\[ u_n(t) = \hat{F}_{n}^{(1)}(t) + \int_{0}^{t} K[t, s, u_n(s)]ds, \quad t \in X_n^{(d)}(n = 0(1)N-1), \]

\[ (2.13) \]

where

\[ \hat{F}_{n}^{(1)}(t) := g(t) + \int_{0}^{t} K[t, s, u(s)]ds, \]

where the set of collocation points is

\[ X_n^{(d)} := \{t_n + c_h : 0 \leq c_1 < \ldots < c_{m-d} \leq 1\} \subset \sigma_n. \]

It is clear that, in general, the integrals in (2.13), i.e.

\[ \int_{0}^{t} K[t_n + c_h t_n + rh, u_n(t_n + rh)]dr (i = 1(1)m) \]

have to be evaluated by numerical quadrature; if we use, for the case \( d = -1 \), an interpolatory quadrature formula based on the m abscissas \( \{t_n + c_h t_n : i = 1(m)\} \), then (2.13) reduces to the PRK method (2.10). For details (and for results on the corresponding attainable order) we refer to [C18], [C21], [A19], related methods (using other piecewise polynomials) may be found in [B12] (where it is shown that classical splines with \( m > 2 \) and \( d = m-1 \) lead to divergent methods), [C39]
To conclude we mention a somewhat different technique given by (or can be approximated by) the finite representation of the kernel since the kernel becomes unbounded at \( s = t \).

The previous section cannot be applied to equations (1.3) with \( 0 < a \leq 1 \) since the kernel becomes unbounded at \( t = t_0 \); see [C38] for details and for an introduction to numerical stability with respect to the model problem (2.15) and for specific stability results (including regions of absolute stability) we refer to [A4] (pp. 788-825), [C52], [C56], [C57], [C40], [C6], [C7], [C1]; we also mention the earlier analyses of [C46] and [C58]: here, the concept of the so-called (exact)-repetition factor of a direct quadrature method (i.e. the smallest positive integer \( r \) for which the weights \( \{w_{n,j}\} \) satisfy \( w_{n+r,j} = w_{n,j} \) for all \( n \geq n_0 \) and \( n_1 \leq j \leq n - n_2 \), with \( n_1 \) independent of \( n \)) was conjectured to have a bearing on the stability or instability of a given method. The role of this repetition factor in the numerical stability of quadrature methods for second-kind Volterra equations has been analyzed in [C53] and, especially, in [C66].

As far as (mixed) VRK methods are concerned it has recently been shown ([C41], [C45]) that so-called \( \gamma \)-modified methods possess considerably better stability properties (but may suffer from a slight loss of order). In such a \( \gamma \)-modified method the term \( \bar{F}_n(\cdot) \) approximating the exact lag term \( F_n(\cdot) \) is replaced by

\[
\bar{F}_n^\gamma(t) := \bar{F}_n(t) + \gamma(t) [y_n - \bar{F}_n(t_n)],
\]

where the function \( \gamma(t) \) in front of the perturbation term \( y_n - \bar{F}_n(t_n) \) is suitably chosen. (See [C42] for an analysis of the important special case \( \gamma(t) \equiv 1 \).) This idea of modifying the approximate history term \( \bar{F}_n(\cdot) \) is also used in the promising class of so-called modified multilag methods ([C68]): these methods combine the advantages (i.e. computational efficiency and favorable stability behavior) of multilag methods and \((\rho, \alpha)\)-reducible methods. A multilag method is defined by

\[
y_n = g(t_n) + \sum_{i=1}^{k} a_{k,i} \bar{T}_n(t_n - \tau_i) + h \sum_{i=0}^{k} \beta_{k,i} K(t_n + \tau_i, y_n),
\]

with given starting values \( \{y_0, \ldots, y_{n_k-1}\} \). Here, the lag term \( \bar{T}_n(t) \) has the form

\[
\bar{T}_n(t) := h \sum_{j=0}^{n_k} w_{n,j} K(t, t_j, y_j).
\]

These methods are based on an idea introduced by Pouzet ([C62]): if we set

\[
\Psi(u, t) := \int_0^t K(t, s, y(s)) \, ds,
\]

we may regard \( \Psi(u, t) \) as the solution of the initial-value problem \( d\Psi(u, t)/du = K(t, u, y(u)) \), \( \Psi(0, t) = 0 \), where
3. VOLterra INTEGRAL EQUATIONS OF THE FIRST KIND

It is clear from the previous section that many of the discretization techniques used for (nonlinear) second-kind Volterra integral equations can be applied to equations of the first kind, as well as to Volterra integrodifferential equations (section 4). However, there are some significant differences, especially with regard to the construction of higher-order direct quadrature methods for the case \( \alpha = 0 \) in (1.4). On the other hand, if the kernel \( G(t,s) \) and the function \( g(t) \) are known analytically and if their derivatives \( \partial G(t,s)/\partial t \) and \( g'(t) \) exist on \( S \) and \( I \), respectively, then (1.4) \((\alpha = 0)\) is equivalent to the second-kind equation

\[
y(t) = g(t)/G(t,t) - \int_0^t \frac{G(t,s)/G(t,t)}{G(t,s)} y(s) \, ds, \quad t \in I
\]

(3.1)

(or to a similar equation if \( \partial G(t,s)/\partial s \) exists; if \( G(t,t) \neq 0 \) for all \( t \in I \)) may be treated by the methods discussed in section 2.

For the discussion of direct methods for (1.4) we shall, as in section 2, treat the cases \( \alpha = 0 \) and \( 0 < \alpha < 1 \) separately; a further class of equations of the first kind with singular kernels will be dealt with in section 4. A recent survey of the numerical analysis of equations of the form (3.1) has been given in [A25]; compare also [A16], [A28] and, especially, [A23] for Abel-type equations.

3.1. The case (1.4) with \( \alpha = 0 \)

The application of direct quadrature methods to (1.4) leads to the triangular system

\[
h \sum_{j=0}^n w_{nj} G(t_n,t_j) y_j = g(t_n) \quad (n = r(1)N; r \geqslant 1), \quad (3.2)
\]

where the starting values \( \{y_0,\ldots,y_{r-1}\} \) have to be prescribed. The convergence properties of the trapezoidal method and the midpoint method (which is known to be preferable to the former) have been studied in [D22], [D23], and [D17]; compare also [A25] for a summary of these results. As far as methods of order \( p \geqslant 2 \) are concerned, it was already observed in [D23] that all the classical (interpolatory) quadrature formulas (like the Simpson rules and the Gregory formulas) lead to divergent quadrature methods when used for first-kind Volterra equations (a result which is in sharp contrast to the situation encountered in equations of the second kind). It was subsequently shown in [D15] that all direct quadrature methods using interpolatory quadrature formulas of degree of precision exceeding one are necessarily divergent. Convergent higher-order methods of the form (3.2) were constructed in [D13], [B8], [D14] (by “stabilizing” the Gregory formulas of order \( 3 \leqslant p \leqslant 6 \); due to their simple structure these formulas allow for a convenient implementation), and in [B11], [D16]; compare also [D28] (where “inverted” backward differentiation formulas are used), [D25], and [D1] (methods based on cyclic linear multistep methods) for related ideas.
leading to convergent higher-order methods for (1.4), $\alpha = 0$. In addition we have to mention the recent work of [B26] (pp. 49-69) generalizing the results of [D15], [D14] and [D28]; the author shows the existence of convergent $(\alpha, \alpha)$-reducible quadrature methods of arbitrary order and derives specific methods (of orders $p \leq 6$) which are reducible to the corresponding backward differentiation formulas.

Another approach permitting the construction of convergent high-order methods is based on collocation techniques in the piecewise polynomial spaces $S^{(-1)}_m(\mathbb{Z}_N)$ and $S^{(0)}_m(\mathbb{Z}_N)$. In analogy to (2.13) the approximating element $u$ is determined by (we illustrate the case $d = -1$)

$$
\int_0^{t_n} G(t_n + c_i h, t_n + c_j h) u_n(t_n + c_j h) \, dr = \Phi_n(t_n + c_i h),
$$

where $\Phi_n(t) := \int_0^t G(t,s) u(s) \, ds$ $(n = 0(1)N-1)$. (3.4)

If the collocation parameters $\{c_i : 0 < c_1 < \ldots < c_{m+1}\}$ satisfy the condition $\prod_{i=1}^{m+1} (1-c_i) c_i < 1$ (which holds trivially if $c_{m+1} = 1$) then, under the usual conditions on $G(t,s)$ (e.g. $G(t,t) \neq 0$ on $I$), the error $e(t) := y(t) - u(t)$ satisfies $e(t_n) = o(h^{m+1})$ for $t_n \in \mathbb{Z}_N$. (We recall that the same convergence order holds for second-kind equations but, in contrast to here, for any choice of the (distinct) parameters $\{c_i\}$.)

In the practical implementation of these collocation methods the moment integrals occurring in (3.3) and (3.4) have to be approximated by numerical quadrature. Certain special choices of (interpolatory) quadrature formulas lead to previously analyzed product integration methods ([D24]), to block-by-block methods ([D18], [D19]; see also [D29] for first-kind equations in $\mathbb{R}^2$), or to the semi-explicit methods of [D21]; compare [D7] and [D8] for details of these connections, and see [D9], [D10] for additional results (showing that, for the mesh points $Z_m$, ($m+1$)-st order convergence is best possible; in the case of Volterra integral equations of the second kind with $\alpha = 0$ in (1.3), a judicious choice of the parameters $\{c_i\}$ yields the convergence orders $p = 2m+1$ (Radau points for $[0,1]$) and $p = 2m$ (Lobatto points). Compare [C18], [C21], [A19] for details). As far as methods of Runge-Kutta type are concerned, i.e. methods of the form

$$
h \sum_{j=1}^{m} a_{ij} G(t_n + d_j h, t_n + c_j h) Y_j(n) = \Phi_n(t_n + c_i h),
$$

(3.5)

with $y_{n+1} = y^{(n)}_{m+1}$, a general theory along the lines of that given in [C20] is still lacking. An explicit third-order method is analyzed in [D21].

3.2. The case (1.4) with $0 < \alpha < 1$ (Abel equations)

If we apply collocation in $S^{(-1)}_m(\mathbb{Z}_N)$ to (1.4) then the recursion (3.3) is replaced by

$$
h^{1-\alpha} \int_0^{t_n} G(t_n + c_i h, t_n + c_j h) \int_0^{t_n + c_j h} u_n(t_n + c_j h) \, dr = \Phi_n(t_n + c_i h),
$$

(3.6)

where the meaning of the right-hand side is clear. Discretization of the moment integrals on the left-hand side will in general be based on product integration techniques (see also [D17], [D7]); special cases have been studied in [D30] (for $S^{(-1)}_0(\mathbb{Z}_N)$, yielding the weighted midpoint method), in [B3] and [D3] (using product integration based on certain Gauss formulas to obtain methods of order $p = 2$ and $p = 3$; we note that [B3] contains also existence and uniqueness results for equations of the form $(V\gamma)(t) = g(t), \alpha = 1/2$). A comprehensive discussion of the errors arising in the numerical integration of singular integrals of the above form and the application to the numerical treatment of Abel equations has been given in [B2]. In addition compare the analysis in [D12] and [D4] (Galerkin techniques).

4. VOLTERA INTEGRO-DIFFERENTIAL EQUATIONS

We illustrate the treatment of (1.5) by considering the case $\alpha = 0$. Equation (1.5) may then be written as

$$
y'(t) = F[t, y(t)] + y(t), t \in I, y(t_0) = y_0,
$$

(4.1)

with

$$
T_n(t) := \int_0^{t_n} K[t, s, y(s)] \, ds, t \in [t_n, t_{n+1}],
$$

(4.2)

or, alternatively, as a system of two coupled Volterra integral equations,

$$
y(t) = y_0 + \int_0^t \{F[s, y(s)] + z(s)\} \, ds,
$$

$$
z(t) = \int_0^t K[t, s, y(s)] \, ds, t \in I
$$

(4.2)

There are basically three different ways of discretizing Volterra integro-differential equations, namely

(i) by applying direct quadrature methods (see section 2), combined with a linear k-step method for the differential expression, yielding

$$
\sum_{j=1}^{k} \beta_j Q(t_n + j h) Y_n(j h) = \sum_{j=0}^{k} \alpha_j \gamma_n + \eta
$$

(3.4)

where

$$
\gamma_n := h \sum_{i=0}^{m} w_{mi} K(t_{mi}, t_{m+1}, y_{m+1}), m = r \geq 0, \text{ and}
$$

$$
Q(t, y(t), z(t)) := F(t, y(t)) + z(t), \text{ with } z(t) \text{ defined above;}
$$

Journal of Computational and Applied Mathematics, volume 8, no 3, 1982 219
which can be reduced to a second-order ordinary initial-value
\[ y'(t) = XY(t) + \int_{t_0}^{t} y(s) \, ds, \quad y(t_0) = y_0 \quad (X < 0, Y < 0), \]
is independent of \( t \) and \( s \).

The development of automatic software for the three types of (singular and nonsingular) Volterra functional equations (1.3), (1.4), and (1.5) is still in its infancy, due to the fact that one still does not possess the tools to deal with the control of the local error and, closely related to this problem, with the influence of local perturbations on the behavior of the approximate solution generated by a given method. A glance at the surveys [F5], [F6], and [F3] on software for Fredholm and Volterra equations shows that a major effort will be needed in this area. At present there exist only a few isolated automatic codes which will be listed below; see also the following section for some additional remarks.

5.1. Volterra integral equations of the second kind

The collection of ALGOL procedures listed under [F8] contains codes for extended and mixed PRK methods (the latter based on an Adams formula); an instability problem occurring in the mixed PRK method has been straightened out in [C7]. The report [F2] contains a program for (extended and mixed) explicit PRK methods based on a Fehlberg-type imbedding approach; this work has its origin in the paper [C49]. A code also using imbedding techniques has already been mentioned in section 2.1, namely [A18] (pp. 70-79). More recently, two related codes (for regular and weakly singular kernels) have been derived in [B17]; they are based on the (product) Simpson formulas and use automatic stepsize control. In addition compare also [B20], [B6] (Hermite-type collocation), [C36] (extrapolation techniques), and [B25] (predictor-corrector procedures using variable step sizes), and the rather different approach of [F12] (and [F13]) which makes use of computer algebra to generate analytic approximations for various types of (first and second kind) equations. However, the most promising code is possibly the one in [F1] : here (see also the final paragraph in section 2.1) the given kernel is replaced by a finitely decomposable one, and the corresponding equivalent system of ordinary differential equations (whose dimension depends on the number of terms in the approximating kernel) is solved by certain standard codes for ordinary differential equations (Runge-Kutta-Fehlberg, Adams). Whether or not this approach can eventually be used also for weakly singular equations (1.3) is not yet clear (since selecting a suitable kernel approximation is already the major difficulty in the case \( \alpha = 0 \)). Finally, we like to mention that a test set of 15 second-kind equations is given in [C8] where the performance of various PRK methods is evaluated.

5.2. Volterra integral equations of the first kind

Since Volterra and Abel equations of the first kind tend to be ill-posed problems (although not quite as seriously as first-kind Fredholm equations; see [G3] for an illuminating discussion) there are two widely different classes of problems to be dealt with, namely the class of those equations where the kernel \( G \) and the nonhomogeneous term \( g \) are given analytically, and the (practically much more important) class of equations where \( g \) (and possibly also \( G \)) are obtained from measurements and are thus subject to random errors (noise). This class of equations will be considered in section 6; as for the former class there exist often explicit inversion formulas for the solution (see [H1]). For more complicated equations there exist the automatic programs of [F11] (modified midpoint method using Richardson extrapolation) for equations with continuous kernels, and of [B3] for nonlinear Abel equations.
5.3. Volterra integro-differential equations

In [F8] there may be found two codes for Volterra integro-differential equations of first and second order (based on Runge-Kutta techniques); the more recent code for general functional differential equations given in [F7] can be adapted to deal with equations of the form (1.5), (α = 0).

6. FIRST-KIND INTEGRAL EQUATIONS WITH NOISY DATA

In most applications of first-kind Volterra-type integral equations the non-homogeneous term g(t) is not given analytically on I but can only be determined experimentally on certain finite subsets I_N; the measured values \( \tilde{g}(t) \), \( t \in I_N \), are thus subject to noise. For equations with continuous kernels the problem of numerical differentiation of \( g(t) \), based on the data \( \tilde{g}(t) \), is probably the most celebrated problem: certain methods are based on the corresponding Volterra equation

\[
\int_0^x y(s) \, ds = g(t), \quad t \in I, \text{ with only } \tilde{g}(t), \ t \in I_N \text{ known.}
\]

(6.1)

Compare [G9] (regularization techniques), [G5] and [G4] (for methods based on regularization and spectral differentiation); see also [H1], [G7] and [F9] (where an automatic code based on a regularization method originally devised for Fredholm equations of the first kind is given). Many physical problems lead to singular Abel-type integral equations whose underlying integral operators are no longer compact (as in (1.4)); these equations are

\[
\int_0^x \frac{k_1(t)k_2(s)}{(sP-xP)\alpha} y(s) \, ds = g(t), \quad t \in [0,T] \text{ and } \alpha > 0, 0 < \alpha < 1.
\]

(6.2)

\[
\int_0^T \frac{k_1(t)k_2(s)}{(sP-xP)\alpha} y(s) \, ds = g(t), \quad t \in [0,T] \text{ and } \alpha > 0, 0 < \alpha < 1.
\]

(6.3)

Here, \( k_1 \) and \( k_2 \) are given continuous functions; the most common values for \( p \) and \( \alpha \) are \( p = 2, \alpha = 1/2 \) (see also [D2], [D6], [D20]). An excellent introduction to various aspects regarding the application, ill-posed nature, and the numerical analysis of these equations see [H1] (but compare also [G3] and [G10]). Among the methods proposed for (6.2) and (6.3) we mention the papers [G8] (using least-squares polynomial approximation), [H36] (where a code based on a modification of the method in [G8] is described), [G1] and [G2] (methods based on spectral differentiation), and [F10] (here, a code based on spline approximation in the explicit inversion formula is given; see also [D27]).

7. APPLICATIONS OF VOLterra EQUATIONS

Due to limitation of space we shall not be able to discuss the wide spectrum of applications of Volterra integral and integro-differential equations in any detail. Instead, the items listed in Section H of the bibliography have been chosen to reflect the typical areas in which modelling processes lead to such functional equations. Also, many of the articles listed contain themselves extensive bibliographies on additional sources of Volterra equations; among these we should like to single out [H1], [H13], [H14], and [H30].

ACKNOWLEDGEMENTS

Part of the material in this survey paper was presented in an invited talk at the meeting of the Belgian Contact Group on Applied Mathematics held at the University of Antwerp in March 1980; the author is indebted to Professor Luc Wuytack for providing this stimulus leading to the writing of the paper. The author is also grateful to the Department of Mathematics of the University of Manchester (particularly to Dr. C. T. H. Baker) where part of this paper was written during two visits (in 1980 and 1981) made possible by an SRC Senior Fellowship.

REFERENCES

A. Monographs, proceedings, surveys


B. Recent doctoral dissertations


B15. KUMAR S. : Numerical methods for Volterra and singular integral equations, India Institute of Technology, Delhi, 1981.

B16. LINZ P. : Numerical methods for Volterra integral equations with applications to certain boundary
value problems, University of Wisconsin, Madison, 1968.

B17. LOGAN J. E.: The approximate solution of Volterra integral equations of the second kind, University of Iowa, Iowa City, 1976.


C. Volterra integral equations of the second kind


C45. VAN DER HOUWEN P. J., WOLKENFELT P. H. M. and BAKER C. T. H.: Convergence and stability analysis for modified Runge-Kutta methods in the numerical treatment of second kind Volterra-in-


C59. OULÈS H. : Résolution numérique d'une équation intégrale singulièere, Rev. Française Traitement In-


D. Volterra integral equations of the first kind


F. Software


F7. NEVES K. W.: Automatic integration of function-


G. Equations of the first kind with noisy data


H. Applications of Volterra equations


H14. DIEKMANN O. : Integral equations and popula-


