LINEAR MULTISTEP METHODS FOR VOLterra INTEGRAL EQUATIONS
OF THE SECOND KIND

Preprint

amsterdam 1982
LINEAR MULTISTEP METHODS FOR VOLterra INTEGRAL EQUATIONS
OF THE SECOND KIND

P.J. VAN DER HOUWEN & H.J.J. TE RIELE

Preprint

kruislaan 413  1098 SJ  amsterdam
Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 65R20
near multistep methods for Volterra integral equations of the second kind *)

J. van der Houwen & H.J.J. te Riele

Abstract

A general class of linear multistep methods for second kind Volterra integral equations is described which include the conventional direct quadrature methods. A convergence theorem is presented and stability is investigated with respect to the linear convolution equation. Examples are given of \( V_0 \)-stable methods. The theory is illustrated by a few numerical examples.

Words & Phrases: numerical analysis; Volterra integral equations of the second kind; linear multistep methods; \( V_0 \)-stability; convergence

*) This report will be submitted for publication elsewhere.
1. INTRODUCTION

The simplest linear multistep (LM) method for solving the Volterra integral equation of the second kind

$$y(t) = g(t) + \int_{t_0}^{t} K(t, \tau, y(\tau)) d\tau, \quad t \in I := [t_0, T],$$

(1.1)

is obtained by writing down this equation in a sequence of equidistant points

$$t_n := t_0 + nh, \quad n = 0(1)N \quad (h \text{ fixed and } t_N = T) \quad (1.2)$$

and by approximating the integral term by some suitably chosen quadrature formula. Such a method is called a direct quadrature (DQ) method for (1.1). Recently, several other LM methods for solving (1.1) have been proposed (cf. the indirect backward differentiation method in [5] and the multilag and modified multilag methods in [9] (see also [12]).

In this paper a general class of linear multistep methods is presented which includes all these methods, and many others (Section 2). This enables us to give a uniform treatment of the problems of consistency (Section 3), of convergence (Section 4) and of stability (Section 6). Since the ordinary differential equation $dy/dt = f(t, y)$, $y(t_0) = y_0$, is a special case of the differentiated version of (1.1), the relation with linear multistep methods for ODEs is analyzed and fixed terms recurrence relations are derived for a class of convolution kernels (Section 5). Finally, two numerical experiments are reported (Section 7).

Space prevents us to include the detailed proofs of the theorems presented here. These may be found in [6]. A number of additional numerical experiments which support and confirm the theory, may also be found in [6].

The work presented here can easily be extended to Volterra integral equations of the first kind, and to Volterra integro-differential equations (cf. [6]).
2. A GENERAL CLASS OF LM METHODS FOR SOLVING (1.1)

Let us associate with (1.1) the so-called lag term

\[ Y(t,s) := g(t) + \int_{t_0}^{t} K(t,\tau, y(\tau))d\tau \quad (2.1) \]

for \((t,s) \in S := \{(t,s): t_0 \leq s \leq t \leq T\}\). Note that \(Y(t,t) = y(t)\). Let \(y_n\) and \(Y_n(t)\) denote numerical approximations to \(y(t_n)\) and to \(Y(t,t_n)\), respectively, and let

\[ K_n(t) := K(t,t_n,y_n), \quad n \geq 0. \quad (2.2) \]

Usually, \(Y_n(t)\) will be computed by a quadrature formula of the form

\[ Y_n(t) = g(t) + \int_{0}^{t} K(t,\tau,y(\tau))d\tau \approx \sum_{j=0}^{n} w_n,j \cdot K(t,t_j,y(t_j)), \quad n \geq n_0, \quad (2.3) \]

where the \(w_n,j\) are given weights and \(n_0\) is sufficiently large to ensure the required order of accuracy. We assume that this quadrature formula is of order \(r\), i.e.,

\[ E_n(h;t) := \int_{0}^{t} K(t,\tau,y(\tau))d\tau - \sum_{j=0}^{n} w_n,j \cdot K(t,t_j,y(t_j)) = 0(h^r) \quad (2.4) \]

as \(h \rightarrow 0\), \(n \rightarrow \infty\), with \(t_n = t_0 + nh\) fixed. Our general LM method for (1.1) consists of the quadrature formula (2.3) and the LM formula

\[ \sum_{i=0}^{k} a_i y_{n-i} + \sum_{j=-k}^{k} \beta_{ij} y_{n-j}(t_n+j) = \]

\[ = h \sum_{i=0}^{k} \sum_{j=-k}^{k} \gamma_{ij} y_{n-i}(t_{n+j}), \quad n = k(1)N, \quad k \text{ fixed}, \quad (2.5) \]

where the parameters \(a_i\), \(\beta_{ij}\) and \(\gamma_{ij}\), \(i=0(1)k, j=-k(1)k\), are to be prescribed. From this scheme the quantities \(y_k, y_{k+1}, \ldots, y_N\) can be computed successively. The quantities \(y_1, \ldots, y_{k-1}\) and \(Y_1(t), \ldots, Y_{k-1}(t)\) are assumed to be precomputed by some starting method. Since the kernel \(K(t,\tau,y)\) is not necessarily defined
outside \( S \), we usually require (cf. Figure 1) that \( \beta_{ij} = \gamma_{ij} = 0 \), for \( j < -i \).

![diagram](image)

**FIG. 1.** Points in the \((t,s)\)-plane needed in (2.5) for \( k = 2 \)

Furthermore, it will be assumed that the points \( t_j \) are equally spaced (cf. (1.2)) although most of the analysis can be carried through for non-uniform spacing (compare a similar situation in the analysis of LM methods for ODEs). It is convenient to characterize the formula (2.5) by the matrices

\[
A = (a_{ij}), \quad B = (\beta_{ij}), \quad C = (\gamma_{ij})
\] (2.6)

where the row index \( i \) assumes the values \( O(1)k \) and the column index \( j \) the values \( -k(1)k \). We now describe four subclasses of (2.5) from which we will borrow several illustrating examples in this paper.

### 2.1 Direct quadrature methods

Consider the LM formula defined by the \((1x1)\) matrices

\[
A = 1, \quad B = -1, \quad C = 0
\] for which (2.5) reduces to

\[
y_n = Y_n(t_n).
\] (2.7)

Evidently, this is the direct quadrature (DQ) method, described in the introduction.

### 2.2 Indirect linear multistep methods

We formally derive this subclass by applying a linear multistep method for ODEs with coefficients \( a_i \) and \( \gamma_i \), \( i = 0(1)k \), to the differentiated version of (1.1) (cf. [5]):
\[ y'(t) = K(t,t,y(t)) + Y_t(t,t), \quad (2.8) \]

where \( Y_t(t,t) \) denotes the partial derivative of \( Y(t,s) \) with respect to its first variable \( t \), in the point \( (t,t) \). This yields the scheme

\[
\sum_{i=0}^{k} a_i y_{n-i} = h \sum_{i=0}^{k} \gamma_i K_{n-i}(t_{n-i}) + h \sum_{i=0}^{k} \gamma_i Y_t(t_{n-i}, t_{n-i}), \quad n \geq k. \quad (2.9)
\]

Now we approximate the derivative \( Y_t \) of \( Y \) by the \( k \)-step forward differentiation formula (cf. [1, Table 25.2])

\[
Y_t(t_{n-i}, t_{n-i}) \approx \frac{1}{h} \sum_{\ell=0}^{k} \delta_{\ell} Y(t_{n+\ell-i}, t_{n-i}). \quad (2.10)
\]

Using (2.3) we obtain

\[
\sum_{i=0}^{k} a_i y_{n-i} + \sum_{i=0}^{k} \sum_{j=-i}^{k-i} \gamma_i \delta_{i+j} Y_{n-i}(t_{n+j}) = h \sum_{i=0}^{k} \gamma_i K_{n-i}(t_{n-i}), \quad n \geq k, \quad (2.11a)
\]

or, equivalently, the generating matrices

\[
A = \begin{pmatrix}
\alpha_0 \\
\alpha_1 \\
\vdots \\
a_k
\end{pmatrix}, \quad B = \begin{pmatrix}
0 & \gamma_0 \delta_0 & \gamma_0 \delta_1 & \cdots & \gamma_0 \delta_k \\
\gamma_1 \delta_0 & \gamma_1 \delta_1 & \cdots & \gamma_1 \delta_k \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\gamma_k \delta_0 & \gamma_k \delta_1 & \cdots & \gamma_k \delta_k
\end{pmatrix}, \quad C = \begin{pmatrix}
0 & \gamma_0 \\
\gamma_1 & 0 \\
\vdots & \vdots \\
\gamma_k & 0
\end{pmatrix}.
\]

These matrices generate an *indirect linear multistep* (ILM) method. When the \( \alpha_i \) and \( \gamma_i \) are the coefficients of a backward differentiation method, (2.11) represents the IBD (indirect backward differentiation) method, analyzed in [5]. We notice that for this IBD method we have \( \gamma_i = 0 \), \( i = 1(1)k \), and \( \gamma_0 \delta_j = a_j \), \( j = 0(1)k \).

### 2.3 Multilag methods

In Wolkenfelt et al. [12] we find methods which can be characterized by the matrices

\[
A = \begin{pmatrix}
\alpha_0 \\
0 \\
\vdots \\
a_k
\end{pmatrix}, \quad B = \begin{pmatrix}
0 & \alpha_1 & 0 \\
0 & \vdots & 0 \\
0 & \alpha_k
\end{pmatrix}, \quad C = \begin{pmatrix}
\gamma_0 \\
\gamma_1 \\
\vdots \\
\gamma_k
\end{pmatrix}. \quad (2.12)
\]

Here, the \( \alpha_i \) and \( \gamma_i \), \( i = 0(1)k \), may be the coefficients of any LM
method for ODEs. If the lag term \( Y(t) \) is computed by using a quadrature rule which is \((a,\gamma)-\)reducible (see Section 2.5), then the resulting method turns out to be equivalent to a DQ method based on the same \((a,\gamma)-\)reducible quadrature rule (provided, of course, that the starting values are identical). Thus, a different implementation of the same method was used for the stability analysis of DQ methods. However, as was pointed out by Wolkenfelt [9], this implementation requires a lot of additional arithmetic operations and, although suitable for theoretical analysis, it is not recommendable in actual computations. In order to avoid this disadvantage, he proposed to compute the lag term simply by a quadrature rule of the form (2.3) to obtain the multilag (ML) methods.

2.4 Modified multilag methods

In [9] Wolkenfelt also introduced a modification of the ML methods, viz., the so-called modified multilag (MML) methods, characterized by the matrices

\[
A = \begin{pmatrix}
\alpha_0 \\
\alpha_1 \\
\vdots \\
\alpha_k
\end{pmatrix}, \quad B = \begin{pmatrix}
0 & -\alpha_1 & \alpha_1 \\
\vdots & \ddots & \vdots \\
0 & \ddots & 0 \\
-\alpha_k & \alpha_k & \gamma_k
\end{pmatrix}, \quad C = \begin{pmatrix}
\gamma_0 \\
\gamma_1 \\
\vdots \\
\gamma_k
\end{pmatrix}.
\]  

(2.13)

The \( \alpha_i \) and \( \gamma_i \) are, again, the coefficients of any LM method for ODEs.

2.5 The quadrature weights of the lag term

In order to define a specific LM method for (1.1) we have not only to specify the generating matrices \( A, B \) and \( C \), but also the quadrature weights \( w_{n,j} \) in (2.3). An important family of quadrature formulas, including the well-known Gregory quadrature formulas, are the so-called reducible quadrature formulas [8]. The weights \( w_{n,j} \) in such formulas are recursively defined by the equations

\[
\sum_{i=0}^{k^*} a_i w_{n-i,j} = \begin{cases}
0 & \text{if } j = 0(1)n-k^*-1 \\
b_j & \text{if } j = n-k^*(1)n
\end{cases}, \quad n = k^*, k^*+1, \ldots
\]  

(2.14)

where the \( a_i \) and \( b_j, i = 0(1)k^* \), are the coefficients of some given LM method for ODEs. Here, we define \( w_{n,j} = 0 \) for \( j > \max(n,k^*-1) \), and the "starting weights" \( w_{n,j} \), \( 0 \leq n,j \leq k^*-1 \) are assumed to be prescribed.
Defining the characteristic polynomials

\[ \rho(z) := \sum_{i=0}^{k^*} a_i z^{k^*-i}, \quad \sigma(z) := \sum_{i=0}^{k^*} b_i z^{k^*-i}, \quad (2.15) \]

the quadrature formulas generated by (2.14) are said to be \((\rho, \sigma)\)-reducible. We note that the characteristic polynomials of the Adams–Moulton methods generate the weights of the Gregory formulas. The backward differentiation methods generate rather unconventional quadrature rules, which were analysed in [11].

3. CONSISTENCY OF THE LM FORMULA (2.5)

Let us associate with the LM formula (2.5) the difference-differential operator \( L_n \) defined by

\[ L_n(Y) := \sum_{i=0}^{k} \left\{ a_i Y(t_{n-i}, t_{n-i}) + \sum_{j=-k}^{k} [\beta_{ij} - \gamma_{ij} h \frac{\partial}{\partial s}] Y(t_{n+j}, t_{n-i}) \right\}, \quad (3.1) \]

where \( Y(t,s) \) is an arbitrary function, differentiable with respect to \( s \) on \( t_0 \leq s \leq T \). As in the case of LM methods for ODEs, the operator \( L_n \) is introduced in order to operate on test functions \( Y \) of sufficient differentiability (cf. e.g. Lambert [7,p.23]). Unlike the ODE case, the relation of the operator \( L_n \) with the LM formula (2.5) is not immediate, and needs some explanation. Suppose that \( Y(t,s) \) is defined by (2.1) with \( y(t) \) the exact solution of (1.1). Observing that \( Y(t,t)=y(t) \) and \( a Y(t,s)=K(t,s,y(s)) \), and using (2.4) we find, on substitution of \( y(t) \) into (2.5), the equation

\[ \sum_{i=0}^{k} \left\{ a_i y(t_{n-i}) + \sum_{j=-k}^{k} [\beta_{ij} y(t_{n+j}) - h \gamma_{ij} K(t_{n+i}, t_{n+i})] \right\} = \]

\[ L_n(Y) - \sum_{i=0}^{k} \sum_{j=-k}^{k} \beta_{ij} E_{n-i}(h; t_{n+j}). \quad (3.2) \]

Thus, the exact solution of (1.1) satisfies the method (2.3)-(2.5) apart from the residual terms in the right-hand side of (3.2). In this section, we concentrate on the first residual term.

Definition 3.1 The operator (3.1) and the associated LM formula (2.5) are said to be consistent of order \( p \) if for all \( Y \in C^{p+1}[S], \ L_n(Y)=O(h^{p+1}) \) as \( h \to 0 \) with nonvanishing error constant.

If \( Y \) corresponds to the theoretical solution of (1.1), then \( L_n(Y) \) will be called the local truncation error of (2.5).
The following theorem provides the consistency conditions in terms of the parameters \(a_i\), \(\beta_{ij}\) and \(\gamma_{ij}\).

**Theorem 3.1** The operator (3.1) and the associated LM formula (2.5) are consistent of order \(p\) if

\[
\sum_{i=0}^{k} \sum_{j=-k}^{k} \left[ (-i)^q a_i - (-i)^{-q} (-i)^{2-1} \left( i\beta_{ij} + \varepsilon \gamma_{ij} \right) \right] = C_q = 0 \quad (3.3)
\]

for \(q = O(1)p\) and \(\varepsilon = O(1)q\) (with \((-i)^{q-1} \varepsilon : = 0\) if \(i = 0\)).

**Corollary 3.1** Let \(\tilde{p}\) be the order of consistency of the LM method for ODEs defined by the coefficients \(\{a_i, \gamma_i\}\) employed in Sections 2.2, 2.3 and 2.4. Then the order of consistency \(p\) of the LM formula (2.5) for (1.1) is given by \(p = \tilde{p}\) for the DQ method, \(p = \min\{k, \tilde{p}\}\) for the ILM method, and \(p = \tilde{p}\) for both the ML method and the MML method.

If the LM formula (2.5) is consistent of order \(p\), then the local truncation error \(L_n(Y)\) can be expressed in terms of the constants defined in (3.3) as follows:

\[
L_n(Y) = h^{p+1} \sum_{\varepsilon = 0}^{p+1} C_{p+1, \varepsilon} \left( p+1 \right) \left( \frac{\partial}{\partial t} \right)^{p+1-\varepsilon} \left( \frac{\partial}{\partial s} \right)^{\varepsilon} Y(t,s)_{|t=s=t_n} + O(h^{p+2}) \quad \text{as} \quad h \to 0. \quad (3.4)
\]

It is of some interest now to compare the values of the error constants \(C_{p+1, \varepsilon}\), \(\varepsilon = 0(1)p+1\), for the various subclasses given in Section 2. We have evaluated and simplified the expressions for these constants as much as possible:

For the ILM method Corollary 3.1 gives \(p = k\), under the (reasonable) assumption that \(\tilde{p} \geq k\). We then find

\[
C_{p+1, \varepsilon} = (-1)^{p-1} \sum_{i=0}^{k} \left[ i^p \{i a_i + (p+1) \gamma_i\} - R \right], \quad (3.5)
\]

where \(R = k! \gamma_i\) if \(\varepsilon = 0\) and \(R = 0\) if \(\varepsilon = 1(1)p+1\).

For both the ML and MML method we find

\[
C_{p+1, \varepsilon} = \begin{cases} 0, & \text{if } \varepsilon = 0(1)p, \\ (-1)^{p-1} \sum_{i=0}^{k} i^p \{i a_i + (p+1) \gamma_i\}, & \text{if } \varepsilon = p+1. \end{cases} \quad (3.6)
\]

We have computed the numerical values of the error constants for two usual choices of the coefficients \(\{a_i, \gamma_i\}\), viz., the **backward differentiation (BD)** method, for which \(\tilde{p} = k\), and the **Adams-Moulton (AM)** method for which \(p = k+1\). Table 1 gives the values of the relevant constants \(C_{p+1, \varepsilon}\) where \(p\) is prescribed by Corollary 3.1. Note that the (M)ML-AM methods have \(p = k+1\), whereas the other methods have \(p = k\).
TABLE 1

Error constants in (3.4) for various choices of \(\{a_i, y_i\}\) in (2.5)

<table>
<thead>
<tr>
<th>Method ({a_i, y_i})</th>
<th>(k=1)</th>
<th>(k=2)</th>
<th>(k=3)</th>
<th>(k=4)</th>
<th>(k=5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILM { (C_{k+1,0})}</td>
<td>(-2)</td>
<td>(0)</td>
<td>(-72/11)</td>
<td>(0)</td>
<td>(-14400/137)</td>
</tr>
<tr>
<td>AM { (C_{k+1,0})}</td>
<td>(-1)</td>
<td>(2)</td>
<td>(-6)</td>
<td>(24)</td>
<td>(-120)</td>
</tr>
<tr>
<td>ML &amp; MML { (C_{k+1,0})}</td>
<td>(-1)</td>
<td>(2)</td>
<td>(-6)</td>
<td>(24)</td>
<td>(-120)</td>
</tr>
<tr>
<td>ILM { (C_{k+1,&gt;0})}</td>
<td>(-1)</td>
<td>(2)</td>
<td>(-6)</td>
<td>(24)</td>
<td>(-120)</td>
</tr>
<tr>
<td>AM { (C_{k+1,&gt;0})}</td>
<td>(-1)</td>
<td>(2)</td>
<td>(-6)</td>
<td>(24)</td>
<td>(-120)</td>
</tr>
</tbody>
</table>

The order of convergence of the LM method is dictated not only by its order of consistency, but also, of course, by the quadrature error (2.4) and by the errors in the starting values \(y_1, \ldots, y_{k-1}\). In the next Section we shall analyze the convergence of the LM method \((2.3)-(2.5)\).

4. CONVERGENCE

Similarly as with LM methods for ODEs, a necessary condition for convergence of the LM method \((2.3)-(2.5)\) is that the characteristic polynomial

\[
a(z) := \sum_{i=0}^{k} a_i z^{-i} \quad (4.1a)
\]

satisfies the root condition, i.e., its roots are on the unit disk, those on the unit circle being simple.

In the sufficient conditions for convergence the parameters \(\beta_{ij}\) and \(\gamma_{ij}\) are also involved. We define

\[
\beta(z) := \sum_{i=0}^{k} \beta_{i} z^{-i} \quad \beta_{i} := \sum_{j=-k}^{k} \beta_{ij}; \quad (4.1b)
\]

\[
\gamma(z) := \sum_{i=0}^{k} \gamma_{i} z^{-i} \quad \gamma_{i} := \sum_{j=-k}^{k} \gamma_{ij}. \quad (4.1c)
\]

Furthermore, we will use the notation

\[
\Delta K_n(t) = K(t, t_n, y(t_n)) - K(t, t_n, y_n),
\]

\[
\Delta E_n(h) = \max_{i \leq j \leq n, i \neq j} \left| E_i(h; t_j) - E_j(h; t_j) \right|,
\]
\[ E_n(h) = \max_{1 \leq j \leq n} \left| E_j(h, t_j) \right|, \]

\[ T_n(h) = \max_{i \leq n} \left| L_i(Y) \right| \quad \text{and} \quad \delta(h) = \max_{j \leq k-1} \left| y(t_j) - y_j \right|. \]

\( E_n(h) \) is the maximal error arising in the approximation of the lag terms \( Y(t_n, t) \) by \( Y_n(t) \) (cf. (2.4)). \( T_n(h) \) may be considered as the maximal local truncation error of the LM formula, and \( \delta(h) \) is the maximal starting error. We now formulate a general convergence theorem which provides an estimate for the global error

\[ \varepsilon_n = y(t_n) - Y_n. \]  

We assume that \( K \) satisfies the Lipschitz conditions

\[ |A(t) - A(t*)| \leq L_1 |t-t*| |\varepsilon(t)|, \]

\[ |A(t) - A(t*)| \leq L_2 |t-t*| |\varepsilon(t)|, \]

where \( L_1 \) and \( L_2 \) are the Lipschitz constants.

Theorem 4.1 Let \( \alpha(z) \) satisfy the root condition.

(i) If \( \alpha(z) = \alpha_0 z^k \) then there exists a constant \( C > 0 \) such that

\[ \left| \varepsilon_n \right| \leq C \left[ h^q \delta(h) + E_n(h) + \Delta E_n(h) + T_n(h) \right], \]

(4.3a)

(ii) If \( \beta(z) = 0 \) then there exists a constant \( C > 0 \) such that

\[ \left| \varepsilon_n \right| \leq C \left[ \delta(h) + h^{-1} \left( \Delta E_n(h) + T_n(h) \right) \right], \]

(4.3b)

Theorem 4.1 Let \( \delta(h) = O(h^q) \), \( E_n(h) = O(h^r) \) (as in (2.4)), \( \Delta E_n(h) = O(h^{r+1}) \) as \( h \to 0 \) and let \( \{a_i, \gamma_i\} \) in (2.5) be the coefficients of a \( p \)-th order consistent LM method for ODEs. Then the order of convergence \( p^* \) of the LM method for (1.1) is given by: \( p^* = \min(q+1,r) \) for the DQ method, \( p^* = \min(q,r,p) \) for the ILM method, \( p^* = \min(q+1,r,p+1) \) for the ML method and \( p^* = \min(q,r,p) \) for the MML method, where \( p \) is the order of consistency of (2.5), given by Corollary 3.1.

5. RELATION WITH LM METHODS FOR ODEs

The Volterra equation (1.1) contains the classes of ordinary differential equations as special cases. For example, if in (1.1) \( g(t) = \text{constant} \) then

\[ K(t, \tau, y) = f(\tau, y) \quad \Rightarrow \quad \frac{dy}{dt} = f(t, y) \]  

(5.1a)

\[ K(t, \tau, y) = (t-\tau) f(\tau, y) \quad \Rightarrow \quad \frac{d^2y}{dt^2} = f(t, y) \]  

(5.1b)
Therefore, it is natural to ask to what method the LM formula (2.5) reduces when it is applied to the special cases (5.1). Furthermore, one may ask for the relationship with LM methods for ODEs of the form (5.1). In order to formulate this relationship we introduce, in addition to the polynomials $a(z)$, $b(z)$ and $y(z)$ (cf. (4.1)), the polynomials

$$
\begin{align*}
\bar{b}(z) := \sum_{i=0}^{k} b_i z^{-i}, & \quad \bar{b}_i := \sum_{j=-i}^{k} b_{ij} \quad (5.2a) \\
\bar{y}(z) := \sum_{i=0}^{k} y_i z^{-i}, & \quad \bar{y}_i := \sum_{j=-i}^{k} y_{ij} \quad (5.2b)
\end{align*}
$$

We shall also employ the shift operator $E$ defined by $E y_n = y_{n+1}$.

**Theorem 5.1** Let $g(t) = \text{constant}$.

(i) If $K(t,\tau,\gamma) = f(\tau,\gamma)$ then the formula (2.5) reduces to

$$
\begin{align*}
a(E)y_n + b(E)y_n(t_n) &= hy(E)f(t_n, y_n), \quad n \geq 0. (5.3a) \\
\end{align*}
$$

(ii) If $K(t,\tau,\gamma) = (t-\tau)f(\tau,\gamma)$, $b(z) = 0$, and if the weights $w_{n,j}$ in (2.3) are $(\rho,\sigma)$-reducible, then the formula (2.5) reduces to

$$
\begin{align*}
a(E)\rho(E)y_n + h^2(\sigma(E)\bar{b}(E) - \rho(E)\bar{y}(E) & \quad (5.3b) \\
- k \rho(E)\gamma(E) + \rho(E)\gamma'(E)E]f(t_n, y_n) &= 0,
\end{align*}
$$

where $\gamma'$ denotes the derivative of $\gamma$.

From part (i) of this theorem it follows that the LM formula (2.5), when applied to the integrated form of the first order equation $dy/dt = f(t, y)$, reduces to a linear multistep method $\{a, y\}$ for this equation, provided that $b(z) = 0$. This statement holds, irrespective of the weights $w_{n,j}$ used in the definition of the lag term $Y(t)$. In other words, if the matrix $B$ is chosen such that the row sums vanish ($\bar{b}_i = 0$) then our linear method is in fact an LM method for ODEs whenever the Volterra equation (1.1) is a (first order) ODE. Such linear methods will be called $(a, y)$-reducible. The recurrence relation (5.3a) plays an important role in the stability analysis of Volterra equations with (5.1a) as test kernel. In particular, for $(a, y)$-reducible methods, the ODE-stability theory directly applies and may suggest suitable polynomials $a$ and $y$ for the construction of stable numerical methods for solving equation (1.1).

**Example 5.1** The ILM and the MML methods are $(a, y)$-reducible, whereas the DQ and the ML methods are not.

Part (ii) of Theorem 5.1 provides us with further information about how we should choose the weights $w_{n,j}$ and the matrices $A$, $B$ and $C$ in order to construct a suitable integration method. Observe, that here the structure of the matrices $B$ and $C$ is such,
that the same set of polynomials $\{\rho, \sigma, \alpha, \gamma\}$ may lead to different recurrence relations.

Example 5.2 Let both $\{a, \gamma\}$ and $\{\rho, \sigma\}$ be the trapezoidal rule, i.e., $a(z) = \rho(z) = z-1$ and $\gamma(z) = \sigma(z) = \frac{1}{2}(z+1)$. Now it is a simple calculation to find that (5.3b) reduces to a LM method $\{\rho^*, \sigma^*\}$ for second order ODEs with $\rho^*(z) = (z-1)^2$ both for the ILM and the MML method, but with $\sigma^*(z) = \frac{1}{2}(z+1)^2$ for the ILM and $\sigma^*(z) = z$ for the MML method, respectively. (Note that both methods have order 2 (cf. [7, p.253]), with error constant $-1/6$ for the ILM and $1/12$ for the MML method, respectively.)

For an extension of Theorem 5.1 to the case of a general convolution kernel $K(t, \tau, y) = \sum_{\chi=0}^{M} (t-\tau)^{\chi} f_\chi(\tau, y)$, the reader is referred to [6].

6. $V_0$-STABILITY

Definition 6.1 A discretization method for (1.1) is said to be $V_0$-stable if $y_n \to 0$ as $n \to \infty$ whenever it is applied, with fixed stepsize $h > 0$, to the test equation

$$y(t) = y_0 + \int_0^t (\lambda + \mu(t-\tau)) y(\tau) d\tau, \quad (6.1)$$

with arbitrary $(\lambda, \mu) \in Q_{\lambda, \mu} := \{ (\lambda, \mu): \lambda < 0, \mu \leq 0 \}$.

Wolkenfelt [10] has shown that the DQ method (2.7) can not be $V_0$-stable when the quadrature weights in (2.3) are $(\rho, \sigma)$-reducible. This negative result raised the question of whether $V_0$-stable methods for (1.1) do exist at all. Brunner, Mørsett and Wolkenfelt [4] answered this question affirmatively for a certain class of so-called one-stage implicit Runge-Kutta methods. In the class of LM methods analysed in the present paper, $V_0$-stable methods do also exist. In particular, they occur in the subclass of ILM methods. To see this, we observe that for the ILM methods we have

$$\beta(z) = -\gamma(z), \quad \gamma(z) = -k\gamma(z) + z\gamma'(z), \quad (6.2)$$

and from Theorem 5.1 we derive the following result.

Theorem 6.1 Let the conditions of Theorem 5.1 part (ii) be satisfied, then the LM method, when applied to the test equation (6.1), assumes the form

$$\{\rho(E)[a(E) - h\lambda y(E)] + h^2 \mu[\sigma(E)\beta(E) - \rho(E)(\gamma(E) + k\gamma(E) - E\gamma'(E))]\}y_n = 0. \quad (6.3)$$

In the ILM case this equation reduces to

$$\{\rho(E)[a(E) - h\lambda y(E)] - h^2 \mu\sigma(E)\gamma(E)\}y_n = 0. \quad (6.3')$$

Since equation (6.3') is identical to the one obtained by Brunner and Lambert ([3]) in their stability analysis of
numerical methods for the test integro-differential equation
\[ \frac{dy}{dt}(t) = \lambda y(t) + \mu \int_0^t y(\tau) d\tau, \quad (6.4) \]
we may find examples of \( V_0 \) -stable ILM methods just by inspecting the stability regions given by Brunner and Lambert. In this way, we immediately conclude from [3] that the four combinations, with \( \{a, \gamma\} \) and \( \{\alpha, \sigma\} \) defining either the trapezoidal rule or the backward Euler rule, are \( V_0 \) -stable methods. It turns out that the MML versions of these methods are not \( V_0 \) -stable. (In fact, as communicated to us by S. Amini, no MML methods can be \( V_0 \) -stable.) In Figure 2 the stability regions of both the IML and the MML methods are given. Evidently, the ILM methods have considerably larger regions of stability.

\[ \{\sigma, \sigma\} \rightarrow \text{trapezoidal} \quad \{\alpha, \gamma\} \rightarrow \text{backward Euler} \]

\[ \begin{array}{c}
(0,-4) \\
(2,4) \\
(2,0) \\
(0,-4) \\
(4,-4)
\end{array} \]

\[ \begin{array}{c}
(1,-2) \\
(2,0) \\
(2,0) \\
(0,-4) \\
(4,-4)
\end{array} \]

**FIG. 2.** Stability regions for MML(///) and ILM(\///) methods
7. NUMERICAL EXPERIMENTS

In this Section we illustrate by a few numerical experiments the convergence theorem 4.1 and the improved stability behaviour of the ILM and the MML methods. In the tables of results we list the accuracy obtained, by

\[ A(h) := - \log_{10}\left(\text{relative error at the end point}\right), \quad (7.1) \]

i.e., the number of correct digits in the numerical solution. The pair \((\rho,\sigma)\) used for computing the lag term will always define a Gregory formula of order \(r\); the pair \((a,\gamma)\) defines either an Adams-Moulton or a backward differentiation formula of order \(p\). Methods are denoted by, e.g., ILM(G\(_r\)-BD\(_p\)).

7.1 Order of convergence

In the first experiment we integrated the equation

\[ y(t) = 1 + \sin(t) - \cos(t) - \int_0^t y(\tau)d\tau, \quad 0 \leq t \leq 2. \quad (7.2) \]

The starting values were taken from the exact solution \(y(t) = \sin(t)\). The generating characteristic polynomials \((\rho,\sigma,a,\gamma)\) were chosen such that, according to Corollary 4.1, all methods listed in Table 2 are just of order \(p^* = 5\). In this Table the values of \(A(h)\) and the corresponding effective order \(p_{\text{eff}}\) are presented, where \(p_{\text{eff}} = [A(h) - A(2h)]/\log_{10}2\).

\[
\begin{array}{cccccc}
\text{Tests of order of convergence} \\
\hline
h^{-1} & DQ(G_3) & ILM(G_5-AM_6) & ML(G_5-AM_4) & MML(G_5-AM_3) \\
\hline
4 & 5.0 & 3.8 & 4.7 & 5.3 \\
& 6.7 & 6.0 & 6.5 & 6.1 \\
8 & 6.7 & 5.3 & 6.3 & 6.8 \\
& 6.4 & 6.2 & 6.3 & 6.1 \\
16 & 8.3 & 6.9 & 7.9 & 8.4 \\
& 6.2 & 6.1 & 6.2 & 6.1 \\
32 & 9.9 & 8.5 & 9.5 & 9.9 \\
\hline
\end{array}
\]

From the results we see that the effective order tends to the asymptotic order as \(h\) decreases. We also see that the ILM method is less accurate than the other methods, which may be explained by its larger error constants (cf. Table 1).

7.2 Stability

In the second experiment we chose an example in which the
kernel has a large Lipschitz constant (obtained by modifying an example given by Bownds[2]):

\[ y(t) = 50(1-t^2)ln(1+t) + 75t^2 - 5lt + l^{-100\int_0^t (1+t-\tau)y(\tau)d\tau}, \]

\[ 0 \leq t \leq 4. \]

Again, the starting values were taken from the exact solution \( y(t) = 1-t \). The results listed in Table 3 clearly show the better stability properties of the ILM method (a negative \( A(h) \)-value may be interpreted as an unstable behaviour). In particular, we observe the only marginally better performance of the MML methods when compared with the ML methods.

**TABLE 3**

<table>
<thead>
<tr>
<th>Stability tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h^{-1} )</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>16</td>
</tr>
</tbody>
</table>

**REFERENCES**


