Numerical Algorithms (2023) 92:65–88
https://doi.org/10.1007/s11075-022-01462-0

ORIGINAL PAPER

Extrapolation quadrature from equispaced samples of functions with jumps

Jean–Paul Berrut[1] · Manfred R. Trummer[2]

Received: 28 April 2022 / Accepted: 15 November 2022 / Published online: 19 December 2022

Abstract
Based on the Euler–Maclaurin formula, the Romberg quadrature method extrapolates trapezoidal values to improve their accuracy when computing the integral of smooth functions from equispaced samples. It has been known at least since an article of Lyness in 1971 that the Euler–Maclaurin formula may be extended to accommodate functions with jumps. In the present work, we develop an extrapolation method, based on this extended formula, for the quadrature of such discontinuous functions. We illustrate the method with numerical examples, using one as well as several sample vectors.

Keywords Numerical quadrature · Equispaced samples · Jumps · Euler–Maclaurin formula · Extrapolation

Mathematics Subject Classification (2010) 65D30 · 65B15 · 65B05

1 Introduction

Consider the numerical approximation of the definite integral

$$I := \int_0^L f(x)dx \tag{1}$$

To Claude Brezinski, the founder and editor-in-chief of this journal, on his 80th birthday

✉ Jean–Paul Berrut
jean-paul.berrut@unifr.ch

Manfred R. Trummer
trummer@sfu.ca

[1] Département de Mathématiques, Université de Fribourg, Pérolles, CH-1700 Fribourg, Switzerland

[2] Department of Mathematics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada

of a smooth function f by sampling it at equispaced points (or nodes) $x_k := kh,\ k = 0, \ldots, n$, $h = L/n$, $f_k := f(x_k)$, and evaluating the composite trapezoidal rule

$$T_f(h) := h\left(\frac{f(x_0)}{2} + \sum_{k=1}^{n-1} f(x_k) + \frac{f(x_n)}{2}\right).$$

In this work, we abide by the signal processing (Wikipedia) definition of a sample as a value at a point in time or space. A sample vector is a vector of equispaced samples.

It is well known that the accuracy of $T_f(h)$ depends on the values of the derivatives of odd orders of f at the extremities 0 and L. In the favorable case where all of those coincide at these endpoints, the convergence is faster than every power of h, i.e., spectral. If f, L-periodically extended on both sides of the interval $[0, L]$, is analytic in a horizontal strip containing $\mathbb{R}$, the convergence even becomes exponential. A survey [19] about $T_f(h)$ has been published in 2014.

In many numerical analysis courses, students learn that the difference between the (composite) trapezoidal rule and the exact integral is given by the *Euler–Maclaurin formula* [2, 13].

Theorem 1 (Euler--Maclaurin formula for the trapezoidal rule) *Let $f \in C^{2m+2}[0, L]$ for some $m \geq 0$. Then, for every $n \in \mathbb{N}$ and $h := \frac{L}{n}$, the error of the trapezoidal rule may be written as follows:*

$$T_f(h) - I = a_2h^2 + a_4h^4 + \ldots + a_{2m}h^{2m} + L\frac{B_{2m+2}}{(2m+2)!}f^{(2m+2)}(\xi)h^{2m+2} \quad (2)$$

for some $\xi \in [0, L]$, where

$$a_{2j} := \frac{B_{2j}}{(2j)!}[f^{(2j-1)}(L) - f^{(2j-1)}(0)].$$

Here B_ℓ denotes the ℓth Bernoulli number.

There are at least two ways of deriving $T_f(h)$. The customary one is to consider I as the sum of the integrals over the subintervals $[x_k, x_{k+1}]$ and to replace each of these integrals with the area of the trapezoid with bases $f(x_k)$ and $f(x_{k+1})$ and height h. But one may also see $T_f(h)$ (up to a constant depending on the length of the interval) as the exact integral of the trigonometric polynomial of minimal degree which interpolates the function values $\big(f(x_0) + f(x_n)\big)/2$, $f(x_1), \ldots, f(x_{n-1})$ at the nodes $x_0, \ldots, x_{n-1}$ [1, 2]. That way, formula (2) is a quantitative description of the influence of the jump at $0 \equiv L \pmod{L}$ of the L-periodically extended function f (on both sides of $[0, L]$) on the accuracy of the exact integral $T_f(h)$ of that trigonometric polynomial as an approximation of I.

In the present work, we shall address the following problem: we assume that a vector (in Section 7 a few vectors) of equispaced samples of a piecewise smooth function is (are) given, together with the location of its jumps. As usual in quadrature by extrapolation, we shall first assume that the number of subintervals is some power of two (or an odd number times such a power). It turns out that a large number of nodes are required in many cases. We therefore also consider the case where several vectors of samples can be taken.

In Section 2, we recall the Euler–Maclaurin formula for functions with jumps, the basis of the extrapolation method presented in this work. The method is described for the case of a single interior jump in Section 3, and an example is given in Section 4. The next section introduces the general method, of which examples are given in Section 6. Up to that point, we use a single vector of samples; Section 7 extends the method to the case where several vectors of samples can be obtained, which can greatly reduce the total number of function evaluations required to achieve a prescribed level of accuracy, and gives corresponding examples. A few remarks conclude the paper.

2 The Euler–Maclaurin formula for functions with jumps

According to the circular interpretation mentioned above, the L-periodized f usually has a jump at $c_0 := 0 \equiv L \pmod{L}$. Nevertheless, when we talk about jumps in the present work, we mean the usual concept, i.e., that f has interior jumps, i.e., jumps at some points c_j, $0 < c_j < L$, $j = 1, \ldots, J$. We assume that these c_j are distinct and ordered according to their index, and that f is continuous on every interval (c_j, c_{j+1}) between two consecutive jumps, and possesses one-sided limits $f(c_j^-)$ and $f(c_j^+)$ at c_j, $j = 0, 1, \ldots, J-1$. Moreover, the value of f at every jump is changed, if necessary, to the average $(f(c_j-)+f(c_j+))/2$ of the two corresponding limits there.

As in [2], we consider the classic Riemann sum generalization of $T_f(h)$ with "offset" t, where $0 \le t < 1$,

$$R_f(h) := h\sum_{k=0}^{N-1} f\big((k+t)h\big) = h\sum_{k=1}^{N} f\big((k-1+t)h\big). \tag{3}$$

For $t = 0$, we have $R_f(h) = T_f(h)$; for $t = \frac{1}{2}$, this is the midpoint rule.

Let us recall the following theorem of [2], which will be the basis of our extrapolation procedure. The result was already given by Lyness in [14] (see the discussion in [2, p. 383]).

Theorem 2 (Euler–Maclaurin formula for Riemann sums of piecewise smooth functions) *Let f be piecewise $C^{m-1}[0, L]$, and pick a t, $0 \le t < 1$. Let c_j denote the abscissae of the $J+1$ jumps of f and, for each of them, $t_j := t - \frac{c_j}{h}$ mod 1, $j = 0, \ldots, J$.*

If $f^{(m)}$ is absolutely integrable between every two consecutive c_j, then the integration error may be written as follows:

$$R_f(h) - I = a_1 h + \sum_{\ell=2}^{m} a_\ell h^\ell - \frac{h^m}{m!}\int_0^L f^{(m)}(x)\sum_{j=0}^{J}\overline{P}_m(t_j - \frac{x}{h})dx \tag{4}$$

with

$$a_1 := \sum_{j=0}^{J}\gamma_j P_1(t_j)\left[f(c_j-) - f(c_j+)\right], \qquad \gamma_j := \begin{cases}0, & t_j = 0,\\ 1, & t_j \ne 0,\end{cases} \tag{5}$$

and

$$a_\ell := \sum_{j=0}^{J} \frac{P_\ell(t_j)}{\ell!} \left[f^{(\ell-1)}(c_j-) - f^{(\ell-1)}(c_j+) \right], \tag{6}$$

where P_ℓ is the Bernoulli polynomial of degree ℓ and $\overline{P}_\ell$ its continuous 1-periodic extension.

In the sequel, we shall refer to formula (4) by the acronym EMF.

The change of variable $y = L(x-a)/(b-a)$ shows that the polynomial part of $R_f(h) - I$ is the same when f is defined on an arbitrary interval $[a, b]$ instead of $[0, L]$; the latter only makes the formulas and the proofs simpler to read.

In the exceptional case where the values of the derivatives are known on both sides of the jumps, substituting them into (4) easily permits the improvement of the Riemann values (3) (see [2]). One can also correct the endpoint weights in $T_f(h)$ to increase its order [8].

The number t_j is the relative distance of c_j to the node following it. In the rather exceptional case where the values of h are such that $t_j = 1/2$ (or constant) for every h, one can directly apply Richardson–Romberg (see Example 1 of [2], where the only interior jump is at the midpoint of the interval of integration). But this is atypical and, usually, the t_j depend on h, so that the factors a_ℓ of the powers of h in the EMF are not constants as they depend on values of the Bernoulli polynomials at the t_j.

3 Extrapolation via a system of equations

Extrapolation eliminates the first powers of h on the right-hand side of the EMF without knowledge of the a_ℓ. Assume that a quantity I is approximated by a formula $B(h)$, and that the latter may be written as follows:

$$B(h) = I + a_1 h + a_2 h^2 + \ldots + a_p h^p + S_p(h) = C_p(h) + S_p(h) \tag{7}$$

with $S_p(h) = \mathcal{O}(h^{p+1})$. Here $B(h) = R_f(h)$ as given in (3). One method for successively eliminating the first powers of h proceeds by linear combination of values of $B(h)$ for different arguments h; another solves a system of equations, and this is what we shall use here.

Suppose $B(h_0), B(h_1), \ldots, B(h_p)$ have been evaluated for a decreasing sequence of h_k monotonely approaching 0. The system of equations with unknowns $\widetilde{I}$ and $\widetilde{a}_\ell$

$$\widetilde{I} + h_k \widetilde{a}_1 + h_k^2 \widetilde{a}_2 + \ldots + h_k^p \widetilde{a}_p = B(h_k), \qquad k = 0, \ldots, p,$$

determines the polynomial $Q_p(h)$ of degree at most p

$$Q_p(h) = \widetilde{I} + \widetilde{a}_1 h + \widetilde{a}_2 h^2 + \ldots + \widetilde{a}_p h^p$$

interpolating the values $B(h_k)$, $k = 0, \ldots, p$.

Recall that the Lebesgue function $\lambda_p(h)$ is defined as follows:

$$\lambda_p(h) = \sum_{k=0}^{p} |\ell_k(h)|, \tag{8}$$

in which the ℓ_k are the Lagrange fundamental polynomials

$$\ell_k(h) = \frac{\prod_{j=0, j\neq k}^{p}(h - h_j)}{\prod_{j=0, j\neq k}^{p}(h_k - h_j)}.$$

The Lebesgue function is the norm of the interpolation operator and indicates the conditioning of the interpolant (see for example [20]).

Theorem 3 *Let the function $B(h)$ of (7) be interpolated in the interval $[0, h_0]$ by the polynomial $Q_p(h)$ described above, and let the Lebesgue function $\lambda_p(h)$ be $\mathcal{O}\big(g(h)\big)$ near 0, for some function $g(h)$. Then*

$$\widetilde{I} - I = \mathcal{O}\big(g(h)h^{p+1}\big). \tag{9}$$

Proof Obviously, $\widetilde{I} - I = Q_p(0) - C_p(0)$. In view of the linearity and uniqueness of polynomial interpolation, the polynomial part $C_p(h)$ of $B(h)$ is interpolated exactly by $Q_p(h)$, so that $Q_p(h) - C_p(h)$ is the interpolant of $S_p(h)$. But, according to a well-known result about the perturbation of the polynomial interpolant [3, 7],

$$|Q_p(h) - C_p(h)| \leq \lambda_p(h)|S_p(h)|.$$

Letting $h \to 0$ in this expression yields (9). $\square$

With a good set of interpolation nodes $\{h_k\}$, such as Chebyshev or Legendre points, one has $\lambda_p(h) = \mathcal{O}(\log p)$ [6], so that the error becomes $\mathcal{O}(h^{p+1} \log p)$. Apart from that of the Chebyshev points of the first kind, we do not know bounds on the Lebesgue function for sets of nodes which accumulate toward a point outside of the interpolation interval, such as 0 here, but our computations with the geometrically decreasing sequence of the first example below demonstrate that its growth at 0 is indeed very slow in comparison with that of $1/h$.

It remains to compute $\widetilde{I} = Q_p(0)$. Since it is the value at a single point of the interpolating polynomial Q_p, one usually uses Neville's algorithm [17], which is stable in the context of numerical quadrature, where the h–values h_k cluster toward the interpolation point 0 and the coefficients a_ℓ do not depend on h_k. However, in our case, they do (through the t_j). General Neville–Aitken type interpolation algorithms going into that direction have been given in the literature. According to the MathSciNet review of [11], Schneider [16] introduced the first one in a special case, before Mühlbach [15] solved the general problem. Brezinski [4] and Håvie [11] later gave different proofs. These same authors [5, 10] also independently derived a general extrapolation algorithm, which Brezinski called the E-algorithm. But none of these articles contains much in terms of numerical experiments demonstrating the stability of the methods, if any; moreover, none seems to have been used to treat functions with interior singularities.

Here we use the (simpler) matrix version (corresponding to the Vandermonde solution to the interpolation problem) and solve the resulting system of linear equations by means of Gaussian elimination.

4 A first example with one jump

As a test and an example illustrating the method, we have started with a function which numerically has no jump at the extremities of the interval and only one in the interior,

$$f(x) := 2e^{-\eta(2x-1)^2} g(x), \tag{10}$$

with

$$g(x) := \begin{cases} \cos(\beta x), & 0 \le x \le c_1, \\ e^{x-c_1}, & c_1 < x < 1. \end{cases} \tag{11}$$

f is plotted for $c_1 = 1/\sqrt{3}$, $\beta = 2$, and $\eta = 35$ in Fig. 1. Here and in the subsequent figures, a small circle at a jump c_j marks the value $(f(c_j-) + f(c_j+))/2$ used in the trapezoidal sum when c_j is one of the nodes.

We limit our computations to the case $t = 0$, i.e., $R_f(h) = T_f(h)$, the composite trapezoidal rule. As the sequence of h_k-values, we have taken the geometric sequence $h_k = 2^{-k}$, as in classic Richardson extrapolation [12, p. 100].

As a means of comparison, we have first applied Richardson extrapolation to the problem of computing (1) with $L = 1$, $c_1 = 1/\sqrt{3}$ (this choice guarantees that the jump at c_1 does not coincide with an integration node for any h) and $\beta = 2$. η is introduced so that all derivatives of f numerically vanish at the extremities 0 and 1; hence, from a practical point of view, only one jump is present, namely at c_1: we chose $\eta = 35$.

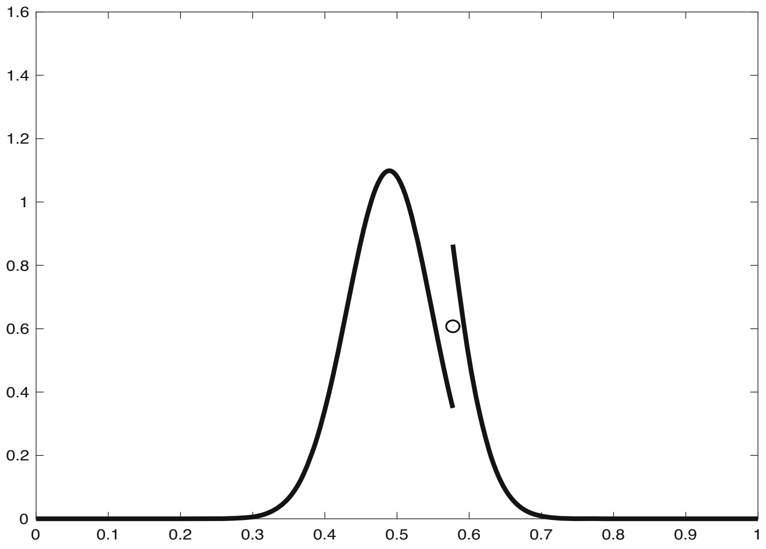

Fig. 1 Function f from (10) with $c_1 = 1/\sqrt{3}$, $\beta = 2$ and $\eta = 35$ on [0, 1]

We have used two calls of the MATLAB quad–routine, one on each subinterval $[0, c_1]$ and $[c_1, 1]$, to obtain an accurate approximate value of (1) with which to compare our results: $I \approx 0.180560634293184$. One could also use Chebfun with "splitting on." (We note, however, that these algorithms are no competitors to our method, as we assume that f is given by one vector of equispaced samples only.) Our results are summarized in Table 1. The first column contains the sequence of h_k-values, the second the error of the trapezoidal value for h_k without extrapolation, and the third the error in the Richardson-extrapolated value (the value at zero of the polynomial of the lowest degree interpolating the trapezoidal values up to h_k, computed here with the `polyfit` routine of MATLAB). The last column shows the error $\widetilde{I} - I$ with the extrapolation method presented below. Every row contains the results when the given samples are those corresponding to the value h_k in the first column, and the extrapolation makes use of all the trapezoidal values up to this h_k.

The trapezoidal values decrease, but not monotonically: this is to be expected, as the relative location of the jump in the interval between the two nodes enclosing it varies with h (see also the values of $t_1(h)$ in the next tables).

The Richardson values do not improve on the direct trapezoidal ones. This is hardly surprising, as the main part of the Euler–Maclaurin formula is no polynomial in h, because of the t_j.

Let us now write down the extrapolation equations of the suggested method. Since there is no jump at the extremities, $f^{(\ell-1)}(c_0-) = f^{(\ell-1)}(c_0+)$ for all ℓ, so that the sums in the a_ℓ start with $j = 1$ ($t_0 = 0$ is present only when there is a jump at the extremities). We denote by

$$d_1^{(\ell)} := f^{(\ell-1)}(c_1-) - f^{(\ell-1)}(c_1+), \qquad \ell = 1, 2, \ldots,$$

Table 1 Example with one interior jump and no jump at the extremities

h_k	Trapezoidal rule error	Richardson error	Error $\widetilde{I} - I$ with extrapolation
1	$-1.8056e-01$		
1/2	$3.5974e-01$	$9.0004e-01$	$-1.3391e-02$
1/4	$8.9754e-02$	$-5.4033e-01$	$-1.3125e-02$
1/8	$4.5382e-03$	$2.2912e-02$	$9.4367e+12$
1/16	$-8.6898e-03$	$2.8676e-03$	$-8.2194e-03$
1/32	$-7.5809e-04$	$2.1220e-02$	$-4.7358e-03$
1/64	$3.7568e-03$	$5.9615e-03$	$-5.2189e-04$
1/128	$1.6354e-03$	$-6.6700e-03$	$1.9178e-05$
1/256	$6.0816e-04$	$7.7813e-04$	$-5.9895e-09$
1/512	$1.0287e-04$	$-4.8423e-04$	$4.1272e-10$
1/1024	$-1.4770e-04$	$-3.9391e-04$	$4.7887e-09$
1/2048	$-2.1896e-05$	$4.7029e-04$	$8.1089e-12$
1/4096	$4.1135e-05$	$3.7051e-05$	$-1.5894e-12$
1/8192	$9.6524e-06$	$-1.0899e-04$	$-3.9972e-12$

the differences of derivatives of f at the jump c_1, which are not known. Then, according to (4), the trapezoidal value equals

$$\begin{aligned} T_f(h) \;=\; & I + \gamma_1 P_1(t_1(h)) d_1^{(1)} h + \frac{P_2(t_1(h))}{2!} d_1^{(2)} h^2 + \frac{P_3(t_1(h))}{3!} d_1^{(3)} h^3 + \ldots \\ & + \frac{P_q(t_1(h))}{q!} d_1^{(q)} h^q + \mathcal{O}(h^q), \end{aligned}$$

where the dependence of t_1 on h is now explicit. We approximate the right-hand side by the pseudo-polynomial with $p+1 \le q$ terms

$$Q_p(h) = \widetilde{I} + \gamma_1 P_1(t_1(h)) \widetilde{d}_1^{(1)} h + P_2(t_1(h)) \widetilde{d}_1^{(2)} \frac{h^2}{2!} + \ldots + P_p(t_1(h)) \widetilde{d}_1^{(p)} \frac{h^p}{p!}$$

for values $\widetilde{I}, \widetilde{d}_1^{(1)}, \ldots, \widetilde{d}_1^{(p)}$ approximating $I, d_1^{(1)}, \ldots, d_1^{(p)}$. The interpolation conditions $Q_p\big(\frac{h_0}{2^i}\big) = T_f\big(\frac{h_0}{2^i}\big)$, $i = 0, \ldots, p$, lead to the system of $p+1$ equations

$$\mathbf{AH} \begin{bmatrix} \widetilde{I} \\ \widetilde{d}_1^{(1)} \\ \widetilde{d}_1^{(2)} \\ \vdots \\ \widetilde{d}_1^{(p)} \end{bmatrix} = \begin{bmatrix} T_f(h_0) \\ T_f\big(\frac{h_0}{2}\big) \\ T_f\big(\frac{h_0}{4}\big) \\ \vdots \\ T_f\big(\frac{h_0}{2^p}\big) \end{bmatrix} \tag{12}$$

in the $p+1$ unknowns $\widetilde{I}, \widetilde{d}_1^{(1)}, \widetilde{d}_1^{(2)} \ldots, \widetilde{d}_1^{(p)}$, with the matrices

$$\mathbf{A} := \begin{bmatrix} 1 & \gamma_1(h_0) P_1(t_1(h_0)) & P_2(t_1(h_0)) & \ldots & P_p(t_1(h_0)) \\ 1 & \frac{1}{2}\gamma_1(\frac{h_0}{2}) P_1(t_1(\frac{h_0}{2})) & \frac{1}{2^2} P_2(t_1(\frac{h_0}{2})) & \ldots & \frac{1}{2^p} P_p(t_1(\frac{h_0}{2})) \\ 1 & \frac{1}{4}\gamma_1(\frac{h_0}{4}) P_1(t_1(\frac{h_0}{4})) & \frac{1}{4^2} P_2(t_1(\frac{h_0}{4})) & \ldots & \frac{1}{4^p} P_p(t_1(\frac{h_0}{4})) \\ \vdots & & \vdots & & \\ 1 & \frac{1}{2^p}\gamma_1(\frac{h_0}{2^p}) P_1(t_1(\frac{h_0}{2^p})) & \frac{1}{(2^p)^2} P_2(t_1(\frac{h_0}{2^p})) & \ldots & \frac{1}{(2^p)^p} P_p(t_1(\frac{h_0}{2^p})) \end{bmatrix}$$

(compare with [9, p. 220]) and

$$\mathbf{H} := \operatorname{diag}\Big(1, h_0, \frac{h_0^2}{2!}, \frac{h_0^3}{3!}, \ldots, \frac{h_0^p}{p!}\Big).$$

We merely need the first component of the solution $\mathbf{x} := \Big[\widetilde{I}, \widetilde{d}_1^{(1)}, \widetilde{d}_1^{(2)}, \ldots, \widetilde{d}_1^{(p)}\Big]^T$. Since the first element of the diagonal of $\mathbf{H}$ is 1, it suffices to find the first component of the solution of $\mathbf{Ay} = \mathbf{b}$ ($\mathbf{y} := \mathbf{Hx}$), where $\mathbf{b}$ is the right-hand side of (12).

If the jump is a quadrature node for every h_k, then $\gamma_1 = 0$ for all of these and the second column of $\mathbf{A}$ vanishes. Then the unknown $\widetilde{d}_1^{(1)}$ must be removed, together with the second column and the last row of $\mathbf{A}$, $\mathbf{H}$ becomes $\operatorname{diag}(1, h_0^2/2!, h_0^3/3!, \ldots, h_0^p/p!)$ and the last component of the right-hand side of (12) must be discarded as well.

These equations could probably be solved analytically to obtain Neville-like formulas. As mentioned at the end of Section 3, we solve the system with Gaussian elimination, in which pivoting almost always guarantees a stable method. This is more expensive, but every value of the Neville tableau is computed independently, so that any numerical error arising in one of the entries does not affect the following ones. Our results are in the last column of Table 1.

The values display a pattern that is common to several of our examples. When starting with $h_0 = b-a$, as we do, quite a large number of subdivisions of the interval are needed to reach the point (here $h_k = 1/64$) where the extrapolated values become better than those obtained with the plain trapezoidal method. But once this stage is attained, our extrapolation scheme rapidly outperforms the latter and we always obtain an accuracy below our tolerance (usually $N * 10^{-16}$, where $N+1$ is the size of the sample vector).

The huge value for $h = 1/8$ is consistent with the large condition number of $\mathbf{A}$: see Section 6.2 for a discussion.

Our function $\cos(2x)$ may seem too nice and simple. We also computed with a more challenging function, namely $\cos(100x)$, as well as Runge's function centered at $c_1/2$, both of which are much more difficult to interpolate with equispaced points: the results are very similar, and the method continues to perform well.

5 Generalization to several jumps

We now address the general case, where f may have several interior jumps at $c_1, c_2, \ldots, c_J$, $J \geq 1$. Since with the trapezoidal rule the extremities 0 and L are quadrature points, the jump at the endpoints has $\gamma_0 = 0$ and formula (4) becomes

$$T_f(h) = I + \sum_{j=1}^{J} \gamma_j P_1(t_j) d_j^{(1)} h + \sum_{\ell=2}^{q} \left(\sum_{j=0}^{J} \frac{P_\ell(t_j)}{\ell!} d_j^{(\ell)} \right) h^\ell + \mathcal{O}(h^q) \tag{13}$$

with

$$d_j^{(\ell)} := f^{(\ell-1)}(c_j-) - f^{(\ell-1)}(c_j+). \tag{14}$$

Here, the t_j are as defined in Theorem 2. The interpolating pseudo-polynomial $Q_p(h)$ with $p+1 \leq q$ terms again is the same expression, in which I is replaced with its approximation $\widetilde{I}$ and the terms of the ℓ–sum stop with h^p. The interpolation conditions $Q_p\left(\frac{h_0}{2^i}\right) = T_f\left(\frac{h_0}{2^i}\right)$, $i = 0, \ldots, (J+1)p - 1$, lead to the system of $(J+1)p$ equations

$$\mathbf{AHx} = \left[T_f(h_0), T_f\left(\frac{h_0}{2}\right), T_f\left(\frac{h_0}{4}\right), \ldots, T_f\left(\frac{h_0}{2^{(J+1)p-1}}\right) \right]^T \tag{15}$$

in the $(J+1)p$ unknowns $\mathbf{x} = \left[\widetilde{I}, \widetilde{d}_1^{(1)}, \ldots, \widetilde{d}_J^{(1)}, \widetilde{d}_0^{(2)}, \ldots, \widetilde{d}_J^{(2)}, \widetilde{d}_0^{(3)}, \ldots, \widetilde{d}_J^{(3)}, \ldots, \widetilde{d}_0^{(p)}, \ldots, \widetilde{d}_J^{(p)} \right]^T$ with

$$\mathbf{H} := \operatorname{diag}\left(1, h_0, \ldots, h_0, \frac{h_0^2}{2!}, \ldots, \frac{h_0^2}{2!}, \frac{h_0^3}{3!}, \ldots, \frac{h_0^3}{3!}, \ldots, \frac{h_0^p}{p!}, \ldots, \frac{h_0^p}{p!}\right),$$

in which h_0 appears J times and the other powers of h_0 show up $J+1$ times. Again, we only want the first component of the solution $\mathbf{x}$, i.e., the first component of the solution of $\mathbf{Ay} = \mathbf{b}$, where $\mathbf{b}$ is the right-hand side of (15) and the matrix $\mathbf{A}$ is given by

$$\mathbf{A} := \left(\mathbf{A}_1 \middle| \mathbf{A}_2 \middle| \ldots \middle| \mathbf{A}_{p-1} \middle| \mathbf{A}_p \right) \tag{16}$$

with $r = (J+1)p - 1$ rows,

$$\mathbf{A}_1 := \begin{bmatrix} 1 & \gamma_1(h_0)P_1(t_1(h_0)) & \gamma_2(h_0)P_1(t_2(h_0)) & \ldots & \gamma_J(h_0)P_1(t_J(h_0)) \\ 1 & \frac{1}{2}\gamma_1(\frac{h_0}{2})P_1(t_1(\frac{h_0}{2})) & \frac{1}{2}\gamma_2(\frac{h_0}{2})P_1(t_2(\frac{h_0}{2})) & \ldots & \frac{1}{2}\gamma_J(\frac{h_0}{2})P_1(t_J(\frac{h_0}{2})) \\ 1 & \frac{1}{4}\gamma_1(\frac{h_0}{4})P_1(t_1(\frac{h_0}{4})) & \frac{1}{4}\gamma_2(\frac{h_0}{4})P_1(t_2(\frac{h_0}{4})) & \ldots & \frac{1}{4}\gamma_J(\frac{h_0}{4})P_1(t_J(\frac{h_0}{4})) \\ 1 & \frac{1}{8}\gamma_1(\frac{h_0}{8})P_1(t_1(\frac{h_0}{8})) & \frac{1}{8}\gamma_2(\frac{h_0}{8})P_1(t_2(\frac{h_0}{8})) & \ldots & \frac{1}{8}\gamma_J(\frac{h_0}{8})P_1(t_J(\frac{h_0}{8})) \\ \vdots & & & & \\ 1 & \frac{1}{2^r}\gamma_1(\frac{h_0}{2^r})P_1(t_1(\frac{h_0}{2^r})) & \frac{1}{2^r}\gamma_2(\frac{h_0}{2^r})P_1(t_2(\frac{h_0}{2^r})) & \ldots & \frac{1}{2^r}\gamma_J(\frac{h_0}{2^r})P_1(t_J(\frac{h_0}{2^r})) \end{bmatrix}$$

and

$$\mathbf{A}_m := \begin{bmatrix} P_m^*(t_0(h_0)) & P_m(t_1(h_0)) & \ldots & P_m(t_J(h_0)) \\ \frac{1}{2^m}P_m^*(t_0(\frac{h_0}{2})) & \frac{1}{2^m}P_m(t_1(\frac{h_0}{2})) & \ldots & \frac{1}{2^m}P_m(t_J(\frac{h_0}{2})) \\ \frac{1}{4^m}P_m^*(t_0(\frac{h_0}{4})) & \frac{1}{4^m}P_m(t_1(\frac{h_0}{4})) & \ldots & \frac{1}{4^m}P_m(t_J(\frac{h_0}{4})) \\ \frac{1}{8^m}P_m^*(t_0(\frac{h_0}{8})) & \frac{1}{8^m}P_m(t_1(\frac{h_0}{8})) & \ldots & \frac{1}{8^m}P_m(t_J(\frac{h_0}{8})) \\ \vdots & & & \\ \frac{1}{(2^r)^m}P_m^*(t_0(\frac{h_0}{2^r})) & \frac{1}{(2^r)^m}P_m(t_1(\frac{h_0}{2^r})) & \ldots & \frac{1}{(2^r)^m}P_m(t_J(\frac{h_0}{2^r})) \end{bmatrix}, \quad 2 \le m \le p. \tag{17}$$

However, there is a caveat here. Recall that all odd degree Bernoulli polynomials vanish at 0, and that $t_0(z) = 0$ (t_0 as defined in Theorem 2) for all $z = h_0/2^k$, $k \in \mathbb{N}$, in the present case. Therefore, the first column of the mth block $\mathbf{A}_m$ of $\mathbf{A}$ vanishes when m is odd and a "jump" is present at the extremities. (The unknown $\widetilde{d}_0$ is absent as well.) We thus implement the following modification of the process: when the increase of p by one unit leads to an odd number, that first column is erased, and simultaneously one row of the matrix is removed, which means that one less trapezoidal approximation is taken into account in the process (making it, in fact, cheaper). We express this by adding a star to the first column of $\mathbf{A}_m$, to indicate that it is absent if p is odd; then $r < (J+1)p - 1$ if p exceeds two. $\mathbf{H}$ changes as well, as some of its elements are eliminated, but that does not affect the method, as only the first component of $\mathbf{x}$ is needed.

A general algorithm, which includes the case where several sample vectors are available, is given in Section 7.

One of the reviewers expressed doubts about our method's efficiency and suggested to replace f over every interval between two jumps by an approximation constructed from the samples there and to sum the exact integrals of these approximations (which amounts to using an open quadrature rule over every subinterval). We have not compared our method to such a scheme. Another comment suggested to use adaptivity, and implement a special procedure near the discontinuities. This is a natural idea, but would ignore our assumption that only equispaced samples are available in our setting.

6 Examples

6.1 A first example with the jump at the boundary

Here we want to integrate the function g in (11) between 0 and 1. The exact value is

$$I = \frac{1}{\beta} \sin\left(\beta c_1\right) + e^{1-c_1} - 1.$$

With $c_1 = 1/\sqrt{3}$ and $\beta = 2$, $I \approx 0.9833366718258914$; the corresponding g is given in Fig. 2. The jump at the extremities — or “boundary jump” — which appears in the classic Euler–Maclaurin formula of Theorem 1 is now present and the corresponding terms must be taken into account in Formula (4).

The extrapolation method described in the previous section yields Table 2, in the caption of which it is called “first version,” to distinguish it from the second version introduced after the table. Its first column again displays the values h_k, the second gives $100 * t_1(h_k)$, the relative location of the jump with respect to the next node, in percent. The third column shows the errors of the trapezoidal value, the fourth the degree p of the extrapolation using h_k, and the last column lists the errors of the extrapolated value computed as the first component of the solution of $\mathbf{Ay} = \mathbf{b}$. A missing value in that last column points to the fact that no extrapolated value is computed after the first new trapezoidal value, when two are needed to interpolate a h^p-term, i.e., p is even.

The abovementioned elimination of one column and simultaneous non-computation of a trapezoidal value have the consequence that the values used in the

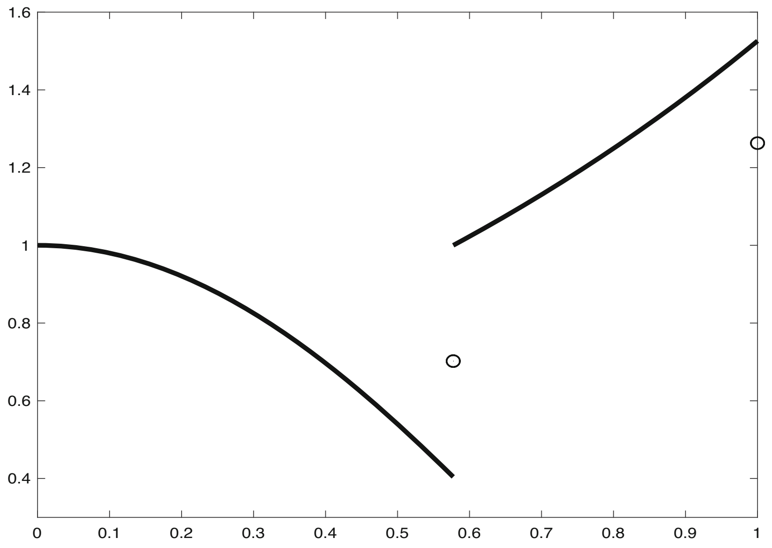

Fig. 2 Function g from (11) with $c_1 = 1/\sqrt{3}$ and $\beta = 2$, as in Fig. 1

Table 2 Example with one interior jump (and the boundary jump), first version

h_k	$100 * t_1(h_k)$	$T_f(h_k) - I$	p	$\widetilde{I} - I$
1	42.26	2.7966e − 01		
1/2	84.53	−8.1686e − 02	1	1.6786e − 01
1/4	69.06	−1.6003e − 02		
1/8	38.12	1.2342e − 02	2	8.7576e − 05
1/16	76.24	−9.1893e − 03	3	−2.0941e − 05
1/32	52.48	−2.2304e − 04		
1/64	49.58	4.1829e − 03	4	7.4957e − 09
1/128	99.17	1.8669e − 03	5	−1.3005e − 09
1/256	19.83	7.0387e − 04		
1/512	39.67	1.2113e − 04	6	−5.5511e − 16
1/1024	79.33	−1.7055e − 04	7	−1.9129e − 13
1/2048	58.66	−2.5152e − 05		
1/4096	17.33	4.7528e − 05	8	−4.4409e − 16
1/8192	34.66	1.1160e − 05	9	−2.4425e − 15

next extrapolation step do not improve in accuracy as much as for even p. *We have therefore repeated the same computations, but by dividing the stepsize h by four when extrapolation is not performed* (with corresponding modification of the matrix $\mathbf{A}$, i.e., a multiplication by $1/4$ instead of $1/2$ of the fraction in front of γ_1 in the second column, and correspondingly in the next columns). The results with this "second version" of the method are given in Table 3. Fewer trapezoidal values are now used for extrapolation for about the same accuracy (11 instead of 14 in this example), which decreases the size of the system of equations.

Table 3 Same as Table 2, but second version

h_k	$100 * t_1(h_k)$	$T_f(h_k) - I$	p	$\widetilde{I} - I$
1	42.26	2.7966e − 01		
1/2	84.53	−8.1686e − 02	1	1.6786e − 01
1/4	69.06	−1.6003e − 02		
1/8	38.12	1.2342e − 02	2	8.7576e − 05
1/32	52.48	−2.2304e − 04	3	−1.8929e − 06
1/64	49.58	4.1829e − 03		
1/128	99.17	1.8669e − 03	4	1.1806e − 08
1/512	39.67	1.2113e − 04	5	−5.0959e − 14
1/1024	79.33	−1.7055e − 04		
1/2048	58.66	−2.5152e − 05	6	−8.8818e − 16
1/8192	34.66	1.1160e − 05	7	−1.4433e − 15

6.2 Another function with one jump

To confirm the above results, we have tried another function with one jump, namely

$$f(x) := \begin{cases} \cos(4x), & -1 \le x < c_1, \\ \big(\cos(4x) + \sin(2.5x)\big)/2, & x = c_1, \\ \sin(2.5x), & c_1 < x \le 3. \end{cases} \tag{18}$$

The exact value is $\int_{-1}^{3} f(x)dx = \big(\sin(4c_1) + \sin(4)\big)/4 + \big(\cos(2.5c_1) - \cos(2.5)\big)/2.5$. We set $c_1 = 1/30$; the corresponding function is plotted in Fig. 3 and the results with the first version, i.e., the division of h by two at every computation of a new $T_f(h)$, are displayed in Table 4. (From here on, we refrain from giving the value of p, which is just the number of the extrapolation step, i.e., of the row in the last column.)

As in Table 1, we encounter a phenomenon that arose in several of our examples: the extrapolated value can be extremely large before h_k becomes small enough (or enough trapezoidal values are computed). But this is not terribly important, as we need small enough h_k to get improvement over the trapezoidal values.

The results also confirm that bad intermediate extrapolated values do not affect the quality of the subsequent approximations $\widetilde{I}$: the only numbers used in performing the extrapolation are the trapezoidal values. This illustrates our comments about the Neville tableau in Section 4.

We stopped with the last h_k given in the table, since the difference between two consecutive extrapolated values was smaller than our tolerance of $N * 10^{-16}$. We then went up to $p = 15$: the accuracy remained below $5 \cdot 10^{-13}$, despite condition numbers of $\mathbf{A}$ larger than 10^{20}.

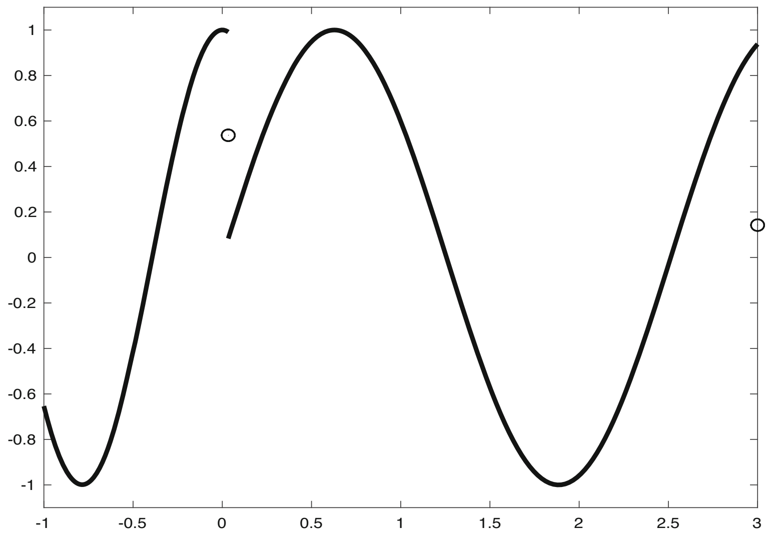

Fig. 3 Function f from (18) with $c_1 = 1/30$

Table 4 Second example with one interior jump, first version

$h_k/(b-a)$	$100 * t_1(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	74.17	4.6472e − 01	
1/2	48.33	1.3773e + 00	1.3469e + 00
1/4	96.67	6.7773e − 01	
1/8	93.33	2.5092e − 01	−4.6420e + 14
1/16	86.67	1.0063e − 01	1.5684e − 02
1/32	73.33	3.2481e − 02	
1/64	46.67	−1.4205e − 04	3.1060e + 08
1/128	93.33	1.2459e − 02	−1.0385e + 07
1/256	86.67	5.2622e − 03	
1/512	73.33	1.6775e − 03	6.7007e − 07
1/1024	46.67	−1.1137e − 05	−2.6336e − 01
1/2048	93.33	7.6903e − 04	
1/4096	86.67	3.2533e − 04	−6.2422e − 14
1/8192	73.33	1.0353e − 04	−4.5927e − 13

These huge condition numbers impact the results only for relatively large h_k. The condition number is, of course, only an upper bound on the amplification of round-off errors in solving linear equations. Whether actual amplification occurs depends on how the right hand side **b** and perturbations to it are aligned with the singular vectors of the matrix. In many practical applications, numerical solutions behave much better than predicted by condition numbers.

In our case, because of the jumps, the first components of **b** behave eratically, and it is not surprising that sometimes the solution of the system suffers gravely from the ill-conditioning. When the abscissae h_k of the last $T_f(h_k)$ cluster close to the evaluation point 0, these trapezoidal values become much more regular, which diminishes the likelihood of a bad **b**. For a generalized Richardson method close to ours, Gaussian elimination has been shown in [18, pp. 73–76] to be stable. All the above likely explains the excellent behavior of our scheme for p large enough.

We then solved the same problem, but, as with the former example, by dividing the stepsize h by four when extrapolation is not performed. The results with that second version appear in Table 5.

When h_k is such that the jump is close to a node, the trapezoidal rule usually performs worse than with h_{k-1}. Nevertheless, the extrapolated values are much more regular than in the first version. We have therefore considered only this second version from here on.

What if, after some steps, the jump coincides with a node? In fact, this does not cause a problem: as the first elements of the corresponding column do not vanish, the matrix **A** remains (mathematically) nonsingular (except for the case where the jump coincides with a node from the onset, but then Romberg may be applied from the start). To demonstrate this, we have repeated the calculations for the last example, but

Table 5 Same as Table 4, but second version

$h_k/(b-a)$	$100 * t_1(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	74.17	$4.6472e-01$	
1/2	48.33	$1.3773e+00$	$1.3469e+00$
1/4	96.67	$6.7773e-01$	
1/8	93.33	$2.5092e-014$	$-4.4620e+14$
1/32	73.33	$3.2481e-02$	$4.8516e-03$
1/64	46.67	$-1.4205e-04$	
1/128	93.33	$1.2459e-02$	$-1.0787e-04$
1/512	73.33	$1.6775e-03$	$-1.9432e-06$
1/1024	46.67	$-1.1137e-04$	
1/2048	93.33	$7.6903e-04$	$3.2082e-11$
1/8192	73.33	$1.0353e-05$	$1.2349e-11$
1/16384	46.67	$-7.3617e-06$	
1/32768	93.33	$4.8027e-05$	$-1.2351e-15$
1/131072	73.33	$6.4652e-06$	$-5.8106e-14$

this time with $c_1 = 1/32$. The results are listed in Table 6: from $n = 128$ on, when c_1 is a node, the trapezoidal values become much more regular (in fact $O(h^2)$ according to the Euler–Maclaurin formula) and the extrapolated values rapidly converge within our tolerance $N * 10^{-16}$.

Table 6 Second example with one interior jump at c_1, with c_1 becoming a node

$h_k/(b-a)$	$100 * t_1(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	74.22	$4.6662e-01$	
1/2	48.44	$1.3792e+00$	$1.3507e+00$
1/4	96.88	$6.7963e-01$	
1/8	93.75	$2.5282e-01$	$-2.3533e+15$
1/16	87.5	$1.0252e-01$	$1.6122e-02$
1/32	75.	$3.4379e-02$	
1/64	50.	$1.7559e-03$	$7.4550e+08$
1/128	0.	$7.3504e-05$	$-1.4381e+09$
1/256	0.	$1.8368e-05$	
1/512	0.	$4.5917e-06$	$-3.9274e-13$
1/1024	0.	$1.1479e-06$	$-2.5814e-13$
1/2048	0.	$2.8697e-07$	
1/4096	0.	$7.1742e-08$	$-4.0704e-14$
1/8192	0.	$1.7936e-08$	$-3.5472e-14$

6.3 An example with two jumps

As a specific example with two jumps, we have taken

$$f(x) := \begin{cases} \cos(4x), & -1 \le x < c_1, \\ \big(\cos(4x) + \sin(2.5x)\big)/2, & x = c_1, \\ \sin(2.5x), & c_1 < x < c_2, \\ (\sin(2.5x) + e^{(x-c_2)})/2 & x = c_2, \\ e^{(x-c_2)}, & c_2 < x \le 3, \end{cases} \tag{19}$$

and we have again integrated from -1 to 3. The exact value is $\big(\sin(4c_1)+\sin(4)\big)/4+\big(\cos(2.5c_1) - \cos(2.5c_2)\big)/2.5 + e^{(3-c_2)} - 1$.

The function with $c_1 = 1/30$ and $c_2 = \sqrt{3}$ is shown in Fig. 4; the exact value becomes $I = 2.945411417434258$ and our results are given in Table 7. The number of trapezoidal values necessary for an extrapolation is not always three, the number of jumps, since for odd p only two such values are required: as in Section 5, $P_m(t_0(h/2^k)) = 0$ for all k, so that $d_0^{(\ell)}$ does not appear in (13) and therefore needs not be determined.

The first extrapolated values are extremely large, in fact about the reciprocal of the unit round-off. This results from the extremely bad conditioning of the matrix $\mathbf{A}$ and, at first, even led us to believe that there was a programming error in our code. To test this, we changed c_2 from $\sqrt{3}$ to $1 + \sqrt{3}$. The results are given in Table 8.

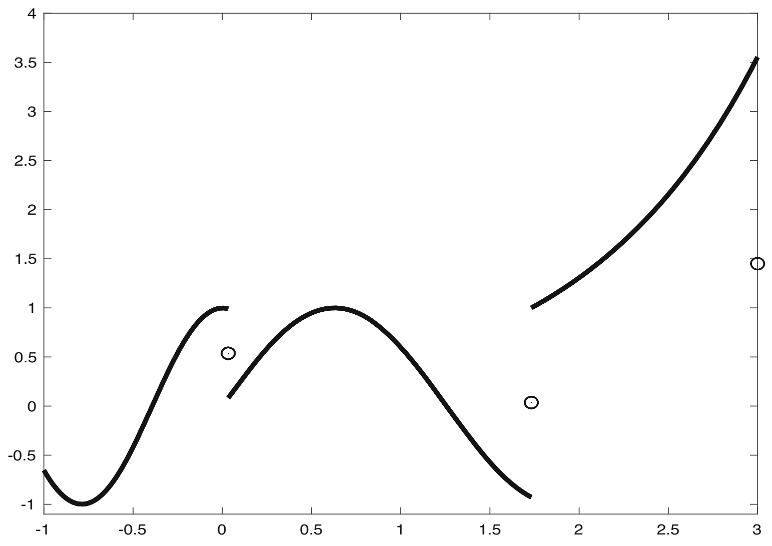

Fig. 4 Function f from (19) with $c_1 = 1/30$ and $c_2 = \sqrt{3}$

Table 7 Example (19) with two interior jumps

$h_k/(b-a)$	$100 * t_2(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	31.70	2.8544e + 00	
1/2	63.40	1.1514e + 00	
1/4	26.79	1.4103e + 00	−1.2524e + 15
1/8	53.59	2.9075e − 01	
1/16	7.18	3.1524e − 01	
1/32	14.36	1.2127e − 01	6.4842e + 13
1/128	57.44	8.2702e − 03	
1/256	14.87	1.5888e − 02	−5.7300e − 04
1/512	29.75	4.7436e − 03	
1/1024	59.50	−8.2221e − 04	
1/2048	19.00	1.9371e − 03	3.7371e − 06
1/8192	75.99	−1.4111e − 04	
1/16384	51.99	−1.6706e − 05	2.2564e − 11
1/32768	3.98	1.5633e − 04	
1/65536	7.96	6.9790e − 05	
1/131072	15.91	2.6519e − 05	−1.5543e − 14

Table 8 Same as Table 7, but with $c_2 = 1 + \sqrt{3}$ instead of $\sqrt{3}$

$h_k/(b-a)$	$100 * t_2(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	6.70	1.0990e + 00	
1/2	13.40	1.6423e + 00	
1/4	26.79	7.5809e − 01	1.1553e + 00
1/8	53.59	2.3895e − 01	
1/16	7.18	1.5756e − 01	
1/32	14.36	5.4763e − 02	−1.9864e − 02
1/128	57.44	1.1337e − 02	
1/256	14.87	7.9106e − 03	−5.2911e − 04
1/512	29.75	2.4376e − 03	
1/1024	59.50	−2.8953e − 04	
1/2048	19.00	1.0598e − 03	3.9057e − 07
1/8192	75.99	4.2617e − 05	
1/16384	51.99	−9.6930e − 06	−5.9827e − 11
1/32768	3.98	7.4990e − 05	
1/65536	7.96	3.2635e − 05	
1/131072	15.91	1.1458e − 05	−1.8568e − 14

7 Extension to several sample vectors

So far we have started with the trapezoidal value corresponding to the interval $[a, b]$ and then recursively divided all intervals in two, or four in the second version when p is odd, thereby doubling (resp. quadrupling) the number of function values at each step. Equivalently, being given the vector of samples corresponding to the last subdivision, we compute all trapezoidal values and solve the system of equations $\mathbf{Ay} = \mathbf{b}$ for the first component of $\mathbf{y}$. The fact that this uses only one vector of samples may be an important advantage of the method.

However, such doubling of the number of function values rapidly results in a very large number of these. For that reason, it has been suggested long ago to alternate powers of two multiples of 2 and 3 in the extrapolation process: the so-called *Bulirsch sequence* [9] of interval numbers, with $n_k = 2n_{k-2}$ for $k \geq 4$, is

$$(1,)2, 3, 4, 6, 8, 12, 16, 24, 32, 48, \ldots\ldots \tag{20}$$

(the sequence is monotonically increasing). This requires two sample vectors, one with $2 \cdot 2^\ell + 1$ and the other with $3 \cdot 2^\ell + 1$ samples.

One may generalize this to several vectors of samples, say q of them, the sth one with $(2s - 1)2^\ell + 1$ components, $s = 1, 2, \ldots, q$, i.e., the first q odd numbers times the first ℓ powers of two. (One could consider taking primes times 2^ℓ to have only different values of f in the interior of the interval, but with $q \leq 4$ this is the same.) We do not know whether it matters, but we have ordered the numbers of intervals. For $q = 3$, and ignoring the possible first element 1, this yields the following sequences of sample indices:

2	4	8	16	32	64	128	256	...
3	6	12	24	48	96	192	384	...
5	10	20	40	80	160	320	640	...

Ordering these numbers turns out to be very simple: it suffices to read them one anti-diagonal after the other, to obtain

$$(1,)2, 3, 4, 5, 6, 8, 10, 12, 16, 20, 24, 32, 40, 48, 64, \ldots \tag{21}$$

or

$$n_k = 2n_{k-3}, \quad k \geq 6. \tag{22}$$

It does not seem as simple for $q = 4$.

These numbers indeed grow slowly and may give trapezoidal approximations too inaccurate to yield a precise extrapolate. Compared with the harmonic sequence, which is the best in the context of extrapolation methods for ODEs [9, p. 221], the sequence grows somewhat faster, but requires only three sample vectors whereas the harmonic sequence requires an unboundedly increasing number of them.

Table 9 Example (19), same as Table 7 but now with $3 \cdot 2^{\ell}$ intervals

$h_k/(b-a)$	$100 * t_2(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	31.70	2.8544e + 00	
1/3	95.10	−1.1649e + 00	
1/6	90.19	−2.8305e − 01	−2.5987e + 14
1/12	80.38	−2.4552e − 02	
1/24	60.77	2.8195e − 02	
1/48	21.54	5.7945e − 02	−1.4786e + 12
1/192	86.16	−1.6147e − 02	
1/384	72.31	−1.5815e − 03	1.9585e − 04
1/768	44.62	1.0326e − 03	
1/1536	89.25	−2.6765e − 03	
1/3072	78.50	−8.3245e − 04	1.4068e − 07
1/12288	13.99	2.5559e − 04	
1/24576	27.98	2.4766e − 05	−2.4951e − 11
1/49152	55.97	−1.6746e − 05	
1/98304	11.94	4.0943e − 05	
1/196608	23.87	1.2096e − 05	−2.6645e − 15

The system of equations to be solved for the $\widetilde{d}_j^{(\ell)}$ is like (15), but now with a modified matrix $\mathbf{A}_m$

$$\mathbf{A}_m := \begin{bmatrix} \frac{1}{n_1^m} P_m^*(t_0(\frac{h_0}{n_1})) & \frac{1}{n_1^m} P_m(t_1(\frac{h_0}{n_1})) & \dots & \frac{1}{n_1^m} P_m(t_J(\frac{h_0}{n_1})) \\ \frac{1}{n_2^m} P_m^*(t_0(\frac{h_0}{n_2})) & \frac{1}{n_2^m} P_m(t_1(\frac{h_0}{n_2})) & \dots & \frac{1}{n_2^m} P_m(t_J(\frac{h_0}{n_2})) \\ \frac{1}{n_3^m} P_m^*(t_0(\frac{h_0}{n_3})) & \frac{1}{n_3^m} P_m(t_1(\frac{h_0}{n_3})) & \dots & \frac{1}{n_3^m} P_m(t_J(\frac{h_0}{n_3})) \\ \frac{1}{n_4^m} P_m^*(t_0(\frac{h_0}{n_4})) & \frac{1}{n_4^m} P_m(t_1(\frac{h_0}{n_4})) & \dots & \frac{1}{n_4^m} P_m(t_J(\frac{h_0}{n_4})) \\ \vdots & & & \\ \frac{1}{n_r^m} P_m^*(t_0(\frac{h_0}{n_r})) & \frac{1}{n_r^m} P_m(t_1(\frac{h_0}{n_r})) & \dots & \frac{1}{n_r^m} P_m(t_J(\frac{h_0}{n_r})) \end{bmatrix}. \qquad (23)$$

The general algorithm for determining an extrapolated value $\widetilde{I}$ may now be described as follows:

- obtain one (or several) sample vector(s) with N_1+1, resp. N_1+1, $N_2+1 \dots N_q+1$ samples; the numbers of intervals N_r should be composite numbers, so that several trapezoidal values can be computed from any one of the vectors;
- determine the sequence $\{n_\ell + 1\}$ of numbers of samples from which the trapezoidal values will be computed $\big($for instance (20) or (21)$\big)$;
- compute the corresponding vector $\mathbf{b}$ of the trapezoidal values; after every step with no extrapolation, a value of n_ℓ (if enough values are available) should

Table 10 Example (19), same as Table 7, but with two sample vectors

$h_k/(b-a)$	$100 * t_2(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1	31.70	2.8544e + 00	
1/2	63.40	1.1514e + 00	
1/3	95.10	−1.1649e + 00	−8.3146e + 15
1/4	26.79	1.4103e + 00	
1/6	90.19	−2.8305e − 01	
1/8	53.59	2.9075e − 01	1.6734e + 15
1/16	7.18	3.1524e − 01	
1/24	60.77	2.8195e − 02	4.0700e + 13
1/32	14.36	1.2127e − 01	
1/48	21.54	5.7945e − 02	
1/64	28.72	2.6526e − 02	−2.5154e − 03
1/128	57.44	8.2702e − 03	
1/192	86.16	−1.6147e − 02	−8.5729e − 04
1/256	14.88	1.5888e − 02	
1/384	72.31	−1.5815e − 03	
1/512	29.75	4.7436e − 03	−5.3350e − 09
1/1024	59.50	−8.2221e − 04	
1/1536	89.25	−2.6765e − 03	1.7583e − 09
1/2048	19.00	1.9371e − 03	
1/3072	78.50	−8.3245e − 04	
1/4096	38.00	5.5157e − 04	2.2249e − 13
1/8192	76.00	−1.4111e − 04	
1/12288	13.99	2.5559e − 04	1.3145e − 13
1/16384	51.99	−1.6706e − 05	
1/24576	27.98	2.4766e − 05	
1/32768	3.98	1.5633e − 04	−9.3703e − 14

be skipped to improve accuracy (this is the more accurate "second method" suggested in Section 5);

- compute the matrix $\mathbf{A}$ from (15) with the submatrices $\mathbf{A}_m$ from (23);
- solve the system of equations $\mathbf{Ay} = \mathbf{b}$ for the first component y_1; the latter is the sought extrapolated value $\widetilde{I}$.

Before experimenting with several sample vectors, we have computed with only the second vector of the above table, i.e., $n_k = 3 \cdot 2^k$, $k = 0, 1, 2, 3, 4, \ldots$, which can be accomplished by a minor change in our program. For the same example as in Section 6.3, Table 9 then replaces Table 7.

This change does not look like a good idea: to obtain, say, an accuracy of 10^{-11}, about 25'000 points are necessary, compared to about 16'000 with the sequence 2^{ℓ}.

Table 11 Example (19), same as Table 7, but with three sample vectors

$h_k/(b-a)$	$100 * t_2(h_k)$	$T_f(h_k) - I$	$\widetilde{I} - I$
1/3	95.10	$-1.1649e+00$	
1/4	26.79	$1.4103e+00$	
1/5	58.49	$5.6667e-01$	$-8.1725e-01$
1/6	90.19	$-2.8305e-01$	
1/8	53.59	$2.9075e-01$	
1/10	16.99	$3.2776e-01$	$-2.8631e-03$
1/16	7.18	$3.1524e-01$	
1/20	33.97	$1.4608e-01$	$-7.3641e-03$
1/24	60.77	$2.8195e-02$	
1/32	14.36	$1.2127e-01$	
1/40	67.95	$-1.2582e-02$	$1.0491e-05$
1/64	28.72	$2.6526e-02$	
1/80	35.90	$7.7531e-03$	$-8.5575e-05$
1/96	43.08	$-4.7299e-03$	
1/128	57.44	$8.2702e-03$	
1/160	71.80	$-6.3044e-03$	$-5.3723e-07$
1/256	14.88	$1.5888e-02$	
1/320	43.59	$-2.3700e-04$	$2.5267e-08$
1/384	72.31	$-1.5815e-03$	
1/512	29.75	$4.7436e-03$	
1/640	87.19	$-3.5123e-03$	$-2.7220e-11$
1/1024	59.50	$-8.2221e-04$	
1/1280	74.37	$-1.9349e-03$	$-1.7808e-12$
1/1536	89.25	$-2.6765e-03$	
1/2048	19.00	$1.9371e-03$	
1/2560	48.75	$2.7587e-04$	$-8.6731e-13$
1/4096	38.00	$5.5157e-04$	
1/5120	97.50	$-8.3326e-04$	$5.8265e-13$
1/6144	57.00	$8.9776e-05$	
1/8192	76.00	$-1.4111e-04$	
1/10240	94.99	$-2.7964e-04$	$6.7946e-13$

7.1 An example with two sample vectors

To test the method with several sample vectors, we started with two of them, i.e., with the Bulirsch sequence (20). Our results, still with the same example as in Section 6.3, are given in Table 10.

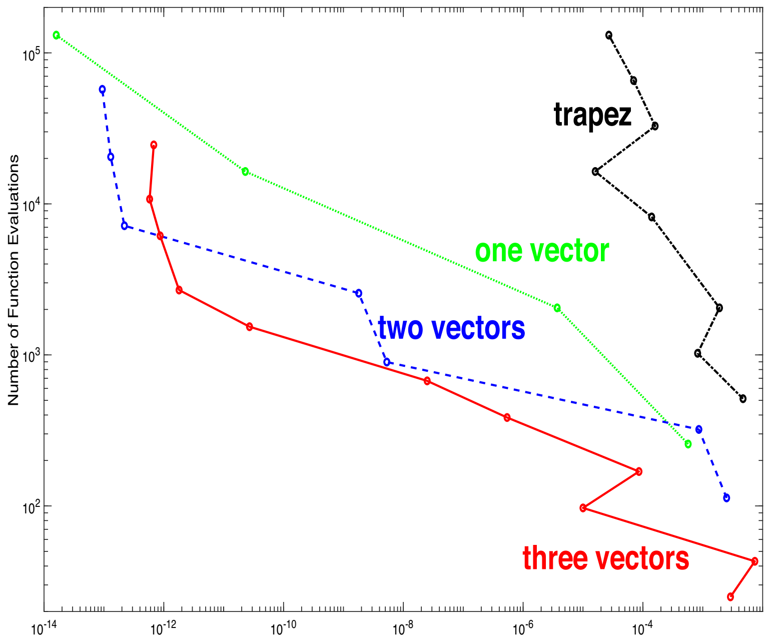

Fig. 5 Comparison of the number of function evaluations needed for a given accuracy with one, two, and three sample vectors

The availability of a second vector of samples is very effective in terms of function evaluations: despite trapezoidal values all in error by more than 10^{-3}, the extrapolation scheme reaches an accuracy better than 10^{-8} with $512+1+384-1 = 896$ values of f, while it does not even reach 10^{-6} with a single vector of 2049 values (Table 7)!

7.2 An example with three sample vectors

In example (19), $J = 2$ and thus three trapezoidal values are needed for the first extrapolation. In order to avoid computing the same function values twice and to use the formula giving $T(h/2)$ from $T(h)$ [17, p. 377], and in view of (22), we started the method with $h = (b-a)/3$, so that the next two interval lengths are $h = (b-a)/4$ and $h = (b-a)/5$. The results are displayed in Table 11.

Here, the extrapolated value is with $h_k = h_0/10$ already better than the plain trapezoidal value, an accuracy of $2.53 \cdot 10^{-8}$ is attained with 320 + 256 + 96 – 1 = 671 function evaluations, and a higher precision than 10^{-10} with 640 + 512 + 340 – 1 = 1491 evaluations, while working with two sample vectors requires 1536 + 1024 = 2060 evaluations for an accuracy of $1.76 \cdot 10^{-9}$.

With the data of Tables 7, 10, and 11, we finally plot in Fig. 5 the number of function evaluations needed to reach a certain accuracy, starting from 10^{-2}, in green and dotted after extrapolation with one vector of samples, in blue and dashed with two, and in red and solid with three (the curves are intended to be read from right to left); the plain trapezoidal values are added in black and dash-dotted for comparison. Note the logarithmic scale!

8 Conclusions

The direct application of the composite trapezoidal rule to a piecewise smooth function f converges slowly and in a non-monotone fashion to the integral of f. The extrapolation method presented in this work accelerates the convergence, and makes it regular as soon as the interval length h is small enough. It converged in all examples we tried, despite obtaining sometimes very bad values at the initial stage. The number of correct digits increased by a factor of two to three compared with the trapezoidal values. One may decrease h without any stability concern, and thus use a stopping criterion based on the agreement of consecutive values of $\widetilde{I}$.

One question is whether better results could be obtained by starting (and extrapolating) from smaller values h_0, to avoid the often erratic first extrapolates. We tried this for the very first example (Section 4): the results were not too encouraging, as large values due to the ill-conditioning of the matrices again arose for small divisors of h_0 and longer sample vectors were needed to reach our tolerance.

Note, finally, that the method requires knowledge of the locations of the jumps. Methods exist for determining them, but most work in a transformation space; we envision one using just the vector(s) of equispaced samples.

Acknowledgements The authors thank Jean-Pierre Gabriel and the reviewers for their comments, which have significantly improved the presentation.

Funding Open access funding provided by the University of Fribourg. The second author was supported by a National Science and Engineering Research Council (NSERC) Canada Discovery grant (RGPIN-2020-04663).

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Berrut J.-P.: Fascinante interpolation. Bull. Soc. Frib. Sc. Nat. **83**, 3–20 (1994)
2. Berrut J.-P.: A circular interpretation of the Euler–Maclaurin formula. J. Comput. Appl. Math. **189**, 375–386 (2006)
3. Berrut J.–P.: The conditioning of a linear barycentric rational interpolant. In: Realization and Model Reduction of Dynamical Systems A Festschrift in Honor of the 70th Birthday of Thanos Antoulas, pp. 23–37. Springer, Berlin (2022)
4. Brezinski, C.: The Mühlbach-Neville-Aitken algorithm and some extensions. BIT **20**, 444–451 (1980)
5. Brezinski, C.: A general extrapolation algorithm. Numer. Math **35**, 175–187 (1980)
6. Brutman L.: Lebesgue functions for polynomial interpolation — a survey. The heritage of P.L.Chebyshev: a Festschrift in honor of the 70th birthday of T.J. Rivlin. Ann. Numer. Math. **4**, 111–127 (1997)
7. Corless R.M., Rafiee Sevyeri L.: The Runge example for interpolation and Wilkinson's examples for rootfinding. SIAM Rev. **62**, 231–243 (2020)
8. Fornberg B.: Improving the accuracy of the trapezoidal rule. SIAM Rev. **63**, 167–180 (2021)
9. Hairer E., Nørsett S.P., Wanner G.: Solving Ordinary Differential Equations I. Nonstiff Problems. Springer, Berlin (1987)
10. Håvie, T.: Generalized Neville type extrapolation schemes. BIT **19**, 204–213 (1979)
11. Håvie, T.: Remarks on a unified theory for classical and generalized interpolation and extrapolation. BIT **21**, 465–474 (1981)
12. Hildebrand F.B.: Introduction to Numerical Analysis. Dover, New York (1987)
13. Kress R.: Numerical Analysis. Springer-Verlag, New York (1998)
14. Lyness J.N.: The calculation of Fourier coefficients by the Möbius inversion of the Poisson summation formula — Part II.Piecewise continuous functions and functions with poles near the interval [0, 1]. Math. Comp. **25**, 59–78 (1971)
15. Mühlbach G.: Neville-Aitken algorithms for interpolation by functions of Čebyšev-systems in the sense of Newton and in a generalized sense of Hermite. In: Law, A., Sahney, B.N. (eds.) Proceedings of a Conference, University Calgary, 1975, pp. 180–199, Academic Press, New York (1976)
16. Schneider C.: Vereinfachte Rekursionen zur Richardson-Extrapolation in Spezialfällen. Numer. Math **24**, 177–184 (1975)
17. Schwarz H.R.: Numerische Mathematik 4te Aufl., Teubner, 1997; English Translation of the Second Edition: Numerical Analysis. A Comprehensive Introduction, Wiley, New York (1989)
18. Sidi A.: Practical Extrapolation Methods: Theory and Applications. Cambridge University Press, Cambridge (2003)
19. Trefethen L.N., Weideman J.A.C.: The exponentially convergent trapezoidal rule. SIAM Rev. **56**, 385–458 (2014)
20. Trefethen, L.N.: Approximation Theory and Approximation Practice. SIAM (2020)