Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

Parallel methods for nonstiff VIDEs

P.J. van der Houwen

Modelling, Analysis and Simulation (MAS)

MAS-R9811 July 31, 1998

Report MAS-R9811
ISSN 1386-3703

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Parallel Methods for Nonstiff VIDEs

Dedicated to Professor Phil Anselone at the occasion of his retirement

P.J. van der Houwen
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
We consider numerical methods for nonstiff initial-value problems for Volterra integro-differential equations. Such problems may be considered as initial-value problems for ordinary differential equations with expensive righthand side functions because each righthand side evaluation requires the application of a quadrature formula. The often considerable costs suggest the use of methods that require only one righthand side evaluation per step. One option is a conventional linear multistep method. However, if a parallel computer system is available, then one might also look for methods with more righthand sides per step, but such that they can all be evaluated in parallel. In this paper, we construct such parallel methods and we show that on parallel computers they are by far superior to the conventional linear multistep methods which do not have scope for parallelism. Moreover, the (real) stability interval is considerably larger.

1991 Mathematics Subject Classification: 65L06
Keywords and Phrases: Volterra integro-differential equations, general linear methods, parallelism.
Note: Work carried out under project MAS 1.4 - Exploratory research: Analysis of ODEs and PDEs.

1. Introduction

We consider explicit numerical methods for nonstiff initial-value problems (IVPs) for Volterra integro-differential equations (VIDEs) of the form

$$(1.1)\qquad \frac{d\mathbf{y}(t)}{dt} = \mathbf{f}\big(\mathbf{y}(t), \mathbf{q}(t)\big),\ \mathbf{q}(t) := \int_{t_0}^{t} \mathbf{k}\big(\mathbf{y}(t), \mathbf{y}(x)\big)\,dx, \qquad \mathbf{y}, \mathbf{f}, \mathbf{k} \in \mathbb{R}^d,\quad t_0 \le t \le t_{end}.$$

Such IVPs may be considered as IVPs for ordinary differential equations (ODEs) with expensive righthand sides (RHSs) because each RHS evaluation requires the evaluation of the integral term $\mathbf{q}(t)$. In the numerical solution of (1.1), the often considerable costs of the RHSs suggest the application of methods that use only one RHS per step, such as in the conventional linear multistep methods, or if a parallel computer system is available, methods of which all RHSs per step can be evaluated in parallel. This leads us to consider methods of the form

$$(1.2)\qquad \mathbf{Y}_{n+1} = (R\otimes I)\mathbf{Y}_n + h(S\otimes I)\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n),\ n \ge 1.$$

Here, R and S denote k-by-k matrices, h is the stepsize $t_{n+1} - t_n$, and $\otimes$ denotes the direct product between matrices (Kronecker product). Each of the k components $\mathbf{y}_{n,i}$ of the kd-dimensional solution vector $\mathbf{Y}_n$ represents a numerical approximation to $\mathbf{y}(t_{n-1} + a_i h)$ and each of the k components $\mathbf{q}_{n,i}$ of the kd-dimensional vector $\mathbf{Q}_n$ represents a quadrature formula for $\mathbf{q}(t_{n-1} + a_i h)$. The vector $\mathbf{a} := (a_i)$ is called the *abscissa vector*, $\mathbf{Y}_n$ is called the *stage vector* and its components $\mathbf{y}_{ni}$ the *stage values*. Furthermore, for any vector $\mathbf{Y}_n = (\mathbf{y}_{ni})$, $\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n)$ contains the RHS values $(\mathbf{f}(\mathbf{y}_{ni}, \mathbf{q}_{ni}))$. We shall always assume that $a_k = 1$.

Since the k components of $\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n)$ can be computed in parallel (provided that k processors are available), (1.2) requires only one *effective* righthand side evaluation per step (here, *effective* means that RHSs that *can* be evaluated in parallel *are* evaluated in parallel).

In the ODE case ($\mathbf{f}$ independent of $\mathbf{q}$), the method (1.2) belongs to the wide class of general linear methods (GLMs) introduced by Butcher in 1966 (see the text books [4] and [6] for a detailed analysis). Examples of such GLMs are (i) linear multistep methods with $\mathbf{a} = (i - k + 1)^T$ and a matrix S whose first k-1 rows vanish, or (ii) the multi-block methods of Chu and Hamilton [5] characterized by $\mathbf{a} = (i/k)^T$ and by (in principle) full matrices R and S. Multiblock methods with general (nonequidistant) abscissae have been considered in [7] as a special case of block Runge-Kutta methods, but specific methods were only given for k = 2.

In this paper, we want to derive methods of the type (1.2) for VIDEs, that is, we should equip the method with a quadrature method based on the $\mathbf{y}$-values available at the points $t_{ni} := t_{n-1} + a_i h$. We shall consider two options, viz. (i) quadrature formulas using all points $\{t_{ni}: n \geq 1,\ 1 \leq i \leq k\}$, so-called *extended* methods, and (ii) quadrature formulas only using the step points $\{t_n: n \geq 1\}$, so-called *mixed* methods. In the case of extended methods, it will be an advantage if the points t_{ni} are more or less equidistant. If the stage order of the GLM is sufficiently high, then this would make the quadrature formula considerably more accurate than the conventional linear multistep approach where only step points are available, or the explicit Runge-Kutta formulas where the off-step points cannot be used because of their low stage order. However, a disadvantage is the large storage requirement if many integration steps are involved. An alternative is the use of the storage economic mixed methods. Since here only the step point values are involved in the quadrature formula, we should try to choose the abscissae such that we have superconvergence at the step points. Given a sufficiently accurate quadrature formula, the methods constructed in this paper have stage order k, step point order k+1, and satisfactory large real stability boundaries.

2. Construction of methods

Given a procedure to compute the quadrature terms, suitable methods can be constructed by imposing consistency conditions on the arrays **a**, R and S. The consistency of (1.2) is defined by substitution of the exact solution into the GLM and by requiring that the residue vanishes as h tends to zero. The rate by which the residue tends to zero determines the *order of consistency*. We shall call the GLM (or the stage vector $\mathbf{Y}_{n+1}$) *consistent of order p* if the residue upon substitution of the exact solution values $\mathbf{y}(t_n + a_i h)$ into (1.2) is of order h^{p+1}. Assuming that the quadrature formulas are sufficiently accurate we find by expansion into Taylor series the consistency conditions

$$(R + zS)\exp(\mathbf{b}z) - \exp(\mathbf{a}z) = O(z^{p+1}), \quad \mathbf{b} := \mathbf{a} - \mathbf{e}, \quad \mathbf{e} := (1, \ldots, 1)^T,$$

where we used the componentwise notation of function of vectors, that is, for any vector $\mathbf{v} := (v_i)$, $\exp(\mathbf{v})$ denotes the vector with components $\exp(v_i)$. This leads to the equations

$$R\mathbf{e} = \mathbf{e}, \quad R\mathbf{b}^j + j\, S\mathbf{b}^{j-1} = \mathbf{a}^j, \quad j = 1, \ldots, p.$$

The error constants are given by the components of the vector

(2.1) $$\mathbf{C}(p+1) := R\mathbf{b}^{p+1} + (p+1)\, S\mathbf{b}^p - \mathbf{a}^{p+1}.$$

Let us introduce the k-by-p matrices U(p), V(p) and W(p):

(2.2) $$U(p) := (\mathbf{a}, \mathbf{a}^2, \ldots, \mathbf{a}^p), \quad V(p) := (\mathbf{b}, \mathbf{b}^2, \ldots, \mathbf{b}^p), \quad W(p) := (\mathbf{e}, 2\mathbf{b}, 3\mathbf{b}^2, \ldots, p\mathbf{b}^{p-1}).$$

The consistency conditions can now be expressed as

(2.3) $$R\mathbf{e} = \mathbf{e}, \quad RV(p) + SW(p) = U(p).$$

Given the abscissa vector **a**, the system (2.3) yields k(p+1) linear equations.
However, in order to have convergence, that is, $\mathbf{y}_{n+1,i} \to \mathbf{y}(t_n + a_i h)$ as $h \to 0$ for all grid points in the integration interval, the GLM should satisfy the *necessary* condition of *zero-stability*, that is, R has its eigenvalues on the unit disk and the eigenvalues of modulus one have multiplicity one. Therefore, in the construction of GLMs, one usually prescribes a (family of) zero-stable matrix R, satisfying the condition $R\mathbf{e} = \mathbf{e}$, and next the remaining order conditions are solved.

2.1. Generalized Adams-Bashforth methods

In this paper, we confine our considerations to the case where the matrix $R = \mathbf{e}\mathbf{e}_k^T$ with $\mathbf{e}_k$ denoting the kth unit vector. Evidently, this matrix is zero-stable. Substitution into (2.3) and setting p = k, and by virtue of our assumption $a_k = 1$, yields the matrix

(2.4) $$S = (U(k) - \mathbf{e}\mathbf{e}_k^T V(k))W^{-1}(k) = U(k)W^{-1}(k).$$

The resulting methods may be considered as generalizations of the classical Adams-Bashforth methods (AB methods), because just as in AB methods, each stage value is defined by the most recent **y**-vector available and the k already computed RHS values. Furthermore, as with the k-step AB methods, they have order p = k and possess a matrix R with one eigenvalue 1 and k-1 eigenvalues 0. The methods {(1.2),(2.4)} will be referred to as *Generalized Adams-Bashforth methods* (GAB methods).

In this paper, we shall choose one of the still free abscissae a_i such that the GLM contains an embedded formula for stepsize control. Suppose that $a_{k-1} = 2$. Then, $\mathbf{y}_{n+1,k} - \mathbf{y}_{n,k-1}$ provides an $O(h^{k+1})$ local error estimate. However, this estimate will be more effective if $\mathbf{y}_{n+1,k}$ is of higher order than $\mathbf{y}_{n,k-1}$. By virtue of the structure of R, this can be achieved by requiring that the kth component $C_k(k+1)$ of the error vector $\mathbf{C}(k+1)$ vanishes. This equation imposes a condition on the abscissae a_i. In order to derive a simple expression for this condition, we consider the equation in the system (1.2), viz.

$$\mathbf{y}_{n+1,k} = \mathbf{y}_{n,k} + h(\mathbf{e}_k^T S \otimes I)\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n), \tag{2.5}$$

and we compare this equation with the relation

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}\big(\mathbf{y}(t), \mathbf{q}(t)\big)\, dt. \tag{2.6}$$

Thus, $h(\mathbf{e}_k^T S \otimes I)\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n)$ may be considered as an interpolatory quadrature formula for the integral term in (2.6) using the quadrature points $t_{ni} := t_{n-1} + a_i h = t_n + b_i h$, where $b_i := a_i - 1$. Such quadrature formulas possess an approximation error of the form (see e.g. [1, p. 55])

$$\frac{1}{k!} \int_{t_n}^{t_{n+1}} \pi_k(t)\, \frac{d^k \mathbf{f}\big(\mathbf{y}(\theta(t)), \mathbf{q}(\theta(t))\big)}{dt^k}\, dt, \quad \pi_k(t) := \prod_{i=1}^{k} (t - t_{ni}),$$

where $\mathbf{f}\big(\mathbf{y}(t), \mathbf{q}(t)\big)$ is assumed k times continuously differentiable on $[t_n, t_{n+1}]$ and $\theta(t)$ assumes values in the interval (t_n, t_{n+1}). Hence, the polynomial order of accuracy can be raised by one if the integral of $\pi_k(t)$ over the interval $[t_n, t_{n+1}]$ vanishes, that is, if the shifted abscissae b_i satisfy the relation

$$\int_{t_n}^{t_{n+1}} \pi_k(t)\, dt = h^{k+1} \int_0^1 \prod_{i=1}^{k} (x - b_i)\, dx = 0. \tag{2.7}$$

Imposing this superconvergence condition yields a (k-2)-parameter family of GAB methods of order p = k+1. We remark that relation (2.7) can never be satisfied by abscissae a_i in the interval [0,1], i.e.

$-1 \le b_i \le 0$. This follows from the fact that if the quantities b_i would all be nonpositive, then $\pi_k(t)$ has no zeros in the interval $[t_n,t_{n+1}]$, so that the integral of $\pi_k(t)$ over the interval $[t_n,t_{n+1}]$ cannot vanish. If k = 2, then we obtain a uniquely defined third-order method with $\mathbf{a} = (5/3,1)^T$. In this case, the easy error estimate mentioned above is not possible. However, if k > 2, then we may set $a_{k-1} = 2$ to obtain the local error estimate $\mathbf{y}_{n+1,k} - \mathbf{y}_{n,k-1}$. Observing that for $a_k = 1$ and $a_{k-1} = 2$ the superconvergence condition (2.7) is can always be satisfied by choosing the free abscissae a_i symmetrically with respect to 3/2, we are led to define

(2.8)
$$\begin{aligned} &\text{k even:} \quad k = 2: a_k = 1, \quad a_{k-1} = \frac{5}{3}, \\ &\qquad\qquad k \ge 4: a_k = 1, \quad a_{k-1} = 2, a_i = 3 - a_{k-i-1}, \qquad i = 1, \ldots, k-2. \\ &\text{k odd:} \quad k \ge 3: a_k = 1, \quad a_{k-1} = 2, a_{k-2} = \frac{3}{2}, \\ &\qquad\qquad k \ge 5: a_i = 3 - a_{k-i-2}, \qquad\qquad i = 1, \ldots, k-3. \end{aligned}$$

Theorem 2.1. Let (2.8) be satisfied. Then, there exist an abscissae vector **a** such that:

(a) If $k \ge 2$, then the global order p = k+1.
(b) If $k \ge 3$, then the error estimate $\mathbf{y}_{n+1,k} - \mathbf{y}_{n,k-1}$ is of local order k+1. ♦

Condition (2.7) is always true for k odd. If k is even, then we have to spend one abscissae to satisfy (2.7). The still remaining free abscissae may be chosen, for example such that the first k-1 components of the error vector $\mathbf{C}(k+1)$ are of small magnitude, or such that the stability region is sufficiently large. Let us first try to reduce the magnitude of the error constants. Since we assumed $a_k = 1$, we may write $\mathbf{C}(k+1) = (k+1)S\mathbf{b}^k - \mathbf{a}^{k+1}$. We have minimized the error constants $C_i(k+1)$, $i \le k-1$ under the constraint that the norm of S does not increase too much. Thus, we expect that a suitable choice for the free abscissae a_i is obtained by minimizing the quantity

(2.9) $$G(\mathbf{a}) := \max_{\mathbf{a}} \left\{ \| (C_1(k+1), \ldots, C_{k-1}(k+1)) \|_\infty + \gamma \| S \|_\infty \right\},$$

where γ denotes some constant.

Table 2.1. Abscissa vectors, the norm G(**a**), and real stability boundaries.

k	$\mathbf{a}^T$ for GAB methods	G(**a**)	β_{real}(AB)	β_{real}(GAB)
2	$\left(\frac{5}{3}, 1\right)$	1.88	1.0	0.63
	3	$\left(\frac{3}{2}, 2, 1\right)$	4.07	0.53 0.48
4	$\left(\frac{1741}{1364}, \frac{2351}{1364}, 2, 1\right)$	6.34	0.30	0.44
5	$\left(\frac{1137}{1024}, \frac{1935}{1024}, \frac{3}{2}, 2, 1\right)$	10.26	0.16	0.42

6	$\left(\frac{2480}{2279}, \frac{2199}{1643}, \frac{2730}{1643}, \frac{4379}{2279}, 2, 1\right)$	20.59	0.08	0.42
7	$\left(\frac{865}{944}, \frac{571}{476}, \frac{857}{476}, \frac{1967}{944}, \frac{3}{2}, 2, 1\right)$	48.32	0.04	0.41

In Table 2.1, we have listed the GAB abscissa vectors obtained for $\gamma = 10^{-2}$ (this value was chosen experimentally), together with the corresponding (local) minimum value of $G(\mathbf{a})$ and the real stability boundaries for both the AB and GAB methods (see also Section 2.2 for a discussion of the overall stability of the VIDE method). It turns out that the abscissae in the 4-stage GAB method are numerically equal to the Lobatto abscissae in the interval [1, 2]. For larger values of k, there is no relation with the Lobatto points. The stability boundaries of the GAB methods are quite satisfactory, so that there is no reason to look for abscissa vectors which yield still larger boundaries.

2.2. The starting vector and overall stability

The GLM (1.2) needs the starting vector $\mathbf{Y}_1 \approx \mathbf{y}(t_0 + a_i h)$. If all absissae are positive, then this starting vector can be generated by a one-step method, e.g. a Runge-Kutta method. If one or more abscissae are negative, then we need starting values at points left to t_0. Since this is inconvenient in practice, we follow another approach which is based on the redefinition of the points t_n, $n \geq 1$. Let a_{min} denote the minimal absissa value and define for $n \geq 1$, $t_n := t_0 + (n - a_{min})h$ (instead of the original step points $t_n = t_0 + nh$). In particular, we have $t_1 = t_0 + (1 - a_{min})h$. Evidently, none of the points $t_{1i} := t_0 + (a_i - a_{min})h$ are located to the left of t_0. So by using a starting vector $\mathbf{Y}_1$ which approximates the exact solution at these points, i.e. $\mathbf{Y}_1 \approx \mathbf{y}(t_{1i})$, we do not anymore need starting values at points left to t_0. Now, let $\mathbf{y}_{RK}(t)$ denote a Runge-Kutta approximation at the point t. Then we may define the starting vector $\mathbf{Y}_1 := \left(\mathbf{y}_{RK}(t_0 + (a_i - a_{min})h)\right)^T$. In fact, we can also use this starting vector in the case of positive abscissae with the advantage that the starting value corresponding with a_{min} is exact. Thus, given the quantities $\{\mathbf{a}, h, R, S\}$, the starting procedure $\mathbf{Y}_1 = \left(\mathbf{y}_{RK}(t_0 + (a_i - a_{min})h)\right)^T$, and the quadrature formulas $\mathbf{q}_{ni}$, the method (1.2) completely defines the sequence of vectors $\mathbf{Y}_2, \mathbf{Y}_3, \mathbf{Y}_4, \ldots$.

Next we briefly discuss the overall stability of the VIDE method. Evidently, the overall stability is influenced by the stability of the quadrature formula. A stable way of defining quadrature rules converts the integral term into a differential equation and integrates this differential equation by a sufficiently stable ODE solver. For that purpose, we introduce the function

$$\mathbf{z}(t,s) := \int_{t_0}^{s} \mathbf{k}(\mathbf{y}(t), \mathbf{y}(x))\, dx. \tag{2.10}$$

By observing that $\mathbf{q}(t) = \mathbf{z}(t,t)$, we see that we can apply the GLM (1.2) to the ODE (1.1), where the values of $\mathbf{q}(t)$ needed by the GLM are obtained by integrating the initial-value problem

$$\frac{\partial \mathbf{z}(t,s)}{\partial s} = \mathbf{k}(\mathbf{y}(t), \mathbf{y}(s)), \quad \mathbf{z}(t,t_0) = 0 \tag{2.11}$$

from $s = t_0$ until $s = t$.

The underlying integrator should be sufficiently stable because the righthand side in (2.8) is affected by the numerical errors due to the GLM integrator. One option is to apply the same GLM (1.2) as used for integrating (1.1), to obtain an *(R,S)-reducible* method for VIDEs. If the GLM (1.2) is sufficiently stable, then we may also expect overall stability. To be more precise, we should consider the complete integration process, that is, the recursions

$$\begin{aligned} &\mathbf{Y}_{n+1} = (R\otimes I)\mathbf{Y}_n + h(S\otimes I)\mathbf{F}(\mathbf{Y}_n, \mathbf{Q}_n), \quad n \geq 1. \\ &\mathbf{Z}_{n,\nu+1} = (R\otimes I)\mathbf{Z}_{n,\nu} + h(S\otimes I)\mathbf{K}(\mathbf{Y}_n,\mathbf{Y}_\nu), \quad \nu = 0, 1, \ldots, n-1, \\ &\mathbf{Q}_n = \mathbf{Z}_{n,n}, \end{aligned} \tag{2.12}$$

where $\mathbf{K}(\mathbf{Y}_n,\mathbf{Y}_\nu)$ contains the kernel values $\left(\mathbf{k}(\mathbf{y}_{ni}, \mathbf{y}_{\nu i})\right)$. The linear stability of VIDE methods is usual studied by means of the linear test equation (cf. [3])

$$\frac{dy(t)}{dt} = \xi y(t) + \eta \int_{t_0}^{t} y(x)\ dx. \tag{2.13}$$

By writing this equation as a system of the two ODEs

$$\frac{dy(t)}{dt} = \xi y(t) + \eta z(t), \quad \frac{dz(t)}{dt} = y(t), \tag{2.13'}$$

one first shows that separately applying the GLM $\{\mathbf{a}, R, S\}$ to *each* of these two equations is equivalent with applying it directly to the *system* (2.13'). Then, the following result is easily proved (cf. [1, p. 470], [8]).

Theorem 2.2. Let $\mathbb{S}$ be the linear stability region of the GLM (1.2), defined by set of points z where $R + zS$ has its eigenvalues on the unit disk, and let λ and μ be defined by $\lambda+\mu = \xi$, $\lambda\mu = -\eta$. Then, with respect to the linear test equation (2.13), the set $\{(h\xi,h^2\eta): h\lambda\in\mathbb{S},\ h\mu\in\mathbb{S}\}$ defines the region of stability of the (R,S)-reducible GLM {(1.2), (2.12)}. ♦

If this theorem is applied to the case where ξ and η are real (which is relevant in the case of scalar VIDEs) and if the GLM (1.2) has a real stability boundary β_{real}, then the (R,S)-reducible GLM has the stability region (see Table 2.1 for the values of β_{real} corresponding to the AB and GAB methods)

$$-2\beta_{real} \leq h\xi \leq 0, \quad -\beta_{real}^2 \leq h^2\eta.$$

Remark 2.1. Equation (2.11) can of course be integrated by any GLM $\{\mathbf{a}, R^*, S^*\}$ with the same abscissae vector $\mathbf{a}$. This would lead to the recursion

$$\begin{pmatrix} \mathbf{Y}_{n+1} \\ h\mathbf{Z}_{n+1} \end{pmatrix} = M \begin{pmatrix} \mathbf{Y}_n \\ h\mathbf{Z}_n \end{pmatrix}, \quad M := \begin{pmatrix} R + h\xi S & \eta S \\ h^2 S^* & R^* \end{pmatrix}.$$

The stability region is now defined by the set $\{(h\xi,h^2\eta): |\lambda(M)| \le 1\}$, where the eigenvalues $\lambda(M)$ of the amplification matrix M are determined by its characteristic equation

$$\det\left(R + h\xi S - \lambda I\right) \det\left(R^* - \lambda I - h^2\eta S^*(R + h\xi S - \lambda I)^{-1}S\right) = 0. \ \blacklozenge$$

An advantage of the quadrature procedure (2.12) is that the high stage order of all stage values in $\mathbf{Y}_n$ and $\mathbf{Y}_\nu$ can be fully exploited (this is not the case if the underlying method (1.2) is replaced by a Runge-Kutta method). However, as already remarked in the introduction, a disadvantage of these *extended* methods is the large storage requirement if many integration steps are involved.
An alternative is the use of *mixed* methods in which the quadrature formula is only based on the set of step points $\{t_\nu: \nu = 0, \ldots, n\}$ and the right end point t of the integration interval. Let the numerical approximation to $\mathbf{z}(t,t_\nu)$ be denoted by $\mathbf{z}_\nu$, let $\mathbf{k}_\nu := \mathbf{k}(\mathbf{y}_{n,i}, \mathbf{y}_\nu)$, and let the quadrature formula be of the linear m-step form, that is, $\mathbf{z}_\nu$ is defined as a linear combination of values $\mathbf{z}_{\nu-1}, \ldots, \mathbf{z}_{\nu-m}$ and $\mathbf{k}_\nu, \ldots, \mathbf{k}_{\nu-m}$. Then, by observing that the **k**-values indirectly depend on the **z**-values, we should at least require that the linear m-step formula is stable. For example, let (2.11) be integrated by the classical fourth-order Runge-Kutta method whose intermediate points coincide with the step points t_n. Then, this method is equivalent with the linear two-step method

$$\mathbf{z}_{\nu+1} = \mathbf{z}_{\nu-1} + 2h\left(\tfrac{1}{6}\,\mathbf{k}_{\nu-1} + \tfrac{2}{3}\,\mathbf{k}_\nu + \tfrac{1}{6}\,\mathbf{k}_{\nu+1}\right).$$

This method is easily recognized as the Simpson method which has a zero real stability boundary. Thus, although the underlying Runge-Kutta method has a nonzero real stability boundary for ODEs of the form $\mathbf{z}' = \mathbf{k}(\mathbf{z})$, it does *not* have a nonzero real stability boundary in the present situation, because the Runge-Kutta method has changed from a one-step method to a multistep method. However, applying a multistep method to (2.11) leads to quadrature formulas that are equivalent with the same multistep method. Hence, if these multistep methods are sufficiently stable (e.g. Adams-Moulton methods), then the resulting quadrature method is also sufficiently stable.

3. Numerical comparisons

In order to isolate algorithmic properties from implementation properties, all methods were run with fixed stepsizes. The accuracy was measured by the number of correct significant digits csd := $-\log_{10}$(relative maximum error at the end point) and the computational effort by the total, *effective* number of RHS evaluations N, that is, N refers to those RHS evaluations that *have* to be done sequentially. Since the main computational cost of the whole algorithm consists of the evaluation of RHS functions and since the computation of the RHSs is quite costly, the communication costs will be negligible, so that N furnishes an estimate for the effective computational costs.
The VIDE algorithm consists of two main numerical procedures, viz. an ODE solver and a quadrature procedure. For the quadrature procedure we took the 2-step, 3rd-order Adams-Moulton method (AM

method) only based on step point values, using the trapezoidal rule to obtain the necessary starting values. For the ODE solver we took k-stage GAB methods and classical k-stage (i.e. k-step) Adams Bashforth methods, respectively denoted by GABk and ABk. Furthermore, in order to compare with ODE methods requiring more than one effective RHS per step, we also applied the classical 4th-order Runge-Kutta method (RK method). The methods were run on the following test problems:

$$(3.1)\qquad \frac{dy}{dt} = \frac{1}{y}\ln\left(\frac{1+t}{1+t/2}\right) - t - \frac{1}{(1+t)^2} + \int_{t_0}^{t} \frac{1}{1+(1+t)y(x)}\,dx, \quad y(0) = 1,\ 0 \le t \le 1,$$

$$(3.2)\qquad \frac{dy}{dt} = -\exp(y(t)^3) + \int_{t_0}^{t} y(x)\,dx, \qquad y(0) = 1,\ 0 \le t \le 1,$$

$$(3.3)\qquad \frac{dy}{dt} = -\exp(y(t)^8) + \int_{t_0}^{t} \sin(y(t)y(x))\,dx, \qquad y(0) = 1,\ 0 \le t \le 1.$$

The first test problem is the often used example of Brunner and Lambert [3]. The second example is more difficult because of the highly nonlinear ODE part. The third example is the most difficult problem with increased nonlinearity (note also that the kernel depends on both y(t) and y(x)).

Table 3.1 lists values of N and csd for k = 3, ... , 7. These figures show that for a given number of stages the GAB methods are always considerably more accurate than the AB methods. In fact, in most cases, the GAB methods produce about the same accuracy for 25% of the number of effective RHS evaluations (this implies that on a sequential computer, the GAB4-AM method is about as efficient as the AB4-AM method). Furthermore, it seems to pay to use ODE methods of higher order than the quadrature formula. As to the performance of the RK method, we see that the RK-AM results are more or less comparable with the ABk-AM results for $k \ge 4$. Finally, we remark that the GAB methods allow us to use extended quadrature formulas based on all available stage values, which will again improve the accuracy when compared with the AB-AM and RK-AM methods.

Table 3.1. Correct number of significant digits at the end point using 3rd-order AM quadrature.

Problem	N	AB3	GAB3	AB4	GAB4	AB5	GAB5	AB6	GAB6	AB7	GAB7	RK
(3.1)	10	2.9	5.2	3.7	5.5	3.3	5.4	3.7	5.1	3.4	5.8	3.8
	20	4.0	6.7	4.7	6.7	3.8	7.0	4.4	6.4	4.2	6.9	4.4
	40	4.5	8.6	5.8	7.9	5.6	8.5	5.4	7.6	5.2	7.8	5.5
	80	5.5	8.8	7.1	9.0	6.7	9.0	6.5	8.5	6.2	8.8	6.7
	160	6.2	9.6	6.7	10.0	6.1	9.8	7.5	9.4	7.4	9.7	7.5
(3.2)	40	1.8	3.9	2.1	4.0	4.1	4.2	2.2	4.1	2.2	3.8	2.4
	80	2.6	5.5	3.1	4.8	3.9	4.9	3.1	4.8	3.0	4.7	3.4
	160	3.4	6.1	4.3	5.7	4.3	5.7	4.1	5.6	3.9	5.6	4.1
	320	4.3	6.7	5.3	6.6	5.7	6.6	5.2	6.5	6.3	6.5	4.9
(3.3)	40	0.4	2.2	0.8	2.5	0.9	2.9	1.0	3.9	1.3	2.9	0.6
	80	1.2	3.1	1.5	3.6	1.8	4.2	1.9	4.4	2.3	4.0	1.7
	160	2.0	4.2	2.5	4.7	2.9	5.4	3.0	5.1	3.9	4.9	2.8

320	2.8	5.3	3.5	5.8	4.2	6.0	4.2	5.9	4.4	5.8	3.9

4. Concluding remarks

In this paper, we constructed explicit k-stage GLMs with step point order k+1 and stage order k such that all RHSs per step can be evaluated in parallel. Application to VIDEs with fixed stepsizes and a 3rd-order Adams-Moulton quadrature formula only based on step points showed a theoretical speed-up by a factor about 4 with respect to Adams-Bashforth methods. These quite promising results motivate future research in the following directions:

(i) providing the methods with an extended quadrature procedure based on all available stage values and with automatic stepsize control based on the embedded local error estimate $\mathbf{y}_{n+1,k} - \mathbf{y}_{n,k-1}$ (see Theorem 2.1).
(ii) extension to parallel VIDE methods for stiff IVPs (in preparation, see [2]).
(iii) implementation on parallel computer systems.

These topics will be subject of future research.

References

[1] Brunner, H. & Houwen, P.J. van der [1986]: The numerical solution of Volterra equations, CWI Monograph 3, North-Holland, Amsterdam.

[2] Brunner, H. & Houwen, P.J. van der [1998]: Parallel methods for solving Runge-Kutta discretizations of Volterra integro-differential equations, in preparation.

[3] Brunner, H. & Lambert, J. D. [1974]: Stability of numerical methods of Volterra integro-differential equations, Computing 12, 75-89.

[4] Butcher, J.C. [1987]: The numerical analysis of ordinary differential equations, Runge-Kutta and general linear methods, Wiley, New York.

[5] Chu, M.T. & Hamilton, H. [1987]: Parallel solution of ODE's by multi-block methods, SIAM J. Sci. Stat. Comput. 3, 342-53.

[6] Hairer, E., Nørsett, S.P. & Wanner, G. [1987]: Solving ordinary differential equations, Vol. I. Nonstiff problems, Springer-Verlag, Berlin.

[7] Houwen, P.J. van der & Sommeijer, B.P. [1991]: Block Runge-Kutta methods on parallel computers, ZAMM 72, 3-18.

[8] Sommeijer, B.P., Couzy, W. & van der Houwen, P.J. [1992]: A-stable parallel block methods for ordinary and integro-differential equations, Appl. Numer.Math. 9, 267-281.