http://doc.rero.ch

# THE LINEAR BARYCENTRIC RATIONAL QUADRATURE METHOD FOR VOLTERRA INTEGRAL EQUATIONS[*]

J.-P. BERRUT[†], S. A. HOSSEINI[‡], AND G. KLEIN[§]

**Abstract.** We introduce two direct quadrature methods based on linear rational interpolation for solving general Volterra integral equations of the second kind. The first, deduced by a direct application of linear barycentric rational quadrature given in former work, is shown to converge at the same rate as the rational quadrature rule but is costly on long integration intervals. The second, based on a composite version of this quadrature rule, loses one order of convergence but is much cheaper. Both require only a sample of the involved functions at equispaced nodes and yield an infinitely smooth solution of most classical examples with machine precision.

**Key words.** Volterra integral equations, direct quadrature method, linear barycentric rational interpolation, linear rational quadrature

**AMS subject classifications.** 65R20, 45G10

**1. Introduction: Volterra integral equations and the direct quadrature method for their solution.** The modeling of certain physical phenomena with history leads to Volterra integral equations of the second kind,

$$(1.1) \qquad y(t) = f(t) + \int_a^t K\big(t, s, y(s)\big)\, ds,$$

for the unknown function $y(t)$ given on some interval $I := [a, T]$. The functions $f$ and $K$ are assumed here to be known on a discrete subset of $I$ and on $\Omega := I \times I \times (-\infty, \infty)$, respectively. In practice, we shall use $f$ and $K$ (except for its third variable) only at equispaced values of the variables.

Classical theorems, among them the following one, on the existence and uniqueness of the solution of (1.1) can be found in [8, 21, 24].

THEOREM 1.1. *Let $f$ be continuous on $I$, let $K$ be continuous in $\Omega$, and assume that $K$ satisfies a Lipschitz condition with respect to its third variable. Then (1.1) possesses a unique continuous solution $y$. Moreover, this solution is $r$ times continuously differentiable on $I$ whenever $f$ and $K$ are $r$ times continuously differentiable on $I$ and $\Omega$, respectively.*

Most Volterra integral equations must be solved by numerical means. A whole panoply of methods exists to that end. In some cases, such equations may be beautifully and very efficiently solved globally with spectral methods such as that pre-

sented in [9]. The convergence of such methods is very rapid, usually exponential for equations involving analytic $f$ and $K$; moreover, the order adapts automatically to the degree of smoothness of these data. As generalizations of ordinary differential equations, however, Volterra integral equations constitute one-directional evolution problems: the variable $t$ usually is time, and the solution at some value $t^*$ only depends on the past. Global methods, therefore, are not among the widely used: they are barely mentioned in books on ordinary differential equations, such as [13], or in books on Volterra integral equations, such as [8, 21]. Their main drawbacks are that the functions in the equation must be sampled at points which cluster at the boundary and that a large dense system of (nonlinear) equations must be solved to obtain the approximate solution. The first is not very natural for a time variable, hinders the efficient handling of nonboundary steep gradients with a step control procedure, and requires a start from scratch when one wishes to solve the problem further than the time limit $T$ fixed at the onset: real-time applications seem hardly possible. The second brings about the curse of dimensionality, in particular when solving a system of integral equations.

The quadrature method is the simplest for solving Volterra integral equations. With it, one seeks an approximation of $y$ only at a finite number of values $t_m$ of $t$, which we shall call *nodes*:

$$y_m \approx y(t_m).$$

We shall take these nodes equispaced: $t_m := a + mh$, $h := (T - a)/N$, $m = 0, \ldots, N$ (this is not at all necessary, but natural in a first approach); together they yield a partition $T_N$ of $[a, T]$.

Starting from the given value $y_0 := y(a) = f(a)$, the integral in (1.1) is replaced by a quadrature rule of nonzero weights $hw_k^{(m)}$ and $y_m$ is defined as the solution of the equations

$$(1.2) \qquad y_m = f(t_m) + h \sum_{k=0}^{m} w_k^{(m)} K(t_m, t_k, y_k), \qquad m = 1, \ldots, N.$$

This way, the method merely requires a sample at equispaced values of the arguments of $f$ and of $K$ (with respect to its first two variables, e.g., with linear equations). Exponential convergence is then out of reach [25]. Moreover, the method has another drawback: at step number $m$, the mere $m$ values $y_0, \ldots, y_{m-1}$ are known, so that the quadrature rule cannot contain more than $m + 1$ nodes. For the first (small) values of $m$, the accuracy of the rule (its order) will be very low, and this lack of precision will be carried through the whole integration. To avoid this, one considers integrals involving at least a certain number of subintervals, say, $n$; then the upper limit in the sum in (1.2) becomes $\max\{m, n\}$. As a consequence, the first $n$ equations in (1.2) all contain the unknowns $y_1, \ldots, y_n$ and must be solved simultaneously, even in the linear case. This is called the *starting procedure* of the direct quadrature method and requires knowledge of $K$ on a discrete subset of the square $[a, t_n] \times [a, t_n]$ (for the first two variables); the corresponding $y_m$ are the *starting values*.

To study the convergence of the method, [21] introduces some notation; we shall reproduce the most relevant to the present work. Let $\mathcal{C}$ be a class of equations of the form (1.1). The *(global discretization) error* in $t_m$ is defined as

$$e_m := y(t_m) - y_m, \qquad m = 0, \ldots, N.$$

DEFINITION 1.2. *A direct quadrature method* (1.2) *is* convergent *if, for all equations in $\mathcal{C}$, one has for every $t \in [a, T]$ fixed with $t = t_m$, for some $m$, for every suitable $h$,*

$$\lim_{\substack{h \to 0 \\ t_m = t}} y_m = y(t).$$

*It is* convergent of order $p$ *if*

$$|e_m| \leq Mh^p$$

*for some constant $M > 0$, and if $p$ is maximal with this property.*

DEFINITION 1.3. *Let $y$ be the solution of* (1.1). *The function*

$$\delta(h, t_m) := \int_a^{t_m} K\big(t_m, s, y(s)\big)\, \mathrm{d}s - h \sum_{k=0}^{m} w_k^{(m)} K\big(t_m, t_k, y(t_k)\big)$$

*is called the* local consistency error *for* (1.1) *at $t_m$.*

DEFINITION 1.4. *If for every equation in $\mathcal{C}$*

$$\lim_{h \to 0} \max_{0 \leq m \leq N} |\delta(h, t_m)| = 0,$$

*the approximation method* (1.2) *is said to be* consistent *with* (1.1) *for the class of equations $\mathcal{C}$. If for every equation in $\mathcal{C}$ there exists a constant $c$ (independent of $h$, but usually dependent on $f$ and $K$) such that*

$$\max_{0 \leq m \leq N} |\delta(h, t_m)| \leq ch^p,$$

*then the method is said to be* consistent of order $p$ *in $\mathcal{C}$.*

Sufficient conditions for the convergence of the direct quadrature method are given in the following theorem, whose proof is presented in [21, p. 102]; the discussion following the theorem in that reference implies the last statement.

THEOREM 1.5. *Consider the approximate solution of* (1.1) *by* (1.2)—*taking $\max\{m, n\}$ as upper limit of the sum, $n \in \mathbb{N}$ fixed—and assume that $\mathcal{C}$ is the class of equations* (1.1) *whose kernel $K$ satisfies a Lipschitz condition in the third variable, that the solution $y$ and the kernel $K$ are such that the approximation method is consistent of order $p$ with* (1.1), *that the weights satisfy*

(1.3) $$\sup_{k, m} |w_k^{(m)}| \leq W < \infty \quad \text{for some constant } W,$$

*and that the starting errors $e_k$, $k = 0, \ldots, n$, tend to zero as $h \to 0$. Then the method* (1.2) *is convergent; moreover, if the starting errors tend to 0 with order $p-1$, i.e., if*

$$|e_m| \leq c_m h^{p-1}, \quad m = 0, \ldots, n,$$

*then the order of convergence is at least $p$.*

In order to attain relative machine precision, a direct method (1.2) requires quadrature rules with large $m$. In view of their limited convergence and their instability, Newton–Cotes formulae cannot be used. (They do not satisfy (1.3) either.) One therefore resorts to composite rules [21], with the aesthetic drawback that the

3

same one (except the trapezoid) cannot be used throughout the interval, since a rule is required for every $m$.

In contrast, we may take advantage of the accuracy and stability of the linear barycentric quadrature scheme introduced in [20] to apply it to Volterra integral equations. In section 2, we briefly review some facts on linear barycentric rational interpolation and the aforementioned quadrature scheme. In the following section, we describe a global method for the solution of Volterra equations and prove its convergence. This method is very accurate but slow; we therefore present a more efficient alternative using composite quadrature rules in section 4. For the sake of comparison, we combine in section 5 our linear rational starting procedure with Gregory quadrature into a hybrid method. We illustrate our theoretical results with some numerical examples.

**2. Barycentric rational interpolation and quadrature.** Quadrature methods are usually based on (linear) interpolants. Here we recall the ideas that have led to the introduction of linear barycentric rational quadrature in [20]. Let $t_0, \ldots, t_m$ be $m + 1$ real abscissae and $g_0, \ldots, g_m$ corresponding values. (We denote by $g$ the function to be interpolated or integrated, to avoid confusion with $f$, the forcing function in the Volterra integral equation.) A barycentric rational interpolant to these data will here be an expression of the form

$$(2.1) \qquad r_m(t) = \sum_{j=0}^{m} \frac{\beta_j}{t - t_j} g_j \bigg/ \sum_{j=0}^{m} \frac{\beta_j}{t - t_j}$$

with all *barycentric weights* $\beta_j \neq 0$, so that all points are indeed considered in the interpolant. We shall denote a vector of $\beta_j$ by $\beta$. One sees that $r_m$ is a rational function with numerator and denominator both of degree at most $m$ by multiplying these by $\prod_{k=0}^{m}(t - t_k)$. We refer the reader to [6] for a review on polynomial barycentric interpolation and to [5] for its rational extension. An important advantage of the barycentric representation is that it guarantees interpolation independently of the $\beta_j$. In [4], the first author has introduced pole-free *linear* barycentric *rational* interpolants, in which the $\beta_j$ depend only on the $t_j$, not on the $g_j$, making $r_m$ linear in the latter; see also [2, 5]. Then every choice $\beta = (\beta_0, \ldots, \beta_m)^T$ usually defines another linear space of interpolants.

In [10], Floater and Hormann have given weights $\beta$ that, in principle, provide an arbitrary high rate of convergence. For equispaced nodes, their results may be summarized as follows: to interpolate a function $g \in C^{d+2}[t_0, t_m]$, $1 \leq d \leq m$, with a rate of convergence $\|g - r_m\|_\infty = \mathcal{O}(h^{d+1})$, the weights are simply to be chosen proportional to

$$(2.2) \qquad \beta_j = \frac{(-1)^{j-d}}{2^d} \sum_{i \in J_j} \binom{d}{j-i}, \quad J_j := \big\{ i \in \{0, \ldots, m-d\} : \ j - d \leq i \leq j \big\};$$

as mentioned by the authors, these $\beta_j$ differ from 1 only in the vicinity of the extremities, and the values different from 1 should be considered as endpoint corrections to the interpolant with all weights equal ($d = 0$) given in [4]. As mentioned in [10], the rational interpolants with the weights in (2.2) can be written in a form different from (2.1), namely,

$$(2.3) \qquad r_m(t) = \frac{\sum_{j=0}^{m-d} \lambda_j(t) p_j(t)}{\sum_{j=0}^{m-d} \lambda_j(t)} \quad \text{with} \quad \lambda_j(t) = \frac{(-1)^j}{(t - t_j) \ldots (t - t_{j+d})}$$

4

and where $p_j$ is the unique polynomial of degree at most $d$ that interpolates the values $g_j, \ldots, g_{j+d}$ at the $d + 1$ consecutive nodes $t_j < \cdots < t_{j+d}$. We shall call this the *original form* of the barycentric interpolant with the Floater–Hormann weights.

In [20], two quadrature methods based on rational interpolants (2.1) have been introduced, called direct, respectively, indirect, rational quadrature. We concentrate here on the direct version; it simply consists in numerically integrating (2.1), which is a rational function and thus cannot be integrated analytically without additional knowledge of its properties, e.g., its poles. The idea in [20] is to take advantage of the excellent stability properties of the barycentric form (2.1) at any point to integrate $r_m$ with arbitrary precision by means of an efficient nonequispaced quadrature rule such as Gauss–Legendre or Clenshaw–Curtis.

To obtain the quadrature along this path, we rewrite (2.1) as

$$(2.4) \qquad r_m(t) = \sum_{k=0}^{m} g_k \ell_k^{(\beta)}(t), \qquad \ell_k^{(\beta)}(t) := \frac{\beta_k}{t - t_k} \Bigg/ \sum_{j=0}^{m} \frac{\beta_j}{t - t_j},$$

where the superscript $(\beta)$ refers to the particular choice of the barycentric weights $\beta_0, \ldots, \beta_m$, and we define the corresponding quadrature weights as

$$(2.5) \qquad \omega_k^{(m)} := \int_{t_0}^{t_m} \ell_k^{(\beta)}(t)\, \mathrm{d}t,$$

which leads to the quadrature formula:

$$(2.6) \qquad Q = \sum_{k=0}^{m} \omega_k^{(m)} g_k.$$

The stability of the quadrature rule can be assessed via the sum of the absolute values of the quadrature weights (2.5), since this sum is a bound on the amplification of the imprecision in the data. A crude but sufficient bound for our setting involves the Lebesgue constant $\Lambda_m$ and the Lebesgue function $L_m(t)$ associated with linear rational interpolation,

$$(2.7) \qquad \Lambda_m = \max_{t_0 \le t \le t_m} L_m(t) = \max_{t_0 \le t \le t_m} \sum_{j=0}^{m} \left| \frac{\beta_j}{t - t_j} \right| \Bigg/ \left| \sum_{j=0}^{m} \frac{\beta_j}{t - t_j} \right|.$$

It thus follows that

$$(2.8) \qquad \sum_{k=0}^{m} |\omega_k^{(m)}| \le \int_{t_0}^{t_m} \sum_{k=0}^{m} \left| \frac{\frac{\beta_k}{t - t_k}}{\sum_{j=0}^{m} \frac{\beta_j}{t - t_j}} \right| \mathrm{d}t = \int_{t_0}^{t_m} L_m(t)\, \mathrm{d}t \le (T - a)\Lambda_m.$$

If all the quadrature weights are positive then the left-hand side is equal to $(T - a)$. Otherwise, the Lebesgue constant associated with the barycentric interpolation scheme gives an estimation of the stability of the quadrature rule.

In practice we use the Floater–Hormann weights (2.2); notice that they change with $m$. The integrals (2.5) are then computed numerically, for instance, by means of routines implemented in the Chebfun system [3, 27, 29] or, alternatively, with Gauss–Legendre (see [14] for a fast implementation) or Clenshaw–Curtis rules (see, e.g., [28]), to obtain approximate weights $\omega_k^{(m)\mathcal{D}}$ to the best possible accuracy; see the appendix

for a possible implementation in MATLAB. From (2.8), it follows that this quadrature rule with equispaced nodes is extremely stable as long as $d$ is not taken too large, since the upper bound on the Lebesgue constant associated with these interpolants contains a logarithmic factor of $m$ and a factor $2^d$; see [7].

The convergence of the resulting integration scheme is given by the following theorem, derived from [20, Thm. 6.1] and [12].

THEOREM 2.1. *Suppose $m$ and $d$, $d \leq m/2 - 1$, are positive integers, $g \in C^{d+3}[t_0, t_m]$, and $r_m$ is the rational interpolant (2.4) with equispaced nodes $t_j$ and $\beta$ from (2.2). Then, for any $\ell = 0, \ldots, m$,*

$$\left| \int_{t_0}^{t_\ell} \big(g(t) - r_m(t)\big)\, \mathrm{d}t \right| \leq Ch^{d+2},$$

*where $C$ is a constant depending only on $d$, on derivatives of $g$, and on the length $t_\ell - t_0$ of the interval.*

In all our numerical tests we have chosen a sufficiently high precision for the computation of the integrals, so that the discretization error essentially stems from the rational approximation. For that reason we refrain from mentioning the superscript $\mathcal{D}$ in what follows; it should nevertheless be kept in mind.

In practice, we will always take a fixed value of $d$. This avoids Runge's phenomenon; see [10].

A result similar to Theorem 2.1 also holds in the case when the number of nodes $m$ is constant and $h$ shrinks. It will arise in the starting procedure, see section 3, and also in the composite method, see section 4, where each integral involves a fixed number $n + 1$ of nodes. If $g \in C^{d+2}[t_0, t_m]$, one obviously has for $\ell = 0, \ldots, m$,

$$(2.9) \qquad \left| \int_{t_0}^{t_\ell} \big(g(t) - r_m(t)\big)\, \mathrm{d}t \right| \leq \ell h \max_{t_0 \leq t \leq t_\ell} |g(t) - r_m(t)| \leq Ch^{d+2},$$

where $C$ again is a constant that only depends on $d$, on derivatives of $g$, and on the length of the interval $[t_0, t_m]$, but not on the number of nodes involved. (We will denote such generic constants throughout the paper by $C$.) If $m - d$ is odd, the bound on the interpolation error, as given in [10, Thm. 2], involves an additional factor $mh$, so that one can expect an increase to $d + 3$ of the order of the related quadrature rule, which is not the case for $m - d$ even.

To close this section, we present a bound on the weights in the above quadrature method; it will be used to prove the corresponding Theorem 1.5 for the rational quadrature method.

PROPOSITION 2.2. *Suppose $m$ and $d$, $d \leq m/2 - 1$, are positive integers and let the nodes be equispaced. Then the quadrature weights (2.5) with the barycentric weights from (2.2) satisfy*

$$(2.10) \qquad \left| \omega_k^{(m)} \right| \leq Ch, \qquad k = 0, \ldots, m,$$

*where the constant $C$ does not depend on $m$.*

*Proof.* We prove (2.10) for $0 \leq k \leq (m+1)/2$, since the quadrature rule is symmetric; see [20, Thm. 6.3]. By the definition of the weights from (2.2),

$$(2.11) \qquad \omega_k^{(m)} = \int_{t_0}^{t_m} \frac{\frac{\beta_k}{t - t_k}}{\sum_{j=0}^{m} \frac{\beta_j}{t - t_j}}\, \mathrm{d}t = \frac{2^d \beta_k}{h^d d!} \int_{t_0}^{t_m} \frac{\frac{1}{t - t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)}\, \mathrm{d}t,$$

where we switched from the barycentric to the original form (2.3); see [10, sect. 4] for the explicit relation between both expressions involving the $\lambda_j$ and the nonsimplified barycentric weights. To prove the result, we need to show that the integral in (2.11) behaves like $h^{d+1}$. We begin by looking at the two subintervals $[t_{k-1}, t_k]$ and $[t_k, t_{k+1}]$, provided they both exist:

$$\int_{t_{k-1}}^{t_k} \frac{\frac{1}{t-t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)} \, \mathrm{d}t = -\int_{t_{k-1}}^{t_k} \frac{t - t_{k-1}}{(t - t_{k-1})(t_k - t) \sum_{j=0}^{m-d} \lambda_j(t)} \, \mathrm{d}t.$$

The denominator in the integrand of the above right-hand side can be bounded as

$$(t - t_{k-1})(t_k - t) \left| \sum_{j=0}^{m-d} \lambda_j(t) \right| \geq \frac{1}{d! h^{d-1}};$$

see the proof of Theorem 1 in [7]. With the standard estimate of an integral, we obtain

$$\left| \int_{t_{k-1}}^{t_k} \frac{\frac{1}{t-t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)} \, \mathrm{d}t \right| \leq d! h^{d-1} \int_{t_{k-1}}^{t_k} (t - t_{k-1}) \, \mathrm{d}t = d! h^{d+1}/2,$$

and similarly for $[t_k, t_{k+1}]$, so that we may ignore these subintervals in what follows. In a next step, we split the integral over the whole interval $[t_0, t_m]$ into three major parts, namely, $[t_0, t_{d+1}]$, $[t_{d+1}, t_{m-d-1}]$, and $[t_{m-d-1}, t_m]$, and treat for each part the two cases when $t_k$ lies inside and outside. For the first major part, let $k \geq d + 2$; then

$$\int_{t_0}^{t_{d+1}} \frac{\frac{1}{t-t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)} \, \mathrm{d}t = \frac{1}{\sum_{j=0}^{m-d} \lambda_j(\xi)} \int_{t_0}^{t_{d+1}} \frac{\mathrm{d}t}{t - t_k} = \frac{-1}{\sum_{j=0}^{m-d} \lambda_j(\xi)} \log\left(\frac{k}{k - d - 1}\right)$$

for some $\xi \in [t_0, t_{d+1}]$, which follows from the mean value theorem for integrals, since $t - t_k$ does not change its sign in the interval under consideration. The logarithmic term is bounded from above by $\log(d + 2)$ and the absolute value of the reciprocal of the sum of the $\lambda_j$ by $d! h^{d+1}$ (see [10]), so that

$$\left| \int_{t_0}^{t_{d+1}} \frac{\frac{1}{t-t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)} \, \mathrm{d}t \right| \leq d! h^{d+1} \log(d + 2).$$

If $k \in [0, d+1]$, then we also split this first part into three smaller ones, provided they arise, namely, $[t_0, t_{k-1}]$, $[t_{k-1}, t_{k+1}]$, and $[t_{k+1}, t_{d+1}]$. The middle part has already been dealt with and the two others can be treated in a very similar way as above, so that the absolute value of the integrals can be shown to be bounded from above by $d! h^{d+1} \log(d + 2)$ as well.

The third major part of the interval does not contain $t_k$ by assumption. The absolute value of the integral over that part can again be shown to be bounded by $d! h^{d+1} \log(d + 2)$ in a similar fashion as above.

Only the middle major part remains to be treated. We reuse the following notation introduced in [20]:

$$\Omega_m(t) := \int_{t_{d+1}}^{t} \frac{\mathrm{d}s}{\sum_{j=0}^{m-d} \lambda_j(s)}, \qquad t \in [t_{d+1}, t_{m-d-1}].$$

7

As before, we begin with the case $k \leq d$ and now apply integration by parts:

$$\int_{t_{d+1}}^{t_{m-d-1}} \frac{\frac{1}{t-t_k}}{\sum_{j=0}^{m-d} \lambda_j(t)} \, dt = \frac{\Omega_m(t_{m-d-1})}{t_{m-d-1} - t_k} + \int_{t_{d+1}}^{t_{m-d-1}} \frac{\Omega_m(t)}{(t-t_k)^2} \, dt.$$

With the hypothesis on $d$, the assumption on $k$ made here, and similar arguments as in the proof of Theorem 6.1 in [20], the absolute value of the first term on the right-hand side may be bounded from above by $Ch^{d+1}$ with a constant $C$ that does not depend on $m$. For the second term, we observe that $(t-t_k)^2$ is positive, so that we may again apply the mean value theorem for integrals and, after a few basic computations, bound the absolute value of that term by $|\Omega_m(\xi)|/h$ for some $\xi \in [t_{d+1}, t_{m-d-1}]$. It was shown in [12] that $|\Omega_m(\xi)|$ is bounded by $Ch^{d+2}$. Therefore, this second major part is also bounded by $Ch^{d+1}$ if $k \notin [t_{d+1}, t_{m-d-1}]$. Otherwise, we split it into three others, as we did with the first major part, and apply partial integration and similar arguments as for the previous case for $k$, to draw the same conclusion. From (2.11) and the bounds on the three major parts of the integral in that equation, the claimed bound on $|\omega_k^{(m)}|$ follows. □

**3. The global method.** Let $T_N = \{a = t_0, \ldots, t_N = T\}$ be a uniform partition of the given interval $I$ with the step length $h = \frac{T-a}{N}$, $N \in \mathbb{N}$, as in section 1. Applying the linear barycentric rational quadrature formula (2.6) to the integral part of (1.1) yields for $m = n+1, \ldots, N$,

$$(3.1) \qquad y_m = f(t_m) + \sum_{k=0}^{m} \omega_k^{(m)} K(t_m, t_k, y_k), \quad \text{with} \quad \omega_k^{(m)} = \int_{t_0}^{t_m} \frac{\frac{\beta_k}{s-t_k}}{\sum_{j=0}^{m} \frac{\beta_j}{s-t_j}} \, ds,$$

for the approximation $y_m$ of the exact value $y(t_m)$. To comply with (1.2), we define here $w_k^{(m)} := \omega_k^{(m)}/h$.

As mentioned in the introduction, a starting procedure must be devised to guarantee the adequate accuracy of the first approximate values $y_0, \ldots, y_n$. We obtain $y_m$, $m \leq n$, by approximating the integral from $t_0$ to $t_m$ on its turn with a barycentric rational quadrature formula (2.6) with the $n+1$ mesh points $t_0, \ldots, t_n$. The parameter $d$ is thus bounded by $n$, see (2.9), which is not a restriction, since $n$ can be chosen accordingly to the choice of $d$, and $d$ is usually not taken larger than about 20. This yields the system of equations

$$y_0 = f(t_0),$$

$$(3.2) \qquad y_m = f(t_m) + h \sum_{k=0}^{n} w_k^{(m)} K(t_m, t_k, y_k), \qquad m = 1, \ldots, n,$$

for $y_1, \ldots, y_n$, where the quadrature weights for the starting procedure must be taken as

$$(3.3) \qquad \omega_k^{(m)} := \int_{t_0}^{t_m} \frac{\frac{\beta_k}{s-t_k}}{\sum_{j=0}^{n} \frac{\beta_j}{s-t_j}} \, ds, \qquad k = 0, \ldots, n,$$

and the $\beta_j$ depend on $n$, not on $m$.

We now study the convergence of this rational global method for Volterra integral equations of the second kind. We begin with the convergence of the starting values.

THEOREM 3.1. *Let* $\mathbf{y}_n = (y_0, \ldots, y_n)^T$ *be the approximate starting values, given by the system* (3.2) *with the errors* $\mathbf{e}_n = (e_0, \ldots, e_n)^T$, *and assume that* $f \in C^{d+2}(I)$ *and* $K \in C^{d+2}(\Omega)$ *in* (1.1) *with* $d \leq n$. *Then, as* $h \to 0$, *the maximum error* $\|\mathbf{e}_n\|_\infty \to 0$ *as* $\mathcal{O}(h^{d+2})$.

*Proof.* Theorem 1.1 with the assumptions on $f$ and $K$ imply that $y \in C^{d+2}(I)$. Applying the linear barycentric rational quadrature formula (2.6) to the integral part of (1.1) yields at $t = t_m$

$$
y(t_m) = f(t_m) + \int_a^{t_m} K\big(t_m, s, y(s)\big)\, \mathrm{d}s
$$

$$
(3.4) \qquad = f(t_m) + h \sum_{k=0}^n w_k^{(m)} K\big(t_m, t_k, y(t_k)\big) + R_m, \quad m = 1, \ldots, n,
$$

where $R_m$ is the quadrature error, $R_0 = 0$, and $R_m = \mathcal{O}(h^{d+2})$, which follows from (2.9) with Theorem 1.1 and Theorem 2 of [10]. Subtracting (3.2) from (3.4) gives

$$
e_m = h \sum_{k=0}^n w_k^{(m)} \big[ K\big(t_m, t_k, y(t_k)\big) - K(t_m, t_k, y_k) \big] + R_m
$$

$$
(3.5) \qquad = h \sum_{k=0}^n w_k^{(m)} \frac{\partial}{\partial y} K(t_m, t_k, \xi_k) e_k + R_m, \qquad m = 1, \ldots, n,
$$

where we used the mean value theorem with $\xi_k$ between $y(t_k)$ and $y_k$. To write this in vector form, we introduce the matrix $\mathbf{W}_n := (w_{m,k})_{k,m=0}^n$, whose entries are given by $w_{m,k} = w_k^{(m)} \frac{\partial}{\partial y} K(t_m, t_k, \xi_k)$. Then (3.5) may be written as

$$
(3.6) \qquad (\mathbf{I}_n - h\mathbf{W}_n)\mathbf{e}_n = \mathbf{R}_n,
$$

where $\mathbf{R}_n := (R_0, \ldots, R_n)^T$ and $\mathbf{I}_n$ is the $(n+1) \times (n+1)$ identity matrix. Note that we take the exact solution at the left end of the interval, so that $e_0 = 0$.

We now show that the matrix $h\mathbf{W}_n$ may be made smaller than 1 in norm by choosing $h$ small enough. To this end, we rewrite the quadrature weights (3.3) with the change of variable $v = (s - a)/(mh)$ as

$$
(3.7) \qquad \omega_k^{(m)} = mh \int_0^1 \frac{\frac{\beta_k}{vm - k}}{\sum_{j=0}^n \frac{\beta_j}{vm - j}}\, \mathrm{d}v.
$$

Notice that in the starting procedure, $n$ is fixed and only $h$ is variable. Thus the $(n+1)^2$ weights $w_k^{(m)} = \omega_k^{(m)}/h$ have a maximum absolute value, and also the $\frac{\partial}{\partial y} K(t_m, t_k, \xi_k)$ because of the differentiability hypothesis on $K$, so that there is a constant $V_1$ such that $\|\mathbf{W}_n\|_\infty \leq V_1 n$. Since $n$ is fixed, $h\|\mathbf{W}_n\|_\infty$ may be made as small as necessary by diminishing $h$. With $h$ small enough and by a well-known result (e.g., [1, Thm. 7.11]), $\mathbf{I}_n - h\mathbf{W}_n$ in (3.6) is nonsingular, so that

$$
\mathbf{e}_n = (\mathbf{I}_n - h\mathbf{W}_n)^{-1} \mathbf{R}_n,
$$

and there is a constant $V_2$ such that

$$
\|(\mathbf{I}_n - h\mathbf{W}_n)^{-1}\|_\infty \leq \frac{1}{1 - h\|\mathbf{W}_n\|_\infty} \leq V_2.
$$

Therefore $\|\mathbf{e}_n\|_\infty \leq V_2 \|\mathbf{R}_n\|_\infty = \mathcal{O}(h^{d+2})$. $\quad\square$

Our main theorem now follows from Theorems 1.5, 2.1, and 3.1 and Proposition 2.2. We observe that, in order for the consistency condition in Theorem 1.5 to hold, Theorem 2.1 requires $f \in C^{d+3}(I)$ and $K \in C^{d+3}(\Omega)$; see Theorem 1.1. Notice that an argument similar to the discussion accompanying (2.9) also holds for the computation of the first few $y_m$ with $m > n$, so that $d \leq n + 1$ is the only restriction on $d$ for the claimed order of convergence to hold.

THEOREM 3.2. *Let $n$ and $d$, $d \leq n + 1$, be positive integers, let the nodes be equispaced, and $f \in C^{d+3}(I)$ and $K \in C^{d+3}(\Omega)$ in (1.1). Consider the approximate solution of (1.1) by (3.1) and (3.2). If the parameter in the computation of the starting values is at least $d - 1$ and the consecutive approximations are computed with parameter $d$, then the method is convergent of order $d + 2$.*

We can construct an infinitely smooth approximate solution on the whole interval $[a, T]$ by interpolating the discrete approximations $y_0, \ldots, y_N$ with a linear barycentric rational interpolant (2.1) and the weights from (2.2). Under the hypotheses of Theorem 3.2 and with the choice $d + 1$ of the parameter in that interpolant, the absolute maximum error of that global approximation $r_N$ will behave as

$$\|y - r_N\|_\infty \leq \max_{a \leq t \leq T} \left| y(t) - \frac{\sum_{j=0}^N \frac{\beta_j}{t - t_j} y(t_j)}{\sum_{j=0}^N \frac{\beta_j}{t - t_j}} \right|$$

$$+ \max_{a \leq t \leq T} \left| \frac{\sum_{j=0}^N \frac{\beta_j}{t - t_j} y(t_j)}{\sum_{j=0}^N \frac{\beta_j}{t - t_j}} - \frac{\sum_{j=0}^N \frac{\beta_j}{t - t_j} y_j}{\sum_{j=0}^N \frac{\beta_j}{t - t_j}} \right|$$

$$\leq C h^{d+2} + \Lambda_N \max_{0 \leq j \leq N} |y(t_j) - y_j|$$

$$(3.8) \qquad \leq C \log(N) h^{d+2},$$

where $\Lambda_N$ is the Lebesgue constant associated with $r_N$ (see (2.7)), which grow logarithmically with $N$, and $C$ again is a generic constant that does not depend on $N$. The order of convergence of the rational interpolant of the exact values of $y$ to $y$ is guaranteed by the hypotheses of Theorem 3.2 together with Theorem 1.1.

**4. A composite method.** As mentioned in section 3, the global method requires computing $m + 1$ new quadrature weights for every $y_m$. Since the number $N$ of integrals increases like $1/h$ as $h$ decreases, the computational cost of these weights grows with $N$. To remedy this, we present a cheaper method, based on composite linear barycentric rational quadrature, to obtain an approximate solution of (1.1) with a simpler structure while retaining good approximation properties.

Let $T_N$ again be a uniform partition of the interval $I$. To define our composite rational quadrature rule over an interval $[a, t_m]$ with $m \leq N$, we choose a value of the parameter $d$, a fixed number $n$ of nodes with $d \leq n \leq m/2$ for the local quadrature rule, and we set $p := \lfloor m/n \rfloor - 1$. It can easily be seen from (3.7) that the quadrature weights $w_k^{(m)}$ are scale and translation invariant. We may therefore approximate a definite integral with the following composite rule:

$$\int_{t_0}^{t_m} g(t) \, dt = \sum_{j=0}^{p-1} \int_{t_{jn}}^{t_{(j+1)n}} g(t) \, dt + \int_{t_{pn}}^{t_m} g(t) \, dt$$

$$(4.1) \qquad \approx h \sum_{j=0}^{p-1} \sum_{k=0}^n w_k^{(n)} g(t_{jn+k}) + h \sum_{k=0}^{m-pn} w_k^{(m-pn)} g(t_{pn+k}).$$

The last sum is longer, in fact, up to twice as long, than the others to guarantee similar approximation and stability properties over the whole interval. From this construction and since each local quadrature formula has an error $\mathcal{O}(h^{d+2})$, if $g \in C^{d+2}[t_0, t_m]$, see (2.9), $n$ is fixed, and there are $p + 1 = \mathcal{O}(h^{-1})$ integrals to be computed; this composite rule behaves as follows.

PROPOSITION 4.1. *Suppose $m$, $n$, $n \leq m$, and $d$, $d \leq n$, are positive integers, $g \in C^{d+2}[t_0, t_m]$. Then the absolute error in the approximation of the integral of $g$ with the composite quadrature rule (4.1) and equispaced nodes is bounded by $Ch^{d+1}$, where the constant $C$ depends only on $d$, on derivatives of $g$, and on the interval length $t_m - a$.*

Moreover, if $n - d$ is odd, one can expect the order of convergence to be up to one unit larger than stated above, for the reasons mentioned just after (2.9).

The composite method for the solution of the Volterra integral equation of the second kind (1.1) is now quite simple. The starting procedure described in section 3 remains unchanged, so that the starting values are again given by the solution of the system (3.2). Thereafter, for the computation of $y_m$ for $n < m < 2n$, we continue in the same way as with the global method from section 3. Only for $m \geq 2n$ do we start applying the composite quadrature rule to the integral part of (1.1) as in (4.1), with the same $n$ as for the starting procedure, so that
(4.2)

$$y_m = f(t_m) + h \sum_{j=0}^{p-1} \sum_{k=0}^{n} w_k^{(n)} K(t_m, t_{jn+k}, y_{jn+k}) + h \sum_{k=0}^{m-pn} w_k^{(m-pn)} K(t_m, t_{pn+k}, y_{pn+k}).$$

The composite method is very similar to the global one, except that the order of consistency is typically one less than that of the global method under the same hypotheses. Consequently, the following result can be proved with the same ingredients as Theorem 3.2 and the help of Proposition 4.1.

THEOREM 4.2. *Let $n$ and $d$, $d \leq n$, be positive integers, let the nodes be equispaced, and let $f \in C^{d+2}(I)$ and $K \in C^{d+2}(\Omega)$ in (1.1). Consider the approximate solution of (1.1) by (3.2) and the composite method (4.2). If the parameter in the computation of the starting values is at least $d-1$ and if the subsequent approximations are computed with parameter $d$, then the method is convergent of order $d + 1$.*

Again, the order might be up to one unit larger if $n - d$ is odd and $n \geq 2(d+1)$. Here also, an infinitely smooth approximate solution can be obtained by interpolating the discrete approximations $y_0, \ldots, y_N$ by a linear barycentric rational interpolant with the parameter $d$ if $n - d$ is even, and $d + 1$ otherwise, provided $f$ and $K$ are sufficiently smooth. The reason for this choice of parameter can easily be deduced from (3.8).

**5. Hybrid methods.** A natural question about the methods presented in sections 3 and 4 is how they compare with some existing ones. Among those requiring only a sample of values of $f$ and $K$ at equispaced nodes, Gregory quadrature is well suited for such a comparison. However, it does not yield a starting procedure using only such values: more sophisticated methods are usually used, such as Runge–Kutta ones [18]. The quality of the results with our linear rational methods leads us to suggest hybrid methods, consisting of our rational starting procedure combined with any other efficient quadrature method for equispaced nodes. The resulting values can then again be interpolated with a linear rational interpolant.

The convergence of such a method simply follows from weak assumptions on the quadrature rule used after the starting procedure.

PROPOSITION 5.1. *Let $Q$ be the quadrature rule that defines the hybrid method, with bounded quadrature weights. Let $n$ and $d$, $d \leq n$, be positive integers, let the nodes be equispaced, let $f \in C^{d+1}(I)$ and $K \in C^{d+1}(\Omega)$ in (1.1), and let $f$ and $K$ fulfill the conditions for $Q$ to have convergence order $d + 2$. Consider the approximate solution of (1.1) by (3.2) and the quadrature rule $Q$. If the parameter in the computation of the starting values is at least $d - 1$, then the hybrid method is convergent of order $d + 2$.*

The hybrid method that we will use in the following section is based on Gregory quadrature; see [15] for an explicit construction and [22, 23] for elegant error bounds. To every order of convergence $D$ there corresponds a Gregory rule. According to Proposition 5.1 and [22, 23], for the corresponding hybrid method to converge with order $d + 2$, it is sufficient that $f \in C^{d+2}(I)$ and $K \in C^{d+2}(\Omega)$. The quadrature weights are bounded for any such order $D$, since the quadrature weights are all equal to 1 except those $2D - 2$ corresponding to the nodes near the extremities of the interval, if the number of nodes is larger than $2D - 2$. This method is now quite efficient; the weights can be computed with next to no effort, the order can be made fairly large, and the regularity conditions on $f$ and $K$ are relatively low. Methods based on Gregory rules have a long tradition; see, e.g., [11, 26].

**6. Numerical experiments.** In this section we present some numerical results of our global and composite methods, with various values of $d$, $n$, and $N$, applied to linear and nonlinear Volterra integral equations, to demonstrate the efficiency and accuracy of the schemes. We add the results from the hybrid method with the Gregory quadrature rule and the same parameters. The weights in the Gregory quadrature rule were computed with a short code given in course notes by Bengt Fornberg. Our main goal in displaying the results with the hybrid method is to provide a well-studied method of reference—except for the starting procedure—to situate the performance of the rational method among existing ones. For a given choice of the parameter $d$, the starting values are computed with parameter $d-1$, since this is sufficient according to the hypotheses of Theorems 3.2 and 4.2. Notice that the regularity of $f$ and $K$ only gives an upper bound on $d$, in addition to the one from the number of nodes, and that $d$ can be chosen much smaller if necessary, e.g., to guarantee stability. The interpolation of the discrete approximation of the solution is carried out with parameter $d+1$ except for the composite method when $n - d$ is even. For every example, the approximation quality is revealed via $e_S$, the maximal absolute error of the approximation of the $n+1$ starting values, and $e_N$, the error at $t = t_N$, the extremity of the whole interval. To convey an impression of the approximation throughout the interval, we also display in the tables the maximal absolute error $e_I$ of the interpolation of the approximations $y_m$, $m = 0, \ldots, N$, evaluated at about 3000 equispaced points of the interval. In the rows labeled $W$, we give $\|\mathbf{W}_n\|_\infty$ from the proof of Theorem 3.1, which is sufficiently small in all our examples, so that the order of convergence predicted by Theorem 3.1 for the staring values can actually be observed in practice. Moreover, we display the respective experimental approximation orders, $O_S$ for the starting procedure, $O_N$ for the approximation of $y(t_N)$, and $O_I$ for the interpolation. We took advantage of Chebfun for the computation of the quadrature weights; see the appendix for the corresponding code.

As a first example, we consider the integral equation

$$(6.1) \qquad y(t) = f(t) + \int_{-1}^{t} \frac{1 + 25t^2}{1 + 25s^2} y(s) \, \mathrm{d}s, \quad t \in [-1, 1],$$

TABLE 6.1
*Numerical results for example (6.1) with $d = 3$ and $n = 4$.*

| $N$ | | 10 | 20 | 40 | 80 | 160 | 320 | 640 |
|---|---|---|---|---|---|---|---|---|
| Starting procedure | $e_S$ | 1.8e−02 | 6.4e−05 | 1.6e−06 | 7.3e−08 | 4.0e−09 | 2.3e−10 | 1.4e−11 |
| | $O_S$ | | 8.1 | 5.3 | 4.5 | 4.2 | 4.1 | 4.0 |
| | $W$ | 2.2 | 2.5 | 3.2 | 3.6 | 3.8 | 3.9 | 4.0 |
| Global method | $e_N$ | 1.4e−00 | 3.4e−02 | 6.8e−05 | 1.4e−08 | 3.5e−10 | 1.0e−11 | 3.1e−13 |
| | $O_N$ | | 5.4 | 9.0 | 12.3 | 5.3 | 5.1 | 5.0 |
| | $e_I$ | 1.4e−00 | 3.4e−02 | 1.5e−04 | 4.5e−06 | 1.4e−07 | 4.5e−09 | 1.4e−10 |
| | $O_I$ | | 5.4 | 7.8 | 5.1 | 5.0 | 5.0 | 5.0 |
| Composite method | $e_N$ | 2.1e−00 | 2.8e−01 | 9.6e−03 | 4.0e−05 | 9.9e−10 | 3.8e−11 | 1.2e−12 |
| | $O_N$ | | 2.9 | 4.9 | 7.9 | 15.3 | 4.7 | 4.9 |
| | $e_I$ | 2.1e−00 | 2.8e−01 | 9.6e−03 | 4.0e−05 | 3.9e−07 | 4.5e−08 | 4.7e−10 |
| | $O_I$ | | 2.9 | 4.9 | 7.9 | 6.7 | 4.8 | 5.0 |
| Hybrid method | $e_N$ | 5.2e−01 | 2.0e−03 | 1.3e−07 | 5.0e−09 | 2.4e−10 | 8.3e−12 | 2.7e−13 |
| | $O_N$ | | 8.0 | 13.9 | 4.7 | 4.4 | 4.8 | 4.9 |
| | $e_I$ | 5.3e−01 | 4.6e−03 | 2.2e−04 | 7.9e−06 | 2.5e−07 | 8.0e−09 | 2.5e−10 |
| | $O_I$ | | 6.8 | 4.6 | 4.8 | 4.9 | 5.0 | 5.0 |

where

$$f(t) = \frac{1}{1 + 25t^2} - \frac{t}{2} - \left(1 + 25t^2\right)\left(\frac{1}{10}\arctan(5t) + \frac{1}{10}\arctan(5) + \frac{1}{52}\right).$$

The solution is Runge's function $y(t) = 1/(1 + 25t^2)$. Table 6.1 shows the numerical results with $d = 3$ and $n = 4$. Since $n - d$ is odd, we took the parameter $d + 1$ for the interpolation of the values obtained with our methods. The errors decrease with $N$ until about machine precision, as predicted by the theoretical results, and there is no Runge phenomenon, which does not occur with this kind of rational interpolants and fixed $d$ (see [10]), nor with the Gregory rule. The starting values are approximated with parameter $d_S = 2$, and the observed order is 4, which is $d_S + 2$, as one should expect from Theorem 3.1. The errors by the global method decrease with order 5, as well as their interpolation with $d = 4$, as predicted by Theorem 3.2 and the remark following it. The composite method and the interpolation with $d = 4$ of the discrete solution behave similarly since $n - d$ is odd (see Theorem 4.2), but with slightly larger errors. The errors with the hybrid method are similar to those with the rational methods for large $N$ and even smaller with low values of $N$; the order is about 5 as given by Theorem 5.1 applied to this particular method. The computation time with the composite and hybrid methods is significantly shorter than with the global one, though. With this example the absolute error in the $y_m$ is maximal in the middle of the interval, where Runge's function has its maximal curvature.

Next we study the classical test example from [8, Tab. 8.2.3a],

$$(6.2) \qquad y(t) = \frac{1}{2}t^2 e^{-t} + \frac{1}{2}\int_0^t (t - s)^2 e^{s-t} y(s)\,\mathrm{d}s, \quad t \in [0, 6],$$

with the exact solution $y(t) = \frac{1}{3}(1 - e^{-\frac{3}{2}t}(\cos(\frac{\sqrt{3}}{2}t) + \sqrt{3}\sin(\frac{\sqrt{3}}{2}t)))$. We display the results from our methods with $d = 3$ and $n = 7$ in Table 6.2. Again, they match the theoretical expectations very well. This time, $n - d$ is even and therefore the order with the composite method is only 4. Our methods give similar results as in [8] and remain stable even with large numbers of nodes.

TABLE 6.2
*Numerical results for example (6.2) with d = 3 and n = 7.*

| $N$ | | 10 | 20 | 40 | 80 | 160 | 320 | 640 |
|---|---|---|---|---|---|---|---|---|
| Starting procedure | $e_S$ | 2.0e−01 | 1.4e−03 | 3.4e−05 | 1.1e−06 | 3.9e−08 | 1.4e−09 | 4.6e−11 |
| | $O_S$ | | 7.1 | 5.4 | 5.0 | 4.8 | 4.8 | 4.9 |
| | $W$ | 1.4e+01 | 1.2e−00 | 6.0e−01 | 2.2e−01 | 6.6e−02 | 1.8e−02 | 4.8e−03 |
| Global method | $e_N$ | 6.6e−02 | 4.2e−04 | 9.5e−06 | 2.5e−07 | 6.6e−09 | 1.8e−10 | 5.3e−12 |
| | $O_N$ | | 7.3 | 5.5 | 5.3 | 5.2 | 5.2 | 5.1 |
| | $e_I$ | 2.4e−01 | 2.0e−03 | 6.2e−05 | 2.4e−06 | 8.8e−08 | 3.2e−09 | 1.1e−10 |
| | $O_I$ | | 6.9 | 5.0 | 4.7 | 4.7 | 4.8 | 4.8 |
| Composite method | $e_N$ | 6.6e−02 | 4.3e−04 | 9.3e−06 | 4.7e−07 | 2.9e−08 | 1.9e−09 | 1.3e−10 |
| | $O_N$ | | 7.3 | 5.2 | 4.6 | 3.8 | 4.0 | 3.8 |
| | $e_I$ | 2.2e−01 | 2.1e−03 | 8.7e−05 | 4.6e−06 | 2.6e−07 | 1.5e−08 | 9.1e−10 |
| | $O_I$ | | 6.7 | 4.6 | 4.2 | 4.1 | 4.1 | 4.1 |
| Hybrid method | $e_N$ | 6.6e−02 | 4.6e−04 | 1.2e−05 | 3.8e−07 | 1.2e−08 | 3.6e−10 | 1.1e−11 |
| | $O_N$ | | 7.2 | 5.2 | 5.0 | 5.0 | 5.0 | 5.0 |
| | $e_I$ | 2.4e−01 | 2.0e−03 | 5.8e−05 | 2.0e−06 | 6.9e−08 | 2.3e−09 | 7.2e−11 |
| | $O_I$ | | 6.9 | 5.1 | 4.8 | 4.9 | 4.9 | 5.0 |

TABLE 6.3
*Numerical results for example (6.3) with d = 6 and n = 8.*

| $N$ | | 10 | 20 | 40 | 80 | 160 | 320 | 640 |
|---|---|---|---|---|---|---|---|---|
| Starting procedure | $e_S$ | 1.8e−00 | 1.4e−03 | 2.0e−06 | 5.0e−09 | 1.6e−11 | 5.4e−14 | 4.4e−16 |
| | $O_S$ | | 10.4 | 9.4 | 8.7 | 8.3 | 8.2 | 6.9 |
| | $W$ | 107.6 | 12.4 | 5.4 | 3.8 | 4.2 | 4.6 | 4.9 |
| Global method | $e_N$ | 2.4e−01 | 2.0e−04 | 3.4e−07 | 2.9e−09 | 1.4e−11 | 6.2e−14 | 3.1e−15 |
| | $O_N$ | | 10.2 | 9.2 | 6.9 | 7.7 | 7.9 | 4.3 |
| | $e_I$ | 2.1e−00 | 1.5e−03 | 2.2e−06 | 8.8e−09 | 5.1e−11 | 2.9e−13 | 1.2e−14 |
| | $O_I$ | | 10.5 | 9.4 | 8.0 | 7.4 | 7.5 | 4.5 |
| Composite method | $e_N$ | 2.4e−01 | 2.0e−04 | 2.9e−07 | 2.2e−09 | 8.7e−12 | 3.0e−14 | 1.8e−15 |
| | $O_N$ | | 10.2 | 9.4 | 7.0 | 8.0 | 8.2 | 4.1 |
| | $e_I$ | 2.1e−00 | 1.5e−03 | 2.2e−06 | 7.2e−09 | 3.0e−11 | 2.0e−13 | 1.2e−14 |
| | $O_I$ | | 10.5 | 9.4 | 8.3 | 7.9 | 7.2 | 4.1 |
| Hybrid method | $e_N$ | 2.4e−01 | 1.4e−04 | 9.2e−07 | 6.4e−09 | 3.1e−11 | 1.4e−13 | 4.4e−16 |
| | $O_N$ | | 10.7 | 7.3 | 7.2 | 7.7 | 7.8 | 8.2 |
| | $e_I$ | 2.2e−00 | 1.6e−03 | 2.9e−06 | 9.5e−09 | 3.7e−11 | 1.5e−13 | 8.4e−15 |
| | $O_I$ | | 10.5 | 9.1 | 8.3 | 8.0 | 8.0 | 4.1 |

We also studied the classical nonlinear Volterra integral equation

$$(6.3) \qquad y(t) = e^{-t} + \int_0^t e^{s-t}\big(y(s) + e^{-y(s)}\big)\,\mathrm{d}s, \quad t \in [0, 10],$$

from [8, 19] with the exact solution $y(t) = \log(t+e)$. To solve the nonlinear system for the starting values and the nonlinear equations for the subsequent approximations, we used Newton's method, which is applicable, since we anyway need to impose enough differentiability of the involved functions to satisfy the hypotheses of our theoretical results. Table 6.3 shows the absolute errors and the estimated approximation orders of the proposed methods with $d = 6$ and $n = 8$. Here we achieve higher orders than in the cited references. Moreover, the errors with the composite method are at least as small as those obtained from the global and hybrid methods and, due to the high order, they reach machine precision with $N = 640$, which explains some smaller experimental orders in the last column of the table.

The presented methods can also handle oscillatory kernels; we emphasize, however, that there are more efficient methods for highly oscillatory problems; see, e.g., [30].
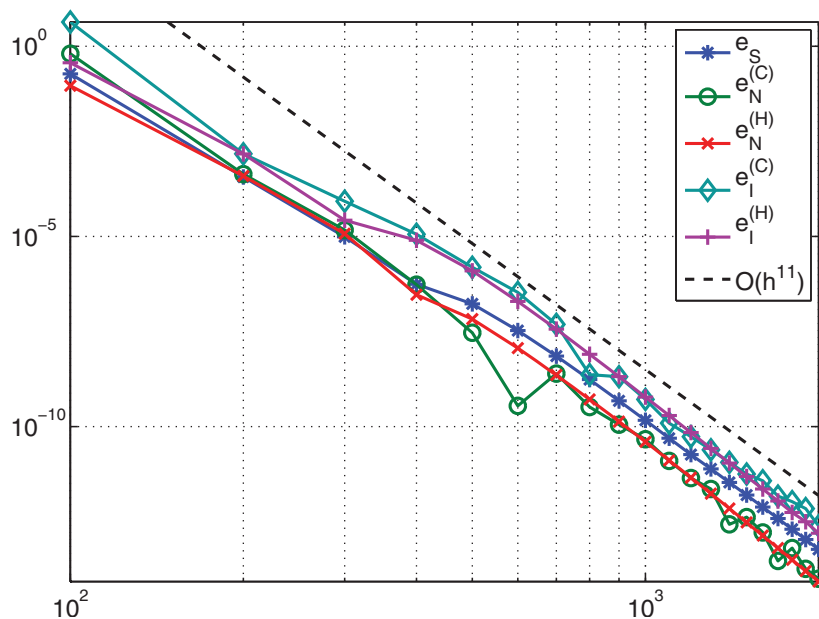
FIG. 6.1. *Numerical results for example (6.4) with $n = 15$ and $d = 9$, where $e_S$ is the error in the starting values, the superscript $(H)$ refers to the hybrid method and $(C)$ to the composite method, and the subscript $N$ refers to the error in the final value $y_N$ and $I$ to the error of the interpolated numerical solution.*

We studied the equation

(6.4)
$$y(t) = \cos(\omega t) + \frac{\lambda}{\omega^2 - \lambda^2}\big(\cos(\omega t) - \cos(\lambda t)\big) + \int_0^t \sin\big(\lambda(t - s)\big)y(s)\,\mathrm{d}s, \quad t \in [0, 2],$$

with $\omega = 50$ and $\lambda = 100$, whose exact solution is $y(t) = \cos(\omega t)$. The function $f$ and the kernel are very smooth, so that we can choose rather large values of the parameters: we took $n = 15$ and $d = 9$. For this example, we compared the composite method with the hybrid one, since a relatively large number of nodes was necessary to make the error decrease to nearly machine precision, and the global rational method was too expensive for that. The results for increasing values of $N$ are shown in Figure 6.1. As soon as the number of nodes is sufficiently large, the errors decrease rapidly. We observe again the similarity in the quality of approximation of the two methods.

The theoretical results in sections 3–5 show that the order of convergence of the method is limited by the differentiability of the function $f$ and of the kernel $K$. We demonstrate this by means of the equation

(6.5)
$$y(t) = f(t) + \int_0^t \cos(t + s)y(s)\,\mathrm{d}s, \quad t \in [0, 1],$$

where $f(t)$ is given by (1.1) and the exact solution by $y(t) = (t - c)^3, 0 \leq t \leq c$, and $y(t) = (t - c)^4, c < t \leq 1$. In (6.5), $f$ and $y$ are only twice continuously differentiable with a third derivative of bounded variation. In our tests, we took $n = 10$, $d = 5$, and $c = \sqrt{2}/6$ to avoid that the discontinuity be on a node, and to have it close enough to the left extremity of the interval, so that one can see two different error behaviors in
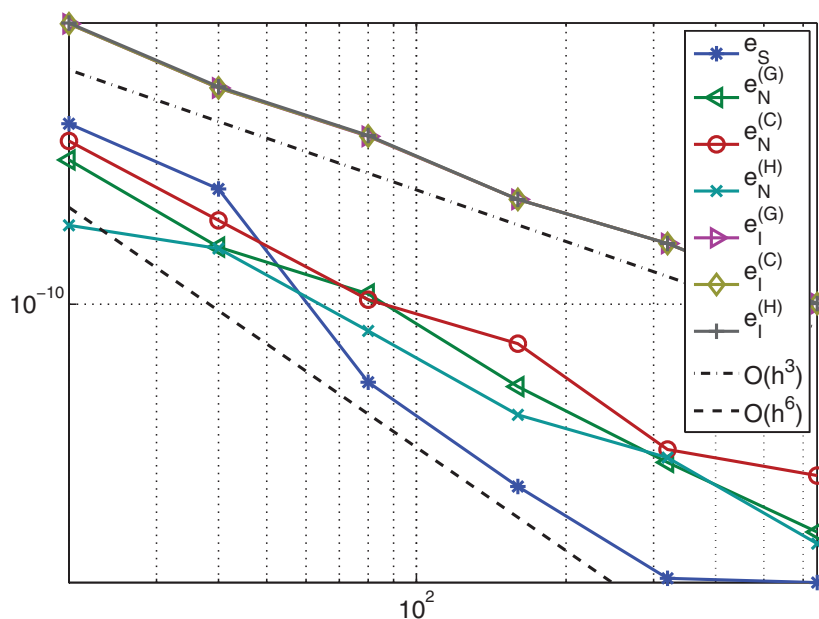
15

FIG. 6.2. *Numerical results for example* (6.5) *with* $n = 10$ *and* $d = 5$, *where the meaning of the labels is the same as in Figure* 6.1, *except for the superscript* $(G)$, *which refers to the global rational method.*

the starting values $y_1, \ldots, y_n$; see Figure 6.2. Indeed, as long as the starting procedure involves nodes on each side of the singularity $t = c$, the order is small because of the limited smoothness of $f$. Once $h$ is small enough, the starting procedure only involves nodes to the left of the singularity, where $f$ is infinitely smooth, and the observed order in the starting values is $d + 1 = 6$, as predicted by the theory for that case. The order of the interpolation of the overall solution is 3, while that of the values given by the three methods at the end of the interval is somewhat larger than 3, even a bit larger than what our theoretical results predict in this case; one can nevertheless clearly see the impact of the discontinuity in the third derivative.

As a final example, we solved the classical stiff nonlinear Volterra integral equation from [16, 8],

$$y(t) = f(t) - 10 \int_0^t \frac{1 + t}{1 + s} y^2(s) \, ds, \quad t \in [0, 19],$$

where

$$f(t) = \left((1 + t)e^{-10t} + 1\right)^{\frac{1}{2}} + (1 + t)(1 - e^{-10t}) + 10(1 + t)\log(1 + t)$$

and with the exact solution $y(t) = ((1 + t)e^{-10t} + 1)^{\frac{1}{2}}$. A Volterra integral equation is called stiff [8] when $\partial K / \partial y \ll 0$ or the Lipschitz constant of the kernel with respect to $y$ is large; in this example $\partial K / \partial y = -20$. We refrain from providing a table here, as the relative error with the rational methods jumps from about 100%, as long as $N$ is too small to resolve the stiffness, to the unavoidable rounding error when $N$ becomes large enough to accommodate the exponentially decaying solution. Instead, we illustrate with one sample that our rather general purpose composite rational method is stable enough to resolve even stiff equations until close to machine precision.

With $N = 2000$, $n = 10$, and $d = 5$, we observed $e_S = 7.0\mathrm{e}{-}09$, $e_N = 4.8\mathrm{e}{-}14$, and $e_I = 4.5\mathrm{e}{-}08$. With the hybrid method, we found $e_N = 1.0\mathrm{e}{-}13$ and $e_I = 1.6\mathrm{e}{-}08$. With this method, however, the error does not exhibit a jump in its decay as the composite rational method, but starts decreasing with smaller values of $N$ already. We did not carry out experiments with the global method on this example, as the computation time and cost would have been excessively high, whereas the composite method combined with Newton's method was still very fast.

**7. Conclusion.** We have presented two versions of an elegant and efficient quadrature method, based on linear barycentric rational interpolation, for the numerical solution of Volterra integral equations. For the sake of comparison, we have also suggested and tested a further method, namely, the combination of our linear rational starting procedure with Gregory quadrature for the main integration. The resulting method is in most cases as good as the global method to a much lesser cost. Like classical quadrature methods using composite Newton–Cotes rules, the input merely consists of a sample at equispaced values of the arguments of $f$ and $K$. The global version is especially elegant, as it integrates a single infinitely smooth interpolant on every interval. The composite version is slightly more involved, but much faster on long intervals. Both deliver an infinitely smooth approximation of the solution. The order may in principle be increased at will.

Every linear barycentric rational quadrature rule is based on the integration of the corresponding linear rational interpolant. Interpolation of the discrete solution therefore is straightforward. This may be of great help to devise a control mechanism of the step size to accommodate difficult stretches of the kernel or of the solution, such as steep gradients.

As other methods for such integral equations, it would be a little harder to implement when $K$ was merely known on the triangle $a \leq s \leq t \leq T$: a smaller interval length $h$ would have to be used in the starting procedure than in the main method. Further questions to address in the future include the stability behavior; see [8, 17].

**Appendix. Computation of the rational quadrature weights.** We give one possibility of computing the rational quadrature weights (2.5) using Chebfun [29] in MATLAB. Yet another possibility among many is to compute the integrals of the rational fundamental functions $\ell_k^{(\beta)}$ via Gauss–Legendre quadrature with sufficiently many points (usually 500–1000, depending on the involved parameters), which can be done efficiently, e.g., with the method from [14], also provided in the Chebfun toolbox. Computing the integrals with such a quadrature method can be faster than with Chebfun, since the rational interpolant does not need to be first approximated globally, but merely needs to be evaluated.

```
function omega = ratquadwts(x, d)
% Computation of the rational quadrature weights.
n = length(x) − 1;
w = fhbarywts(x, d);
omega = zeros(size(x));
f = omega;
for k = 1:(ceil(n/2) + 1)
   f(k) = 1;
   omega([k n+2−k]) = sum(chebfun(@(z) bary(z,f,x,w), x([1, n+1])));
   f(k) = 0;
end
```

```
function w = fhbarywts(x, d)
% Computation of the Floater--Hormann barycentric weights.
n = length(x) − 1;
w = zeros(size(x));
for k = 1:(n+1)
    for m = (k−d):k
        if (m < 1 || m > n + 1 − d), continue, end
        jj = m:(m+d);
        product = 1./prod(x(k)−x(jj(jj∼=k)));
        w(k) = w(k) + (−1)^m*product;
    end
end
```

## REFERENCES

[1] K. E. ATKINSON, *An Introduction to Numerical Analysis*, John Wiley, New York, 1978

[2] R. BALTENSPERGER, J.-P. BERRUT, AND B. NOËL, *Exponential convergence of a linear rational interpolant between transformed Chebyshev points*, Math. Comp., 68 (1999), pp. 1109–1120.

[3] Z. BATTLES AND L. N. TREFETHEN, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., 25 (2004), pp. 1743–1770.

[4] J.-P. BERRUT, *Rational functions for guaranteed and experimentally well-conditioned global interpolation*, Comput. Math. Appl., 15 (1988), pp. 1–16.

[5] J.-P. BERRUT, R. BALTENSPERGER, AND H. D. MITTELMANN, *Recent developments in barycentric rational interpolation*, in Trends and Applications in Constructive Approximation, M. G. de Bruin, D. H. Mache, and J. Szabados, eds., Internat. Ser. Numer. Math. 151, Birkhäuser, Basel, 2005, pp. 27–51.

[6] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517.

[7] L. BOS, S. DE MARCHI, K. HORMANN, AND G. KLEIN, *On the Lebesgue constant of barycentric rational interpolation at equidistant nodes*, Numer. Math., 121 (2012), pp. 461–471.

[8] H. BRUNNER AND P. J. VAN DER HOUWEN, *The Numerical Solution of Volterra Equations*, CWI Monogr., North-Holland, Amsterdam, 1986.

[9] T. A. DRISCOLL, *Automatic spectral collocation for integral, integro-differential, and integrally reformulated differential equations*, J. Comput. Phys., 229 (2010), pp. 5980–5998.

[10] M. S. FLOATER AND K. HORMANN, *Barycentric rational interpolation with no poles and high rates of approximation*, Numer. Math., 107 (2007), pp. 315–331.

[11] L. FOX AND E. T. GOODWIN, *The numerical solution of non-singular linear integral equations*, Philos. Trans. R. Soc. Lond., 245 (1953), pp. 501–534.

[12] S. GÜTTEL AND G. KLEIN, *Efficient high-order rational integration and deferred correction with equispaced data*, submitted.

[13] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations. I. Nonstiff Problems*, Springer, Berlin, 1987.

[14] N. HALE AND A. TOWNSEND, *Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights*, SIAM J. Sci. Comput., 35 (2013), pp. A652–A674.

[15] F. B. HILDEBRAND, *Introduction to Numerical Analysis*, McGraw-Hill, New York, 1956.

[16] F. DE HOOG AND R. WEISS, *Implicit Runge–Kutta methods for second kind Volterra integral equations*, Numer. Math., 23 (1975), pp. 199–213.

[17] S. A. HOSSEINI AND A. ABDI, *On the numerical stability of the linear barycentric rational quadrature method for Volterra integral equations*, in preparation.

[18] S. A. ISAACSON AND R. M. KIRBY, *Numerical solution of linear Volterra integral equations of the second kind with sharp gradients*, J. Comput. Appl. Math., 235 (2011), pp. 4283–4301.

[19] G. IZZO, E. RUSSO, AND C. CHIAPPARELLI, *Highly stable Runge–Kutta methods for Volterra integral equations*, Appl. Numer. Math., 62 (2012), pp. 1002–1013.

[20] G. Klein and J.-P. Berrut, *Linear barycentric rational quadrature*, BIT, 52 (2012), pp. 407–424.

[21] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.

[22] E. Martensen, *Optimale Fehlerschranken für die Quadraturformel von Gregory*, ZAMM, 44 (1964), pp. 159–168.

[23] E. Martensen, Zur Restglieddarstellung der Gregoryschen Quadraturformel ungerader Ordnung, Numer. Math. 15 (1970), pp. 229–233.

[24] D. O'Regan and M. Meehan, *Existence Theory for Nonlinear Integral and Integro-Differential Equations*, Kluwer Academic, Dordrecht, The Netherlands, 1998.

[25] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, *Impossibility of fast stable approximation of analytic functions from equispaced samples*, SIAM Rev., 53 (2011), pp. 308–318.

[26] J. Steinberg, *Numerical solution of Volterra integral equation*, Numer. Math., 19 (1972), pp. 212–217.

[27] L. N. Trefethen, *Computing numerically with functions instead of numbers*, Math. Comput. Sci., 1 (2007), pp. 9–19.

[28] L. N. Trefethen, *Is Gauss quadrature better than Clenshaw–Curtis?*, SIAM Rev., 50 (2008), pp. 67–87.

[29] L. N. Trefethen and others, Chebfun Version 4.2, The Chebfun Development Team, http://www.maths.ox.ac.uk/chebfun/ (2011).

[30] S. Xiang and H. Brunner, *Efficient methods for Volterra integral equations with highly oscillatory Bessel kernels*, BIT, 53 (2013), pp. 241–263.