Numerische Mathematik (2019) 141:1–19
https://doi.org/10.1007/s00211-018-0992-0

Numerische Mathematik

An improved Gregory-like method for 1-D quadrature

Bengt Fornberg[1] · **Jonah A. Reeger**[2]

Received: 5 January 2018 / Revised: 1 July 2018 / Published online: 16 August 2018

Abstract
The quadrature formulas described by James Gregory (1638–1675) improve the accuracy of the trapezoidal rule by adjusting the weights near the ends of the integration interval. In contrast to the Newton–Cotes formulas, their weights are constant across the main part of the interval. However, for both of these approaches, the polynomial Runge phenomenon limits the orders of accuracy that are practical. For the algorithm presented here, this limitation is greatly reduced. In particular, quadrature formulas on equispaced 1-D node sets can be of high order (tested here up through order 20) without featuring any negative weights.

Mathematics Subject Classification Primary: 65D30 · 65D32; Secondary 65B15

1 Introduction

For notational simplicity, we focus much of the following discussion on approximating $\int_0^\infty f(x)dx$ using the step size $h = 1$, i.e.

$$\int_0^\infty f(x)dx \approx \sum_{k=0}^{\infty} w_k f(k). \tag{1}$$

Support received from Department of Defense, the Office of Naval Research (Atmospheric Propagation Sciences of High Energy Lasers) and the Air Force Office of Scientific Research (Radial Basis Functions for Numerical Simulation). The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Navy, Department of Defense, or U.S. Government.

✉ Bengt Fornberg
fornberg@colorado.edu

Jonah A. Reeger
jonah.reeger@gmail.com

[1] Department of Applied Mathematics, University of Colorado, Boulder, CO 80309, USA

[2] Department of Mathematics, United States Naval Academy, Holloway Road, Chauvenet Hall, Annapolis, MD 21402-5002, USA

The quadrature weights w_k can then trivially be adjusted to step size h by multiplying each by h. A method is said to be accurate of order p if the error is of size $O(h^p)$ when $h \to 0$ for a sufficiently smooth function $f(x)$. Since we focus on approximations for which all the weights w_k are the same from some k and onward, a finite interval is treated by applying the non-trivial part of the weights sequence from each end. The non-trivial parts may overlap, as described below.

The next two Sects. 1.1 and 1.2 describe briefly the trapezoidal rule (TR) and its Newton–Cotes (NC) generalization. Following this, we conclude the Introduction by outlining the remaining parts of this paper.

1.1 The trapezoidal rule (TR)

The trapezoidal rule amounts to choosing $w_0 = \frac{1}{2}$ and $w_k = 1,\ k = 1, 2, 3, \ldots$ in (1). This is well known to be a second order method ($p = 2$). It is also well known that it becomes *exponentially accurate* (beyond any finite order) if applied to a periodic problem or if applied over an infinite (rather than semi-infinite) interval [22]. This tells that the dominant errors in the trapezoidal rule come from the ends of the interval (here just one end, at $x = 0$), and this naturally suggests the introduction of 'end corrections'. The Euler-Maclaurin formula, dating from around 1740, is one such approach. When using step h, it provides the asymptotic expansion

$$\int_0^\infty f(x)dx \sim \left(h \sum_{k=0}^{\infty} f(hk) \right) - \frac{1}{2} h f(0) + \frac{h^2}{12} f'(0) - \frac{h^4}{720} f'''(0) + \frac{h^6}{30240} f^{(5)}(0) - \frac{h^8}{1209600} f^{(7)}(0) + - \ldots \tag{2}$$

The coefficients for the derivative terms can be obtained in the same way as we will use in Sect. 2 for the Gregory coefficients; see also (11). While extremely powerful in many situations, the obvious drawback is that the derivative values usually are not known.

1.2 The Newton–Cotes (NC) formulas

One way to arrive at the TR is to approximate $f(x)$ by a linear function between any two adjacent nodes, and then integrate these line segments analytically. One can similarly group the sub-intervals in pairs, fit piece-wise parabolas to their three nodes, group in triples and fit cubics, etc. This leads to the sequence of NC formulas. While the accuracy orders increase (as it happens, in steps: $p = 2,\ 4,\ 4,\ 6,\ 6,\ 8,\ 8, \ldots$), the Runge phenomenon quickly makes these schemes impractical. Polynomial interpolation over increasingly many equi-spaced nodes will feature large oscillations near the end of each sub-interval [9,21]. Corresponding quadrature weights will also be large and oscillatory, as illustrated below in Sect. 2.2. Another disadvantage with the NC formulas is that their weights do not settle in to all become equal. In order to

compensate for errors that are entirely caused by the boundary, the weights are made oscillatory across the complete interval (a strange approach for correcting an error that is local in nature).

1.3 Structure of the paper

Given the background above, it is remarkable that an elegant correction strategy for boundary errors was described well before the developments by Newton and Cotes (which in turn preceded the works by Euler and Maclaurin). With the Gregory method, described in Sect. 2, the boundary corrections are local to the boundary. However, as for Newton–Cotes, the Runge phenomenon causes problems when p is increased. The present approach for addressing this is described in Sect. 3, together with test results. This is followed by our conclusions.

2 Gregory's method

2.1 Derivation

With the notation $\Delta f(x_k) = f(x_{k+1}) - f(x_k)$, it follows (in the case of $h = 1$) that

$$\begin{aligned}
\Delta^0 f(0) &= f(0) \\
\Delta^1 f(0) &= f(1) - f(0) \\
\Delta^2 f(0) &= f(2) - 2f(1) + f(0) \\
\Delta^3 f(0) &= f(3) - 3f(2) + 3f(1) - f(0) \\
&\vdots \qquad \vdots
\end{aligned}$$

with coefficients from Pascal's triangle. We next look for an improved TR formula of the form

$$\int_0^\infty f(x)dx \sim \left(\sum_{k=0}^{\infty} f(k)\right) + \left[b_0\Delta^0 + b_1\Delta^1 + b_2\Delta^2 + \ldots\right] f(0), \tag{3}$$

where, in practice, we will use only a finite number of correction terms. A function over $x \in [-\infty, \infty]$ is commonly represented as a combination of Fourier modes e^{-zx} with z purely imaginary. Similarly, for functions over $x \in [0, \infty]$, one would naturally also allow z to have a positive real part. Substituting $f(x) = e^{-zx}$ into (3) gives

$$\frac{1}{z} = \frac{1}{1-e^{-z}} + \left[b_0 - b_1(1-e^{-z}) + b_2(1-e^{-z})^2 - b_3(1-e^{-z})^3 + - \ldots\right]$$

and, with

$$w = (1 - e^{-z}), \tag{4}$$

i.e. $z = -\log(1-w)$,

$$\frac{1}{\log(1-w)} + \frac{1}{w} = -b_0 + b_1 w - b_2 w^2 + b_3 w^3 - + \ldots \tag{5}$$

The coefficients b_i can readily be calculated recursively from the Taylor expansion of $\log(1-w)$:

$$b_0 = -\frac{1}{2},\ b_1 = \frac{1}{12},\ b_2 = -\frac{1}{24},\ b_3 = \frac{19}{720},\ b_4 = -\frac{3}{160},\ b_5 = \frac{863}{60480},$$
$$b_6 = -\frac{275}{24192}, \ldots \tag{6}$$

Explicit formulas include

$$b_i = \frac{-1}{(i+1)!} \int_0^1 \left(\prod_{j=0}^{i} (x-j) \right) dx \tag{7}$$

$$= (-1)^{i+1} \int_0^\infty \frac{dx}{\left((\log x)^2 + \pi^2\right)(x+1)^{i+1}} \tag{8}$$

and expressions in terms of Bernoulli and Stirling numbers [16]. Using only $b_0 = -\frac{1}{2}$ turns (3) into the TR. For each further term, the accuracy order increases by one.

The passage shown in Fig. 1 appears in a letter that James Gregory wrote in 1670 [13]. We recognize here exactly the same expansion as in (3), (6) (however, with a typo in one of the denominators; 164 instead of 160). It is noteworthy that this letter by Gregory well precedes the first publications on calculus, by Leibniz (1684) and Newton (1687), respectively, as well as Brook Taylor's description in 1715 of what has since become known as Taylor expansions. It is suggested in [12] that Gregory might have refrained from publishing his (genuinely original) work at the time, due to concern that Newton already then possibly had similar ideas[1]. The early history of calculus may well have developed differently, had it not been for Gregory's premature death in 1675 (of stroke, at age 36).

Gregory's method has received relatively little recent attention. Exceptions include the observation in [6] how the $p = 3$ version (also known as either Lacroix's rule or Durand's method) can be generalized to irregular node spacing, put in a more general context in [14]. Even order Gregory rules can be related to spline quadrature [7] Section 10.7, [8]. A hybrid Newton–Cotes–Gregory concept is discussed in [4]. The use of Gregory's method for solving Volterra integral equations is described in [5]. Modern numerical analysis text books virtually always describe the NC methods, but surprisingly rarely the Gregory approach.

[1] Gregory's letter of 1670 (to John Collins, written mostly in English) indeed begins with "I suppose these series I send you here enclosed, may have some affinity with those inventions you advertise me that Mr. Newton have discovered." Only extracts of the letter are preserved. These do not include Gregory's derivation.

Fig. 1 Brief extract from the bottom of page 208 and the top of page 209 in [13]

ponendo $AP = PO = c$
$PB = d$
primam ex differentiis { primis $= f$; secundis $= h$; tertiis $= i$; quartis $= k$; quintis $= l$ }

et omnes differentias affici signo $+$, erit $ABP =$

$$\frac{dc}{2} - \frac{fc}{12} + \frac{hc}{24} - \frac{19ic}{720} + \frac{3kc}{160} - \frac{863lc}{60480} + \&c. \text{ in infinitum.}$$

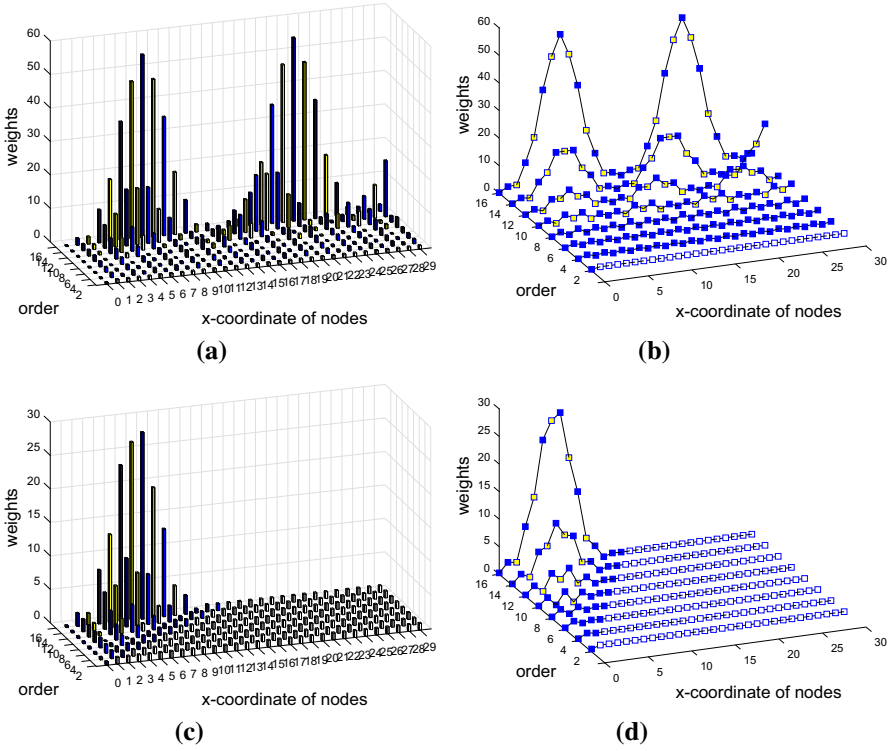

Fig. 2 Bar- and curve-type displays of the magnitudes of the leading NC and Gregory weights. Weights equal to one are shown white, else blue denotes positive and yellow negative weights. Note the factor of two difference in vertical scaling between the NC and Gregory plots. **a** NC weights (bars), **b** NC weights (curves), **c** Gregory weights (bars) and **d** Gregory weights (curves) (color figure online)

2.2 Comparison of weights for Newton–Cotes and Gregory formulas of increasing orders

Figure 2 shows the NC and the Gregory weights, respectively, for accuracy orders $p = 2, 4, 6, \ldots, 16$. Heights correspond to magnitude, with blue meaning a positive weight, and yellow a negative weight. Uncolored markers indicate weights that

are exactly equal to one. For the NC cases, this happens only for the second order trapezoidal rule (front row) whereas each row in the Gregory case takes this form following location $p-2$. The second row in the NC case is the familiar Simpson's rule, with weights $\frac{1}{3}\cdot\{1,\ 4,\ 2,\ 4,\ 2,\ 4,\ 2,\ldots\}$, the third row is known as Boole's rule. It is visually clear that the NC and the Gregory formulas are both severely affected by the Runge phenomenon, and we see in both cases negative weights for orders $p=10$ and above. For the same order, the weights are about twice as large in the NC case. However, the main difference is that, in the Gregory case, non-trivial weights occur only near to the boundary, whereas they for NC repeat periodically forever (or until a right boundary is reached).

Table 1 gives the exact values for the non-trivial (not equal to one) Gregory weights up through order $p=10$.

2.3 Comparison of accuracies for the Newton–Cotes and Gregory formulas

Based on the derivation in Sect. 2.1, a natural way to display the error in a quadrature scheme with weights $w_k,\ k=0,1,2,\ldots$ is to evaluate

$$\text{Error}(z)=\left|\int_0^\infty f(x)dx-\sum_{k=0}^{\infty}w_k f(k)\right| \tag{9}$$

for $f(x)=e^{-zx}$, with z near the origin in the right half-plane. Better still would be to let $z=re^{i\theta}$ and then display over a $\{\log r,\theta\}$-plane, with the $\log r$ scaling in the radial direction motivated by the fact that $\log_{10}\text{Error}(z)$ will then be near-planar surfaces over the $\{\log r,\theta\}$-plane, with slopes in the $\log r$-direction that directly reveal the scheme's actual order of accuracy. Since it 'so happens' that $u(z)=\log z=\log|z|+i\arg z=\log r+i\theta$, we can equivalently interpret the displays as showing $\log_{10}\text{Error}(z)$ as function of $u=\log z$ (complex). For the NC and the Gregory formulas of orders $2,4,8,12,16,$ this produces the result shown in Fig. 3, with slopes matching their respective orders of accuracy. For each order from 4 and upwards, the error constant is seen to be slightly smaller for the NC formulas than their Gregory counterparts. However, Fig. 2 showed their weights to oscillate with about twice as large amplitudes. Comparing the schemes on the basis of errors vs. amplitude of oscillations in the weights, the NC and the Gregory formulas roughly break even.

In the same way as used here for Fig. 3, we can similarly display the accuracy achieved by any weight set w_k, and in this way compare quadrature schemes (as we will do later in part (b) of Figs. 7 and 8).

Away from boundaries, the Gregory approach fully utilizes the exponential accuracy of the TR. In periodic and infinite interval $(-\infty<x<+\infty)$ cases, the NC methods lose rather than gain accuracy when their formal order of accuracy is increased. For example, Simpson's rule can be viewed as the first step of either Richardson extrapolation or Romberg's method [19], i.e. as forming the linear combination of TR results based on steps h and $2h$, as $S(h)=\frac{4}{3}T(h)-\frac{1}{3}T(2h)=T(h)+\frac{1}{3}(T(h)-T(2h))$. With TR exponentially accurate (for sufficiently smooth data), its error is $\varepsilon=O(e^{-c/h})$,

Table 1 The non-trivial Gregory weights w_k up through order $p = 10$

$p =$	Weights									
2	$\frac{1}{2}$									
3	$\frac{5}{12}$	$\frac{13}{12}$								
4	$\frac{3}{8}$	$\frac{7}{6}$	$\frac{23}{24}$							
5	$\frac{251}{720}$	$\frac{299}{240}$	$\frac{211}{240}$	$\frac{739}{720}$						
6	$\frac{95}{288}$	$\frac{317}{240}$	$\frac{23}{30}$	$\frac{739}{720}$	$\frac{157}{160}$					
7	$\frac{19087}{60480}$	$\frac{84199}{60480}$	$\frac{18869}{30240}$	$\frac{37621}{30240}$	$\frac{55031}{60480}$	$\frac{61343}{60480}$				
8	$\frac{5257}{17280}$	$\frac{22081}{15120}$	$\frac{54851}{120960}$	$\frac{103}{70}$	$\frac{89437}{120960}$	$\frac{16367}{15120}$	$\frac{23917}{24192}$			
9	$\frac{1070017}{3628800}$	$\frac{5537111}{3628800}$	$\frac{103613}{403200}$	$\frac{261115}{145152}$	$\frac{298951}{725760}$	$\frac{515677}{403200}$	$\frac{3349879}{3628800}$	$\frac{3662753}{3628800}$		
10	$\frac{25713}{89600}$	$\frac{1153247}{725760}$	$\frac{130583}{3628800}$	$\frac{903527}{403200}$	$-\frac{797}{5670}$	$\frac{6244961}{3628800}$	$\frac{56621}{80640}$	$\frac{3891877}{3628800}$	$\frac{1028617}{1036800}$	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

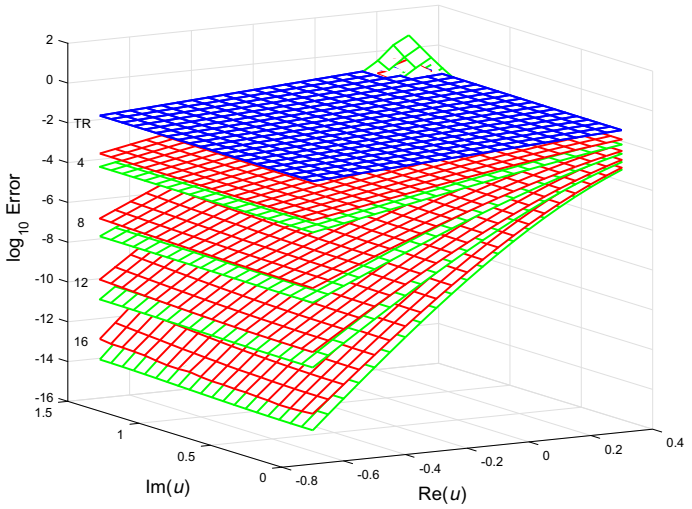

Fig. 3 $\log_{10}$ Error(u) displayed in part of the upper half-plane of $u(z) = \log r + i\theta$ for the Newton–Cotes and Gregory formulas of orders 2 (=TR), 4, 8, 12, 16, respectively. The two schemes are identical in case of order 2 (=TR), colored blue. For higher orders, the Newton–Cotes surfaces (green) fall slightly below the corresponding Gregory surfaces (red) (color figure online)

implying that the Simpson error becomes $O(\varepsilon^{1/2})$, i.e. with only about half as many correct digits as for TR.

2.4 Some comments on negative quadrature weights

If a function can fluctuate up to $\pm\varepsilon$, its integral over an interval of length L can fluctuate at most by $\pm\varepsilon L$. For a numerical quadrature scheme to always preserve this property, it is necessary that all its weights are non-negative. Failing this, even low level 'noise' (such as machine rounding errors) can get amplified. Although there are rare applications when even a single negative quadrature weight can cause an instability, the much weaker requirement that the sum of the magnitude of all the weights remain bounded under refinement suffices to ensure convergence for all continuous integrands [17].

2.5 An RBF-FD variation of Gregory's formulas

The first indication that the Runge problem can be greatly reduced for increasing order Gregory-type methods emerged from [18]. The problem considered there was far more general—numerical quadrature using scattered nodes over bounded curved surfaces in 3-D space. To verify that the RBF-FD (radial basis function-generated finite difference) approach employed was computationally competitive in its handling of surface edges, an extremely simplified test case was considered: 1-D equi-spaced nodes on a finite interval. It then transpired that the RBF-FD approach produced Gregory-like methods of high orders, but with reduced Runge phenomenon. This in turn was in line with a

similar RBF-FD observation for derivative approximations, as described in [3]. For a general background on RBF-FD approximations, see [10,11]. These results motivated the present study, aiming to more directly obtain quadrature formulas for equi-spaced node sets, without any reference to RBFs.

3 The present method

3.1 Concept

Weights w_k that make (1) exact for all functions $e^{-z\,x}$ would satisfy $\frac{1}{z} = \sum_{k=0}^{\infty} w_k e^{-z\,k}$ (with Re $z > 0$). Subtracting from this the identity $\frac{1}{1-e^{-z}} = \sum_{k=0}^{\infty} 1 \cdot e^{-z\,k}$ then gives

$$\frac{1}{z} - \frac{1}{1-e^{-z}} = \sum_{k=0}^{\infty} d_k e^{-z\,k}, \tag{10}$$

where $d_k = w_k - 1$.

The requirement Re $z > 0$ is necessary. With $z = x + iy$ and $x = 0$, the left hand side (LHS) of (10) evaluates to $-\frac{1}{2} + i\left(-\frac{1}{y} + \frac{1}{2}\cot\left(\frac{y}{2}\right)\right)$. The right hand side (RHS) takes then the form of a Fourier series in y but, lacking terms with negative k-indices, it cannot match the LHS even for $-\pi < y < \pi$. For what we want (high order agreement for z near the origin), a better starting point is to Taylor expand the LHS of (10):

$$\frac{1}{z} - \frac{1}{1-e^{-z}} = -\frac{1}{2} - \frac{z}{12} + \frac{z^3}{720} - \frac{z^5}{30240} + \frac{z^7}{1209600} - + \ldots\,. \tag{11}$$

Except for the sign of the constant term, these coefficients are the negatives of those in (2), unsurprisingly since the derivation indicated above for (2) utilizes this relation (11). Since we aim for only a finite number of non-zero coefficients d_k, both sides of (10) are analytic functions around the origin, and we can equate leading coefficients.

The task has now become to find sequences $\{d_k\}$ that

1. satisfy $d_k \geq -1$ (or, at least, stay bounded in magnitude),
2. approach zero rapidly, and then become identically zero from some early point onward, and
3. satisfy (10) to high orders in z when expanded around $z = 0$.

The strategy will be to decide on some order to which we want to exactly match the Taylor expansions of the two sides of (10), and then use a few more than the minimal number of d_k coefficients. Exploiting the freedom of having more variables than equations allows us to look for solutions that also satisfy the requirements 1,2 above. Assume that we want to match the two sides of (10) up through power z^n using a coefficient set $\{d_k,\ k = 0, 1, 2, \ldots, N\}$, with all further d_k, $k = N+1, N+2, \ldots$ equal to zero. The constraint equations then become

$$\begin{bmatrix} 0^0 & 1^0 & 2^0 & \cdots\cdots & N^0 \\ 0^1 & 1^1 & 2^1 & \cdots\cdots & N^1 \\ \vdots & \vdots & \vdots & & \vdots \\ 0^n & 1^n & 2^n & \cdots\cdots & N^n \end{bmatrix}_{(n+1)\times(N+1)} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_N \end{bmatrix}_{N+1} = \begin{bmatrix} -1/2 \\ 1/12 \\ 0 \\ -1/120 \\ 0 \\ \vdots \end{bmatrix}_{n+1} . \tag{12}$$

We have here also indicated the sizes of the matrix and vectors. The elements in the RHS vector are $B_{k+1}/(k+1)$, $k = 0, 1, 2, \ldots, n$ where the B's are the *Bernoulli numbers*. Choosing $N = n$, this becomes a square Vandermonde linear system; non-singular, but with some very small singular values, thereby producing the very large coefficients $\{d_k\}$ that characterize the Runge phenomenon.

3.2 Alternate matrix formulation

As shown in the Appendix (in two different ways), the linear system (12) can be rewritten as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & & \cdots & \cdots \\ & 1 & 2 & 3 & 4 & \cdots & \{\text{Pascal's} & \cdots & \cdots \\ & & 1 & 3 & 6 & \cdots & \text{triangle}\} & \cdots & \cdots \\ & & & 1 & 4 & \cdots & \cdots & \cdots & \cdots \\ & & & & 1 & \cdots & \cdots & \cdots & \cdots \\ & & & & & \ddots & \cdots & \cdots & \cdots \end{bmatrix}_{(n+1)\times(N+1)} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_N \end{bmatrix}_{N+1} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}_{n+1} . \tag{13}$$

The entries b_j, $j = 0, 1, 2, \ldots, n$ in the RHS vector are the same as those given in (6). These oscillate in sign while decreasing slowly to zero in magnitude; cf. Fig. 4. For j very large (beyond the current range of interest), it follows from (7) that $b_j \sim \frac{(-1)^{j+1}}{j\,(\log j)^2}$.

In the Gregory-type case of $N = n$, the Pascal matrix is square and its inverse is another Pascal matrix with oscillating signs. Also replacing the RHS vector with its numerical values gives

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_N \end{bmatrix}_{N+1} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & \cdots & \\ & 1 & -2 & 3 & -4 & \cdots & \\ & & 1 & -3 & 6 & \{\text{Pascal's} & \\ & & & 1 & -4 & \text{triangle}\} & \\ & & & & 1 & \cdots & \\ & & & & & \ddots & \end{bmatrix}_{(N+1)\times(N+1)} \begin{bmatrix} -0.5000 \\ 0.0833 \\ -0.0417 \\ 0.0264 \\ -0.0187 \\ 0.0143 \\ -0.0114 \\ \vdots \end{bmatrix}_{N+1} . \tag{14}$$

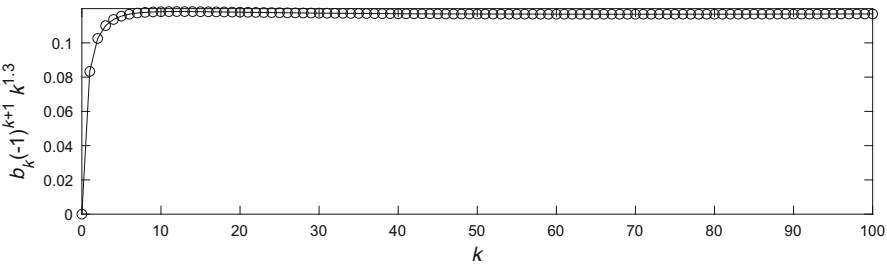

Fig. 4 The sequence $b_k(-1)^{k+1}k^{1.3}$ as function of $k = 0, 1, 2, \ldots, 100$

From this representation, the growth of the Gregory weights become an immediate consequence of the growth in the Pascal triangle entries (note that, due to the sign patterns, no cancellations will occur).

Returning to the case of $N > n$, and Eq. (13), this same Pascal triangle growth means that already small values of the auxiliary coefficients (above d_n) will be able to allow reductions in size of the lower ones (d_0 to d_n).

3.3 Numerical implementation

The strategy already outlined is to choose $N > n$, making either (12) or, equivalently, (13) under-determined, and then look for solutions that, for example, minimize

$$\sum_{k=0}^{N} s^{2k}\, d_k^2, \tag{15}$$

or

$$\sum_{k=0}^{N} s^{k}\ |d_k|\,, \tag{16}$$

Here, s is a scalar number somewhat larger than one (forcing the $\{d_k\}$ sequence to be mostly decaying in magnitude). The choice of s represents a compromise between (i) s is close to one: The $N+1$ non-trivial weights $w_k,\ k = 0, 1, \ldots, N$ will not converge towards one, and (ii) s large: The first $n+1$ weights will approach a Gregory-type case, with large oscillations. Somewhere in-between, best found by trial-and-error, the weight oscillations will be mild, while the weights still approach the value of one.

After we have applied either L_2 or L_1 minimization to the sequence $\{s^k d_k\},\ k = 0, 1, \ldots, N$ (corresponding to minimizing (15) and (16), respectively) to create the $N+1$ nontrivial weights w_k, these are placed from the left end of the interval, followed by as many (if any) trivial weights as needed to reach the opposite end of the interval. If the interval was wide enough that the nontrivial weights do not reach the middle of the interval, the same set can be applied also from the right side. Whenever the non-trivial weights, starting from left, do not extend beyond the right side of the interval, an easy strategy is to apply the weight set from the left and supplement with as many (if any) trivial weights as needed to reach the right side. This weight set is then reflected

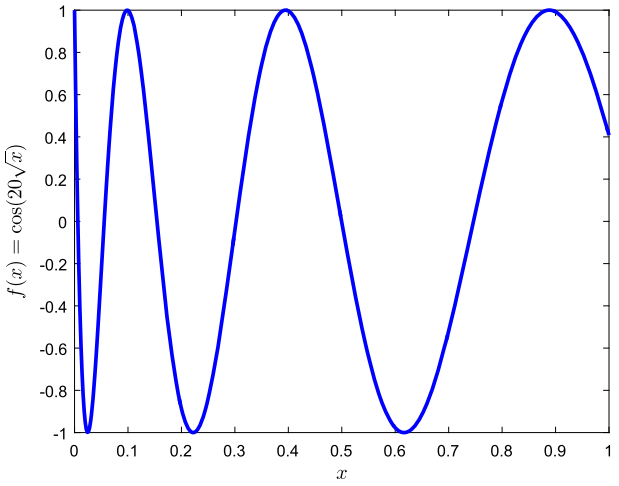

Fig. 5 Test function used for comparing quadrature accuracies: $f(x) = \cos(20\sqrt{x})$

left-right, added to the original set, and the trivial weight subtracted from each of the entries. As long as the number $N + 1$ of non-trivial weights equals or is less than the total number of nodes across the interval, this latter strategy can be applied (without needing to check if the sets of nontrivial weights applied from the two ends of the interval, overlap or not).

3.4 Test problem

Together with displaying the weights for some different schemes, and showing their analytic error surfaces in the complex $u = \log z$-plane, we will also apply the weight sets to a test function, which we (quite arbitrarily) choose as

$$f(x) = \cos(20\sqrt{x}) \ , \tag{17}$$

over [0, 1] (see Fig. 5), using increasingly many equispaced nodes. This function is infinitely differentiable. It is very steep at the left boundary, and has high curvature near to it. The exact value for the integral is $\int_0^1 \cos(20\sqrt{x})dx = \frac{1}{200}(\cos 20 + 20 \sin 20 - 1)$. Figure 6 shows the errors that we obtain for this test function when using the Gregory methods of orders 10 and 16 (red curves, solid and dashed, respectively), and also (in blue) the errors with the three instances of the L_2 and L_1 implementations that are described in more detail next (all three with non-negative weights).

3.5 L_2 implementation

This minimization becomes a regular least squares minimization if we simply divide the successive columns of the matrix in (12) by $1, s, s^2, s^3, \ldots, , s^N$, respectively.

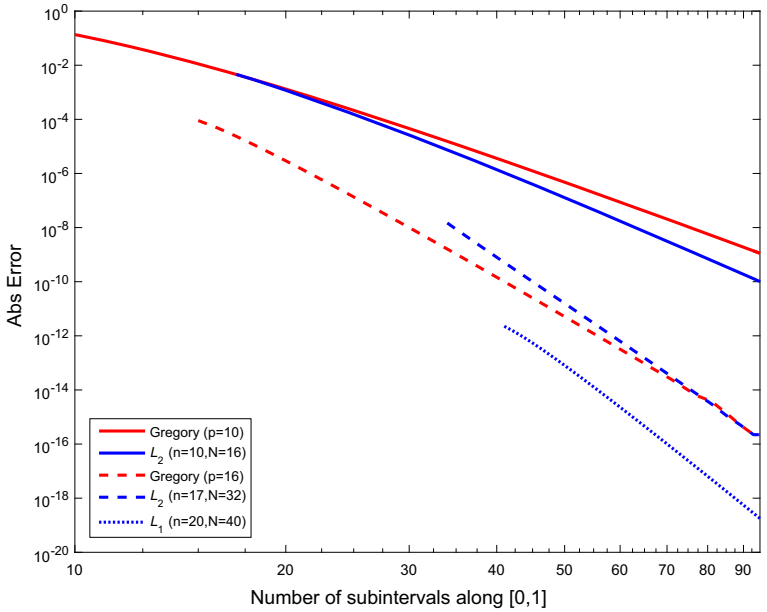

Fig. 6 Errors in evaluating the test integral $\int_0^1 \cos(20\sqrt{x})dx$ with two Gregory methods and three (non-negative weight) minimization-generated schemes, as described in Sects. 3.5 and 3.6. The two Gregory curves have slopes that reflect their orders of accuracy $p = 10$ and $p = 16$, respectively. The different start points for the curves reflect the number of sub-intervals required before the respective method can be applied

Minimal norm solutions of this kind can be obtained directly with MATLAB's function *lsqminnorm*, or by taking the pseudo-inverse of the matrix.

When n and N are relatively large, standard double precision becomes insufficient, since the elements of the coefficient matrices in both (12) and (13) become very different in size—see Sect. 3.7 below. For the test examples in this section, the weight calculations were therefore carried out in extended precision, using the Advanpix software package [1].

The weights of the order $p = 10$ and order $p = 16$ Gregory methods were displayed as the 4$^{\text{th}}$ to last and the last cases, respectively, in Fig. 2c, d, and they are shown again as red markers in part (a) of Figs. 7 and 8. The blue bars in these figures show for comparison the weights the present algorithm produces when using the parameter values described in the respective figure captions. For this L_2 implementation, there are no longer any negative weights. Parts (b) of the figures use our previously described $u = \log z$-plane displays (cf. Fig. 3) to illustrate that the formal orders of the Gregory and L_2 schemes are comparable.

From a practical perspective, the classical Gregory formulas are well suited up to orders around 8 (and have then no negative weights). The present L_2 version provides Gregory-like formulas, again with no negative weights, up through orders around $p = 16$. Pursuing still higher orders, a small modification of the present L_2 algorithm can be considered. If any negative weights w_k were to appear, the corresponding values

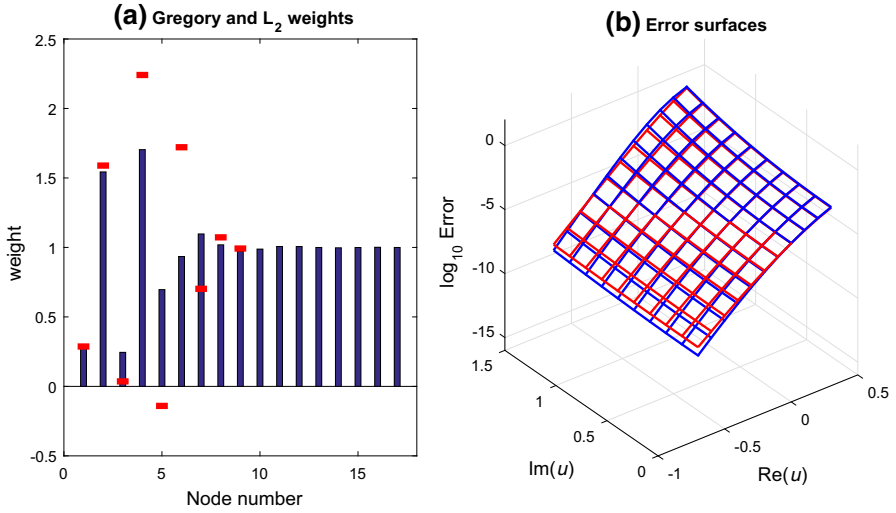

Fig. 7 Comparison between Gregory of order 10 and the present L_2 scheme with the parameter settings $n = 10,\ N = 16,\ s = 1.6$ (red and blue, respectively): **a** non-trivial weights at the left end of the interval, **b** accuracy display in the complex $u = \log z$-plane (color figure online)

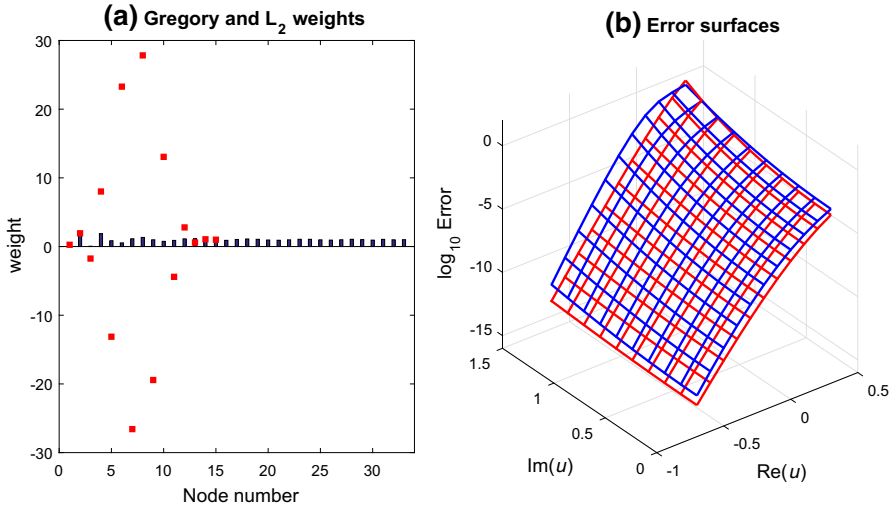

Fig. 8 Comparison between Gregory of order 16 and the present L_2 scheme with the parameter settings $n = 17,\ N = 32,\ s = 1.06$ (red and blue, respectively): **a** non-trivial weights at the left end of the interval, **b** accuracy display in the complex $u = \log z$-plane (color figure online)

for d_k can be set to -1, and the linear system reduced correspondingly in size, after which the least square process can be repeated. While direct enforcement of the the $d_k \geq -1$ inequalities is possible in the context of L_2 minimization, the task then no longer simplifies to the solution of a single linear system. Turning to L_1 minimization

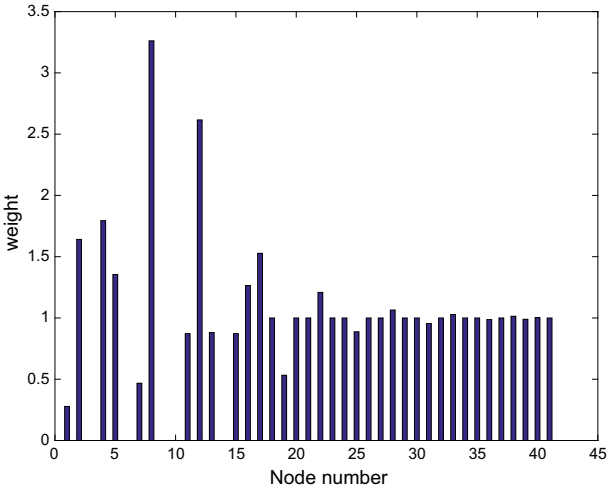

Fig. 9 Weights obtained when using the L_1 approach, and choosing $n = 20,\ N = 40,\ s = 1.2$. Like for the L_2-generated schemes shown in Figs. 7 and 8, the errors when applied to $\int_0^1 \cos(20\sqrt{x})dx$ are shown in Fig. 6

may then be simpler, since such minimization codes frequently allow for both equality and inequality constraints.

3.6 L_1 implementation

Using this time Mathematica, and its built-in function *NMinimize*, we choose $n,\ N,\ s$ and then minimize (16) subject to (12) and to the further conditions $d_k \geq -1,\ k = 0, 1, \ldots, N$. Figure 9 shows for $n = 20,\ N = 40,\ s = 1.2$ the weights obtained. We have omitted comparisons against the 20th order Gregory scheme, since its weights are impractically large (in the range $[-277, +274]$). Applied to our test function (17), Fig. 6 shows it to be the most accurate of all the quadrature schemes considered here.

3.7 Numerical conditioning

With the standard definition of a matrix condition number (ratio of largest to smallest singular value), Table 2 shows $\log_{10}$ of the condition numbers of the Vandermonde and Pascal matrices in Eqs. (12) and (13), respectively, in the test cases illustrated in Figs. 7, 8 and 9. Given that we typically want to obtain weights accurate to machine precision (in double precision 16 decimal digits), we conclude from Table 2 that quad precision (about 34 decimal digits) generally suffices when using the Pascal matrix formulation, but that still higher precision often is needed in the Vandermonde case. Given that the two formulations are mathematically equivalent, we recommend for all cases the Pascal matrix formulation.

Table 2 $\log_{10}$ of the condition numbers for the Vandermonde (V) and Pascal (P) formulations in the test cases shown in Figs. 7, 8 and 9

Figures	n	N	s	$\log_{10}\text{cond}(V)$	$\log_{10}\text{cond}(P)$
7	10	16	1.6	11.4	3.7
8	17	32	1.06	27.0	10.3
9	20	40	1.2	32.0	11.1

4 Conclusions

Computational algorithms often become more cost-effective when their formal order of accuracy is increased, be it for solving ODEs, PDEs, interpolation, or numerical quadrature. In cases where equispaced data is available from one side only, the well-known Runge phenomenon arises—high order polynomial interpolants tend to feature increasingly large oscillations near the ends of an interval. Two desirable features of quadrature methods for equispaced cases are

1. All weights are non-negative (or, at least, stay small in magnitude),
2. Weights become constant some distance away from boundaries (and thereby independent of the total number of nodes; also making the scheme fully able to utilize the trapezoidal rule's spectral accuracy away from boundaries).

Newton–Cotes and Gregory schemes meet the first requirement only for relatively low orders of accuracy. Gaussian quadrature methods achieve non-negative weights, but are not an option if the data is given only at equispaced locations. Their weights depend on the total number of nodes across the interval. The weights in Romberg's quadrature method [19] will also be non-negative, but they oscillate in a somewhat complicated pattern across the full integration interval, which typically will need to have a number of nodes that grows exponentially with the desired order of accuracy. Among classical quadrature schemes for equispaced nodes, the Gregory methods stands out as always satisfying the second condition.

Recent work in the area of RBF-FD [3,18] showed it to be possible to greatly reduce the Runge phenomenon for equispaced node layouts while still featuring high orders of accuracy. In the present study, we have seen that RBFs are not necessary for achieving this. After noting that the classical NC formulas do not have any significant advantage over the even more classical formulas by Gregory, we introduced an approach by which both high orders of accuracy and non-negative weights can be achieved for equispaced quadrature on a bounded interval. It does not entirely eliminate the issue of negative quadrature weights for arbitrarily high orders of accuracy, but it much delays the onset of this adverse property. The more limited goal of keeping the magnitude of the weights relatively small can be achieved for any order.

If the data is not required to be equi-spaced and also if extremely high accuracies are required (well beyond the standard 16 digits for double precision), it is usually beneficial to use refinement near the ends of the interval. It is noted in [2] that Gaussian quadrature is effective up to a few hundred decimal places and that, beyond this, the *tanh-sinh* method [20] often becomes the best option. In this approach, a certain variable change is followed by equispaced TR integration.

Appendix: Derivations of the alternate matrix formulation

Derivation based on Eq. (10)

Changing variable $z = -\log(1-w)$ in (10) gives

$$\begin{aligned} -\left(\frac{1}{\log(1-w)}+\frac{1}{w}\right) &= \sum_{k=0}^{\infty} d_k e^{k\log(1-w)} \\ &= \sum_{k=0}^{\infty} d_k (1-w)^k = \sum_{k=0}^{\infty} d_k \left(\sum_{i=0}^{k} \binom{k}{i}\right) (-1)^i w^i \\ &= \sum_{i=0}^{\infty} (-1)^i w^i \left(\sum_{k=i}^{\infty} d_k \binom{k}{i}\right). \end{aligned}$$

By (5), the LHS above equals $\sum_{i=0}^{\infty}(-1)^i b_i$. Equating coefficients gives (13).

Derivation based on matrix algebra

There are numerous exact relations for Vandermonde matrices, especially in equi-spaced cases. Starting by LU-factorizing it, it transpires that the linear system (12) can be rewritten as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & \cdots & \cdots \\ & 1 & 2 & 3 & 4 & \cdots & \{\text{Pascal's} & \cdots \\ & & 1 & 3 & 6 & \cdots & \text{triangle}\} & \cdots \\ & & & 1 & 4 & \cdots & \cdots & \cdots \\ & & & & 1 & \cdots & \cdots & \cdots \\ & & & & & \ddots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_N \end{bmatrix} = \\ = \begin{bmatrix} \frac{1}{0!} & & & & \\ & \frac{1}{1!} & & & \\ & & \frac{1}{2!} & & \\ & & & \ddots & \\ & & & & \frac{1}{n!} \end{bmatrix} \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ -1 & & 1 & & & \\ & 2 & & -3 & & 1 \\ -6 & & 11 & & -6 & & 1 \\ \vdots & & \vdots & & \vdots & & \vdots & \ddots \end{bmatrix} \begin{bmatrix} -1/2 \\ 1/12 \\ 0 \\ -1/120 \\ 0 \\ \vdots \end{bmatrix}. \tag{18}$$

The entries in the second matrix in the RHS are the first order Stirling numbers $S(i,j)$, $i,j = 0,1,2,\ldots$, with generating function

$$\prod_{j=0}^{i-1}(x-j) = \sum_{j=0}^{i} S(i,j)\, x^j. \tag{19}$$

These numbers are readily calculated by the 'Pascal-like' recursion $s(i,\,j) = s(i-1,\,j-1) - (i-1)s(i-1,\,j)$, initiated by the trivial values when $i = 0$ and $j = 0$. By the identity

$$\sum_{j=1}^{i+1} S(i,\,j-1)\frac{B_j}{j} = -\frac{1}{i+1}\sum_{j=0}^{i+1} S(i+1,\,j)\,\frac{1}{j+1}$$

([15]; combine (7), p. 147 with (5), p. 249), the RHS of (18) simplifies to

$$-\begin{bmatrix} \frac{1}{1!} & & & & \\ & \frac{1}{2!} & & & \\ & & \frac{1}{3!} & & \\ & & & \ddots & \\ & & & & \frac{1}{(n+1)!} \end{bmatrix} \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ 2 & -3 & 1 & & & \\ -6 & 11 & -6 & 1 & & \\ 24 & -50 & 35 & -10 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/3 \\ 1/4 \\ 1/5 \\ 1/6 \\ \vdots \end{bmatrix}. \tag{20}$$

There is now a leading minus sign, both matrices have been shifted one step up and left, and the RHS vector has become much simplified. The result

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & & \cdots & \cdots \\ & 1 & 2 & 3 & 4 & \cdots & \{\text{Pascal's} & \cdots & \\ & & 1 & 3 & 6 & \cdots & \text{triangle}\} & \cdots & \\ & & & 1 & 4 & \cdots & \cdots & \cdots & \\ & & & & 1 & \cdots & \cdots & \cdots & \\ & & & & & \ddots & \cdots & \cdots & \end{bmatrix}_{(n+1)\times(N+1)} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_N \end{bmatrix}_{N+1} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}_{n+1} \tag{21}$$

now follows from combining (7) and (19) with the additional observation that integration in x of the monomials $x,\ x^2,\ x^3,\ \ldots$ produce the factors $\frac{1}{2},\ \frac{1}{3},\ \frac{1}{4},\ \ldots$, matching the RHS vector in (20). The equality of the right hand sides of (18) and (21) can alternatively be deduced from equation (8) in [16].

References

1. Advanpix: Multiprecision computing toolbox for MATLAB. Advanpix LLC, Yokohama, Japan. http://www.advanpix.com/. Accessed 8 Aug 2018
2. Bailey, D.H., Borwein, J.M.: High-precision numerical integration: progress and challenges. J. Symb. Comput. **46**, 741–754 (2011)
3. Bayona, V., Flyer, N., Fornberg, B., Barnett, G.A.: On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. J. Comput. Phys. **332**, 257–273 (2017)
4. Bocher, P., De Meyer, H., Berghe, G.: On Gregory- and modified Gregory-type corrections to Newton–Cotes quadrature. J. Comput. Appl. Math. **50**, 145–158 (1994)
5. Brunner, H., van der Houwen, P.J.: The Numerical Solution of Volterra equations. CWI Monographs. Elsevier, Amsterdam (1986)
6. De Swardt, S.A., De Villiers, J.M.: Gregory type quadrature based on quadratic nodal spline interpolation. Numer. Math. **85**, 129–153 (2000)

7. De Villiers, J.: Mathematics of Approximation. Atlantis Press, Amsterdam (2012)
8. De Villiers, J.M.: A nodal spline interpolant for the Gregory rule of even order. Numer. Math. **66**, 123–137 (1993)
9. Fornberg, B.: A Practical Guide to Pseudospectral Methods. Cambridge University Press, Cambridge (1996)
10. Fornberg, B., Flyer, N.: A Primer on Radial Basis Functions with Applications to the Geosciences. SIAM, Philadelphia (2015)
11. Fornberg, B., Flyer, N.: Solving PDEs with radial basis functions. Acta Numer. **24**, 215–258 (2015)
12. Fraser, D.C.: Newton's interpolation formulas. Further notes. J. Inst. Actuar. **52**(274), 117–135 (1920)
13. Gregory, J.: Letter to J. Collins, 23 November 1670, pp. 203–212. In: Rigaud: Correspondence of Scientific Men. Oxford University Press (1841)
14. Javed, M., Trefethen, L.N.: Euler–Maclaurin and Gregory interpolants. Numer. Math. **132**, 201–216 (2016)
15. Jordan, C.: Calculus of Finite Differences, 2nd edn. Chelsea, New York (1950)
16. Phillips, G.M.: Gregory's method for numerical integration. Am. Math. Mon. **79**(3), 270–274 (1972)
17. Pólya, G.: Über die Konvergenz von Quadraturverfahren. Math. Zeitschrift. **37**, 264–286 (1933)
18. Reeger, J.A., Fornberg, B.: Numerical quadrature over smooth surfaces with boundaries. J. Comput. Phys. **355**, 176–190 (2018)
19. Romberg, W.: Vereinfachte numerische Integration. Det Kongelige Norske Videnskabers Selskab Forhandlinger **28**(7), 30–36 (1955)
20. Takahasi, H., Mori, M.: Double exponential formulas for numerical integration. Res. Inst. Math. Sci. **9**, 721–741 (1974)
21. Trefethen, L.N.: Approximation Theory and Approximation Practice. SIAM, Philadelphia (2013)
22. Trefethen, L.N., Weideman, J.A.C.: The exponentially convergent trapezoidal rule. SIAM Rev. **56**, 384–458 (2014)