Results in Nonlinear Analysis 2 (2019) No. 2, 48–60
Available online at www.resultsinnonlinearanalysis.com

Results in Nonlinear Analysis
ISSN 2636-7556

Peer Reviewed Scientific Journal

A close look at Newton–Cotes integration rules

Emre Sermutlu[a]

[a] Çankaya University, Department of Mathematics, 06790, Etimesgut - Ankara, Turkey

Abstract

Newton–Cotes integration rules are the simplest methods in numerical integration. The main advantage of using these rules in quadrature software is ease of programming. In practice, only the lower orders are implemented or tested, because of the negative coefficients of higher orders. Most textbooks state it is not necessary to go beyond Boole's 5-point rule. Explicit coefficients and error terms for higher orders are seldom given literature. Higher-order rules include negative coefficients therefore roundoff error increases while truncation error decreases as we increase the number of points. But is the optimal one really Simpson or Boole?

In this paper, we list coefficients up to 19-points for both open and closed rules, derive the error terms using an elementary and intuitive method, and test the rules on a battery of functions to find the optimum all-round one.

Keywords: quadrature, Newton–Cotes, truncation Error, MATLAB
2010 MSC: 65D30,65D32.

1. Introduction

If we use polynomial interpolation on equally spaced points to integrate a function numerically, we obtain Newton–Cotes rules. These are probably the easiest to apply in practice, if not the most efficient.

High order Newton–Cotes rules are not very popular, because they contain negative coefficients and therefore cause roundoff errors. In practice, few implementations go beyond Simpson's ($n = 2$) or Boole's ($n = 4$) rules.

In a previous paper [1] we have compared closed Newton–Cotes and Gaussian integration rules and demonstrated the superiority of Gaussian rules. There, we tested Newton–Cotes rules up to $n = 6$ and observed that higher orders gave better results. Yet it is well known that for some integrands, for example $f(x) = \dfrac{1}{1+25x^2}$, in the limit $n \to \infty$, the numerical results do not converge to the exact value. [2] In other words, going to much higher orders is not always a good idea.

Email address: `sermutlu@cankaya.edu.tr` (Emre Sermutlu)

Received April 22, 2019 Accepted May 8, 2019, Online May 10, 2019.

Research on improvement, modification and analysis of Newton-Cotes methods continues. Mikkawy has derived coefficients [3] and truncation errors [4] starting with the Lagrange interpolating polynomial. Here, we use the alternative method of undetermined coefficients. Dehghan et. al. have improved these rules considerably by treating the endpoints as variables. [5],[6]. A very interesting application on the integration of some physical problems is done by Simos, [7] where trigonometrically–fitted symplectic methods based on the closed Newton–Cotes formulae are constructed.

In this paper, we aim to determine the "best" Newton–Cotes rule, by considering both open and closed rules up to 19 points. Truncation error decreases and roundoff error increases with increasing order, therefore we expect an optimum order. Although this optimum may be dependent on the integrand, we may obtain some general recommendations for programmers on which rule to use.

We need the weight coefficients beyond the ones supplied in standard reference works like [8], so we derive necessary equations and list the coefficients in the next section. We also give an explicit formula for the truncation error.

2. Coefficients of Newton–Cotes

The problem is to determine the coefficients w_i in the formula

$$\int_a^b f(x)\,dx \approx (b-a)\sum_{i=0}^{n} w_i f(a+ih), \tag{2.1}$$

where n is the number of subintervals and $h=(b-a)/n$ is the step size. We stipulate the formula to be exact for polynomials up to order n. It can easily be showed that if such a formula applies to an interval, it applies to any interval, therefore we will prefer $[-1,1]$ for simplicity. Then,

$$\int_{-1}^{1} x^k\,dx = 2\sum_{i=0}^{n} w_i\left(-1+\frac{2i}{n}\right)^k, \quad k=0,\,1,\,\ldots,\,n. \tag{2.2}$$

If k is odd, the result of the integral is zero, and this condition can be satisfied by choosing symmetric coefficients $w_i = w_{n-i}$. This means that when n even, the resulting formula is exact for x^{n+1} too as a bonus. As we have the same accuracy with smaller number of function evaluations, we will only work with even number of intervals, or equivalently, odd number of points. In this case, $n=2m$ and we can rewrite equation (2.2) in simpler terms for the interval $[-mh,\,mh]$:

$$\int_{-mh}^{mh} x^k\,dx = 2mh\sum_{i=0}^{2m} w_i\,(-mh+ih)^k\,, \quad k=0,\,1,\,\ldots,\,2m. \tag{2.3}$$

Using the symmetry condition $w_i = w_{n-i}$, we obtain:

$$\begin{aligned} &\sum_{i=0}^{m-1}(m-i)^k w_i = \frac{m^k}{2(k+1)}, \quad k=2,\,4,\,6,\,\ldots,\,2m,\\ &w_m + 2\sum_{i=0}^{m-1} w_i = 1. \end{aligned} \tag{2.4}$$

There are $m+1$ linear equations for the unknowns w_i in (2.4). An explicit solution is provided in [3]. We prefer to solve this set on MATLAB, but care should be exercised. For high m values, the solution will not be accurate due to roundoff errors. We have used a special code that uses integer arithmetic to solve these systems.

For the open Newton–Cotes formulae, the analysis is very similar, therefore we only state the results. The integration formula:

$$\int_a^b f(x)\,dx \approx (b-a)\sum_{i=1}^{n-1} w_i f(a+ih), \quad h=(b-a)/n, \tag{2.5}$$

is exact for polynomials up to degree $n-1$ for $n=2m$. Once again we choose $w_i = w_{n-i}$. The coefficients for this case can be obtained from:

$$\begin{aligned} &\sum_{i=1}^{m-1}(m-i)^k w_i = \frac{m^k}{2(k+1)}, \quad k=2,\,4,\,6,\,\ldots,\,2m-2,\\ &w_m + 2\sum_{i=1}^{m-1} w_i = 1. \end{aligned} \tag{2.6}$$

3. Truncation Error of Newton–Cotes

The error terms for the Newton-Cotes methods have been analyzed in detail in [9],[10]. While it is customary to use the Lagrange interpolating polynomial to derive the truncation error for Newton–Cotes formulas, here we give a simpler and more intuitive method based on Taylor series expansion. We obtain an explicit formula at the end. Let us approximate the integral

$$I = \int_{x_0}^{x_n} f(x)\,dx, \tag{3.1}$$

using closed Newton–Cotes rule of order n. The Taylor series of f around x_0 together with the remainder term is:

$$\begin{aligned} I = \int_{x_0}^{x_n} \Bigg(& f(x_0) + f'(x_0)(x-x_0) + \cdots + \frac{f^{(n+1)}(x_0)}{(n+1)!}(x-x_0)^{n+1} \\ & + \frac{f^{(n+2)}(\xi(x))}{(n+2)!}(x-x_0)^{n+2}\Bigg)\,dx. \end{aligned} \tag{3.2}$$

Since our formula is exact up to polynomials of order $n+1$, the source of the error is the last term:

$$I_1 = \int_{x_0}^{x_n} \frac{f^{(n+2)}(\xi(x))}{(n+2)!}(x-x_0)^{n+2}\,dx. \tag{3.3}$$

The factor $(x-x_0)^{n+2}$ does not change sign in the interval $[x_0, x_n]$. Therefore we can use the weighted mean value theorem for integrals to obtain:

$$I_1 = \frac{f^{(n+2)}(\xi)}{(n+2)!}\int_{x_0}^{x_n}(x-x_0)^{n+2}\,dx, \quad \xi \in [x_0,\,x_n]. \tag{3.4}$$

The difference between this and our approximation will give the truncation error. So the main idea is, for example, to approximate x^{10} using powers $1,\,x,\,\ldots,\,x^8$.

$$\begin{aligned}
\text{Error} &= I_{\text{exact}} - I_{\text{approximate}} \\
&= \frac{f^{(n+2)}(\xi)}{(n+2)!}\left(\int_0^{nh} u^{n+2}\,du - nh\cdot\sum_{i=0}^{n} w_i (ih)^{n+2}\right) \\
&= \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!}\left(\frac{n^{n+3}}{n+3} - n\cdot\sum_{i=0}^{n} w_i (i)^{n+2}\right) \\
&= -K_n h^{n+3} f^{(n+2)}(\xi).
\end{aligned} \tag{3.5}$$

We can summarize our results as:

$$\int_{x_0}^{x_n} f(x)\,dx = nh\sum_{i=0}^{n} w_i f(x_0+ih) - K_n h^{n+3} f^{(n+2)}(\xi), \tag{3.6}$$

where $h = (x_n - x_0)/n, \quad \xi \in [x_0, x_n]$ and

$$K_n = \frac{1}{(n+3)!}\left(n(n+3)\sum_{i=0}^{n} w_i\, i^{n+2} - n^{n+3}\right). \tag{3.7}$$

After similar steps for open Newton–Cotes formulas, we find:

$$\int_{x_0}^{x_n} f(x)\,dx = nh\sum_{i=1}^{n-1} w_i f(x_0+ih) + L_n h^{n+1} f^{(n)}(\xi), \tag{3.8}$$

where

$$L_n = \frac{1}{(n+1)!}\left(n(n+1)\sum_{i=1}^{n-1} w_i\, i^{n} - n^{n+1}\right). \tag{3.9}$$

4. Coefficients

We list all coefficients and the truncation errors explicitly up to 19 points ($n = 18$ for closed and $n = 20$ for open Newton–Cotes rules) for easy reference in Tables (1-2):

A glance at the Figures (1-2) is enough to show that going to much higher orders is not a good idea for numerical integration. We encounter negative coefficients after a certain point, and although the sum of all coefficients is 1, their magnitude is increasing. We are subtracting numbers very close in magnitude, therefore we expect to face problems due to roundoff error.

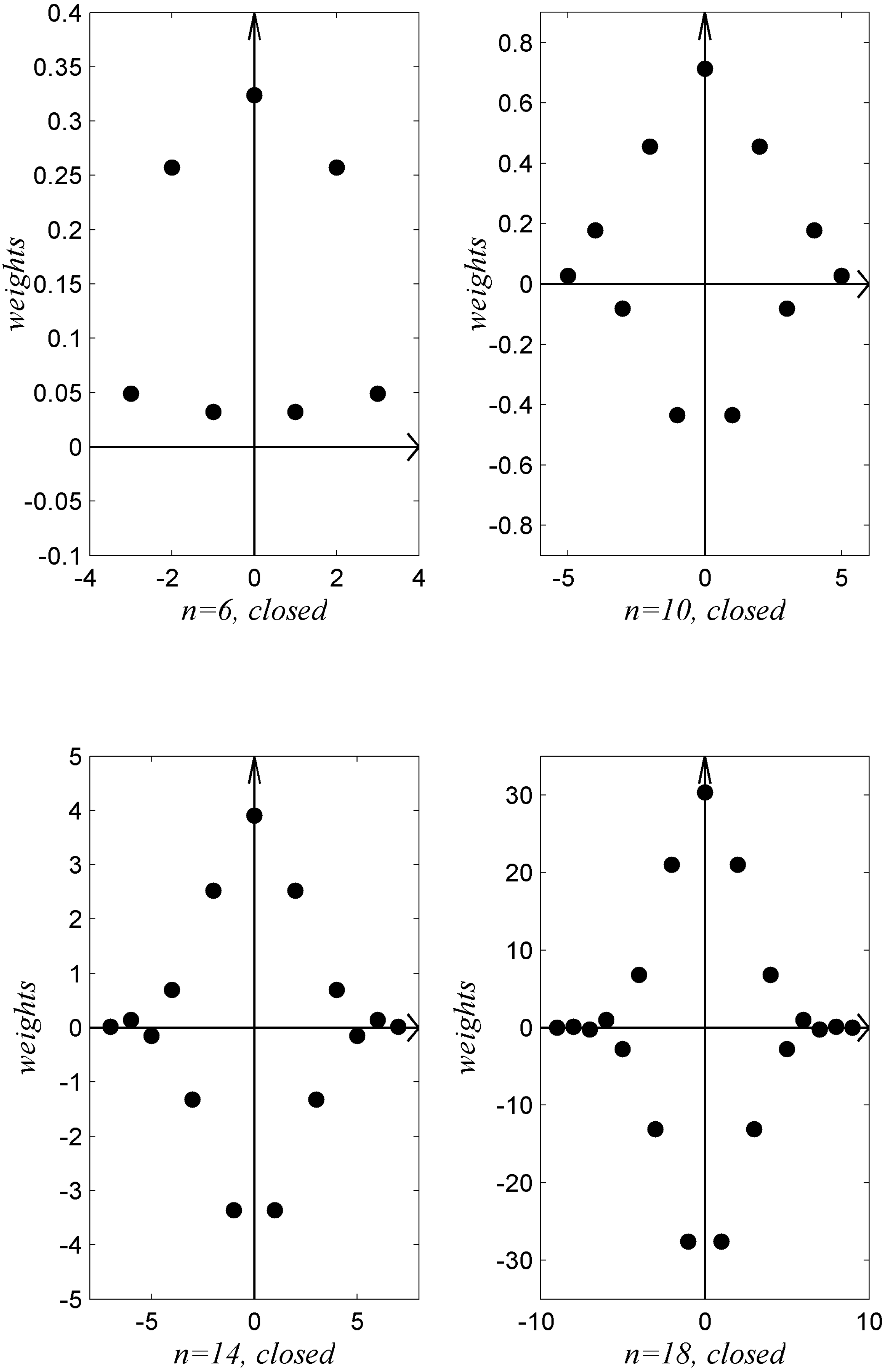
n	Coefficients
2	$w_0 = 1/6, w_1 = 4/6, K_2{=}1/90$
4	$w_0 = 7/90, w_1 = 32/90$
	$w_2 = 12/90$
	$K_4 = 8/945$
6	$w_0 = 41/840$
	$w_1 = 216/840$
	$w_2 = 27/840$
	$w_3 = 272/840$
	$K_6 = 9/1400$
8	$w_0 = 989/28350$
	$w_1 = 5888/28350$
	$w_2 = -928/28350$
	$w_3 = 10496/28350$
	$w_4 = -4540/28350$
	$K_8 = 2368/467775$
10	$w_0 = 16067/598752$
	$w_1 = 106300/598752$
	$w_2 = -48525/598752$
	$w_3 = 272400/598752$
	$w_4 = -260550/598752$
	$w_5 = 427368/598752$
	$K_{10} = 673175/163459296$
12	$w_0 = 1364651/63063000$
	$w_1 = 9903168/63063000$
	$w_2 = -7587864/63063000$
	$w_3 = 35725120/63063000$
	$w_4 = -51491295/63063000$
	$w_5 = 87516288/63063000$
	$w_6 = -87797136/63063000$
	$K_{12} = 3012/875875$

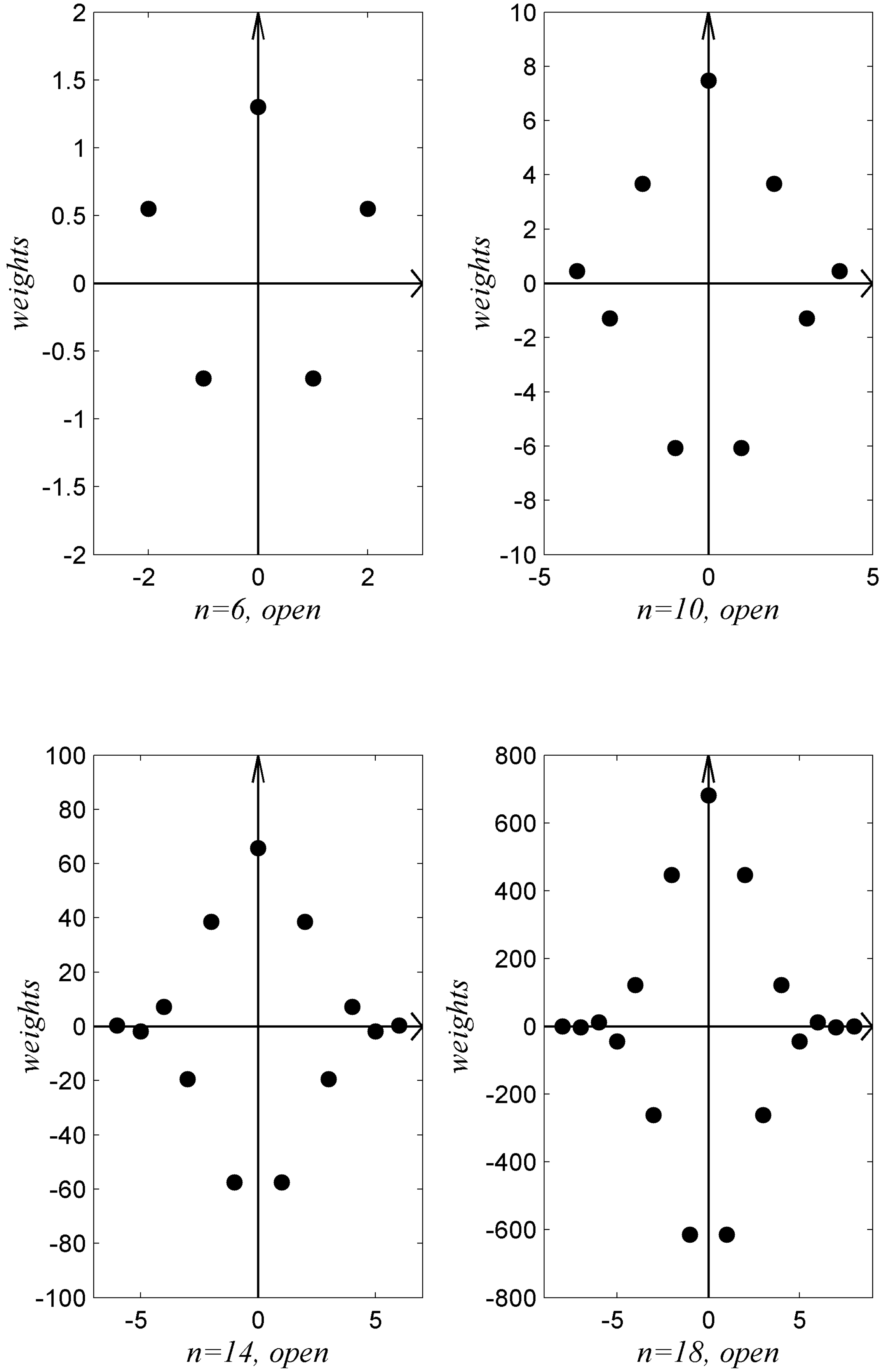
n	Coefficients
14	$w_0 = 90241897/5003856000$
	$w_1 = 710986864/5003856000$
	$w_2 = -770720657/5003856000$
	$w_3 = 3501442784/5003856000$
	$w_4 = -6625093363/5003856000$
	$w_5 = 12630121616/5003856000$
	$w_6 = -16802270373/5003856000$
	$w_7 = 19534438464/5003856000$
	$K_{14} = 3740727473/1275983280000$
16	$w_0 = 15043611773/976924698750$
	$w_1 = 127626606592/976924698750$
	$w_2 = -179731134720/976924698750$
	$w_3 = 832211855360/976924698750$
	$w_4 = -1929498607520/976924698750$
	$w_5 = 4177588893696/976924698750$
	$w_6 = -6806534407936/976924698750$
	$w_7 = 9368875018240/976924698750$
	$w_8 = -10234238972220/976924698750$
	$K_{16} = 99059365376/38979295480125$
18	$w_0 = 203732352169/15209113920000$
	$w_1 = 1848730221900/15209113920000$
	$w_2 = -3212744374395/15209113920000$
	$w_3 = 15529830312096/15209113920000$
	$w_4 = -42368630685840/15209113920000$
	$w_5 = 103680563465808/15209113920000$
	$w_6 = -198648429867720/15209113920000$
	$w_7 = 319035784479840/15209113920000$
	$w_8 = -419127951114198/15209113920000$
	$w_9 = 461327344340680/15209113920000$
	$K_{18} = 622720042317/278833755200000$

Table 1: Coefficients for Closed Type Newton–Cotes Formulas, $n = 2 - 18$

4	$w_1 = 2/3, w_2 = -1/3, L_4 = 14/45$
6	$w_1 = 11/20,\ w_2 = -14/20$ $w_3 = 26/20$ $L_6 = 41/140$
8	$w_1 = 460/945$ $w_2 = -954/945$ $w_3 = 2196/945$ $w_4 = -2459/945$ $L_8 = 3956/14175$
10	$w_1 = 4045/9072$ $w_2 = -11690/9072$ $w_3 = 33340/9072$ $w_4 = -55070/9072$ $w_6 = 67822/9072$ $L_{10} = 80335/299376$
12	$w_1 = 9626/23100$ $w_2 = -35771/23100$ $w_3 = 123058/23100$ $w_4 = -266298/23100$ $w_5 = 427956/23100$ $w_6 = -494042/23100$ $L_{12} = 1364651/5255250$
14	$w_1 = 329062237/833976000$ $w_2 = -1497122214/833976000$ $w_3 = 6058248882/833976000$ $w_4 = -16159538710/833976000$ $w_5 = 32215733235/833976000$ $w_6 = -47966447844/833976000$ $w_7 = 54874104828/833976000$ $L_{14} = 631693279/2501928000$
16	$w_1 = 722204696/1915538625$ $w_2 = -3892087348/1915538625$ $w_3 = 18150263624/1915538625$ $w_4 = -57468376538/1915538625$ $w_5 = 137035461016/1915538625$ $w_6 = -249560348012/1915538625$ $w_7 = 355819203336/1915538625$ $w_8 = -399697102923/1915538625$ $L_{16} = 120348894184/488462349375$
18	$w_1 = 6912171129/19059040000$ $w_2 = -43087461474/19059040000$ $w_3 = 227788759000/19059040000$ $w_4 = -834322842510/19059040000$ $w_5 = 2317367615100/19059040000$ $w_6 = -4988390746282/19059040000$ $w_7 = 8524579147752/19059040000$ $w_8 = -11696802277350/19059040000$ $w_9 = 12990970309270/19059040000$ $L_{18} = 611197056507/2534852320000$
20	$w_1 = 1749481500626/4989349821456$ $w_2 = -12389954060697/4989349821456$ $w_3 = 73278572831682/4989349821456$ $w_4 = -304672055470086/4989349821456$ $w_5 = 966316491145704/4989349821456$ $w_6 = -2400158698258188/4989349821456$ $w_7 = 4782407754794376/4989349821456$ $w_8 = -7751977518223986/4989349821456$ $w_9 = 10322815990097148/4989349821456$ $w_{10} = -11349750778891702/4989349821456$ $L_{20} = 1145302367137/4842604238472$

Table 2: Coefficients for Open Type Newton–Cotes Formulas, $n = 4 - 20$


weights
n=6, closed
weights
n=10, closed
weights
n=14, closed
weights
n=18, closed


weights
n=6, open
weights
n=10, open
weights
n=14, open
weights
n=18, open

1.	$\int_0^{100\pi} x^2 \sin x \, dx$	11.	$\int_1^{1000} \frac{\sin x}{x} \, dx$
2.	$\int_0^{10} \sin x \, dx$	12.	$\int_0^1 x \sin(20\pi x) \cos(2\pi x) \, dx$
3.	$\int_0^4 x^4 e^{-x} \, dx$	13.	$\int_0^{\pi} \cos(\cos x + 3 \sin x + 2 \cos 2x + 3 \cos 3x) \, dx$
4.	$\int_1^{100} \frac{1}{x} \, dx$	14.	$\int_{-1}^1 (23/25) \cosh x - \cos x \, dx$
5.	$\int_{e^1}^{e^2} \frac{1}{x \ln x} \, dx$	15.	$\int_{-1}^1 \frac{1}{x^4 + x^2 + 0.9} \, dx$
6.	$\int_0^{500} \cos x \, dx$	16.	$\int_0^1 x^{3/2} \, dx$
7.	$\int_1^{10000} \ln x \, dx$	17.	$\int_0^1 \frac{1}{1 + x^4} \, dx$
8.	$\int_0^{300} e^{-0.1x} \sin x \, dx$	18.	$\int_0^1 \frac{1}{1 + x} \, dx$
9.	$\int_0^{\pi/2} \frac{1}{1 + \cos x} \, dx$	19.	$\int_0^1 \frac{1}{1 + e^x} \, dx$
10.	$\int_0^{1000} e^{-x/100} \sin x \, dx$	20.	$\int_{0.1}^1 \frac{\sin(100\pi x)}{\pi x} \, dx$
		21.	$\int_0^{400} x^3 e^{-x} \, dx$

Table 3:

5. Tests

We have written a program `newtoncotes` in MATLAB to compare different orders. The user gives the integrand and endpoints, the order, total number of points to be used and an option: `open` or `closed`. The program then rounds the number of points above if necessary, and evaluates the integral using composite Newton–Cotes integration.

For example, `newtoncotes('closed',8,'x^2*sin(x)',0,pi,100)` calculates $\int_0^{\pi} x^2 \sin(x) \, dx$ using $n = 8$ closed Newton–Cotes rule and at least 100 points.
`[res,pts]=newtoncotes('closed',8,'x^2*sin(x)',0,pi,100)` will do the same calculation, and also give `pts=105` which is the exact number of points used.

We have tested the methods extensively, using a variety of integrands given in Tables (3-5). For each integrand, we have gradually increased the number of points until relative errors of 10^{-12} or less have been obtained. We have plotted the logarithm of relative error with respect to number of points to compare methods easily. Some typical graphs are given on Figures (3-4).

The results are dependent on the integrand. For those in Table 3,

- Closed rules are always better than open rules using same number of points
- Higher order closed rules are better than lower orders up to $n = 10$ or 12
- After $n = 12$, for computations using a small number of points (around 10^3), the advantage of still higher orders is insignificant. If we use a large number of points, (around 10^6) higher order rules become less reliable and results start to oscillate.

1. $\int_0^1 \sqrt{x}\,dx$	4. $\int_{10^{-10}}^1 \frac{x}{e^x-1}\,dx$
2. $\int_0^{100} \sqrt{x}e^{-x}\,dx$	5. $\int_{10^{-10}}^1 \frac{1}{x^{0.4}}\,dx$
3. $\int_{10^{-8}}^1 \frac{1}{\sqrt{x}}\,dx$	6. $\int_{10^{-10}}^1 -\ln x\,dx$

Table 4:

1. $\int_{-1}^1 \frac{1}{1.005+x^2}\,dx$	5. $\int_0^1 \frac{1}{1+(230x-30)^2}\,dx$
2. $\int_{10^{-10}}^1 \left(\frac{\sin(50\pi x)}{10x}\right)^2\,dx$	6. $\int_0^1 \frac{2}{2+\sin(10\pi x)}\,dx$
3. $\int_{-10}^{10} \frac{1}{1+x^2}\,dx$	7. $\int_0^{10} e^{-50\pi x^2}\,dx$
4. $\int_0^{2\pi} \cos(300\sin x)\,dx$	8. $\int_0^{10} \frac{50}{\pi(2500x^2+1)}\,dx$

Table 5:

All of these are expected on theoretical grounds.

The recommended rule is closed 10.

The integrands on Table 4 on the other hand, involve singularities or infinite derivatives at end points. For these functions,

- Open rules are always better than closed rules using same number of points
- Higher order open rules have an almost invisible advantage
- If we use a large number of points, (around 10^6) higher order rules become less reliable and results start to oscillate.

Taking all these into account, we recommend the rule open 6.

The integrals on Table 5 are more difficult in a numerical sense. They may involve rapid oscillations or rational functions where approximating by polynomials may be inappropriate. For such functions, the smaller orders give better results, so we recommend Simpson's rule. (closed 2)

It is well known that fast oscillatory functions are among the most difficult to integrate numerically, and specialized techniques are necessary. Some ideas and methods can be seen in: [11],[12],[13],[14].

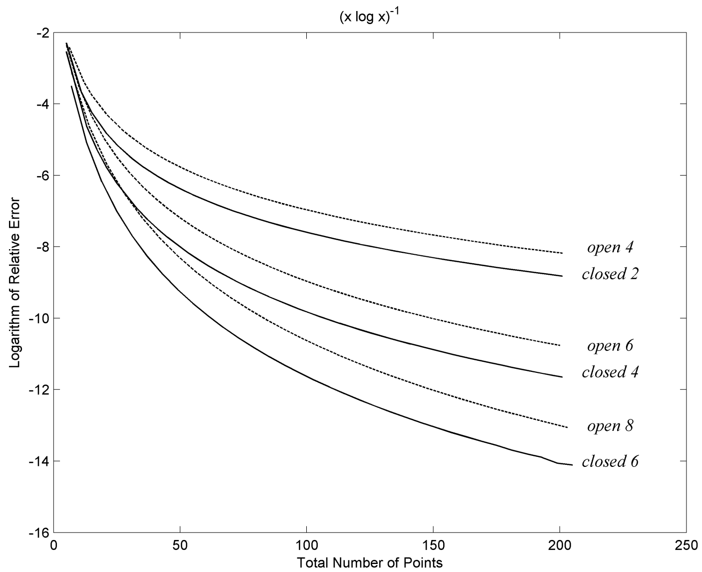
The application of related ideas to differential equations can be found in: [15],[16],[17].

6. Conclusion

While Newton-Cotes rules are elementary, the application of these rules become cumbersome as we go to higher orders. In this work, we have derived formulas and tabulated coefficients up to $n = 18$ for closed and $n = 20$ for open rules.

The conventional wisdom is to use the lowest orders $n = 2$ or $n = 4$. After extensive testing, we have confirmed that for high orders ($n > 12$) increasing roundoff errors make the application unfeasible. But the best overall rule is closed rule of $n = 10$ for most proper integrals. The existence of negative coefficients does not necessarily make this rule unreliable. For integrals with singularities or infinite derivatives at the endpoints, open rule of $n = 6$ is recommended, and going to higher orders makes almost no difference.

The Simpson rule should still exist on the arsenal of numerical integrator, but it gives better results for exceptionally troublesome integrals only.



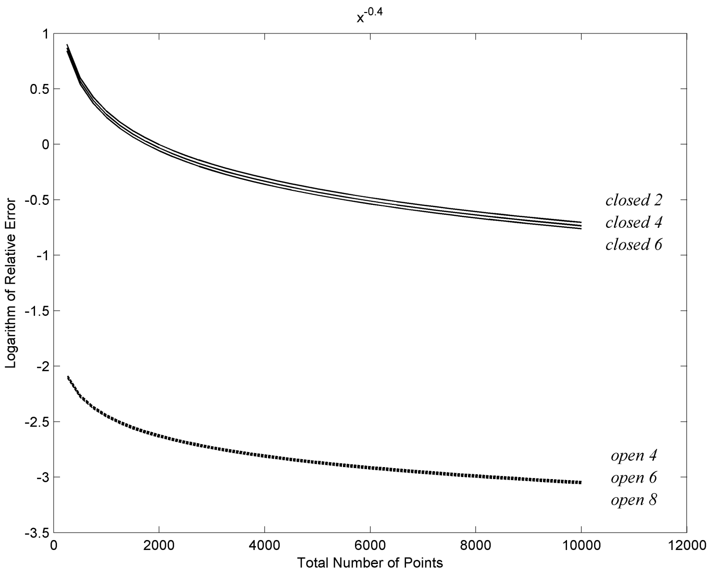
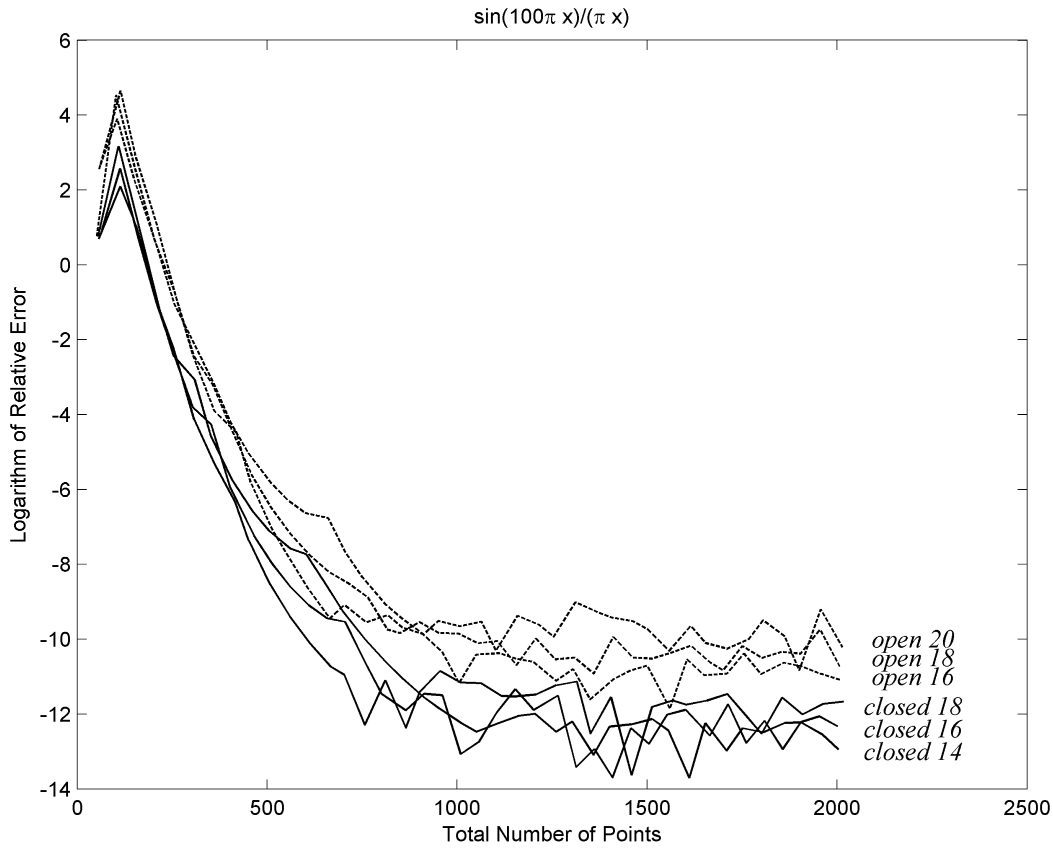


Figure 3: Test Results for $\int_{e^1}^{e^2} \frac{1}{x \ln x}\, dx$ and $\int_{10^{-10}}^{1} \frac{1}{x^{0.4}}\, dx$



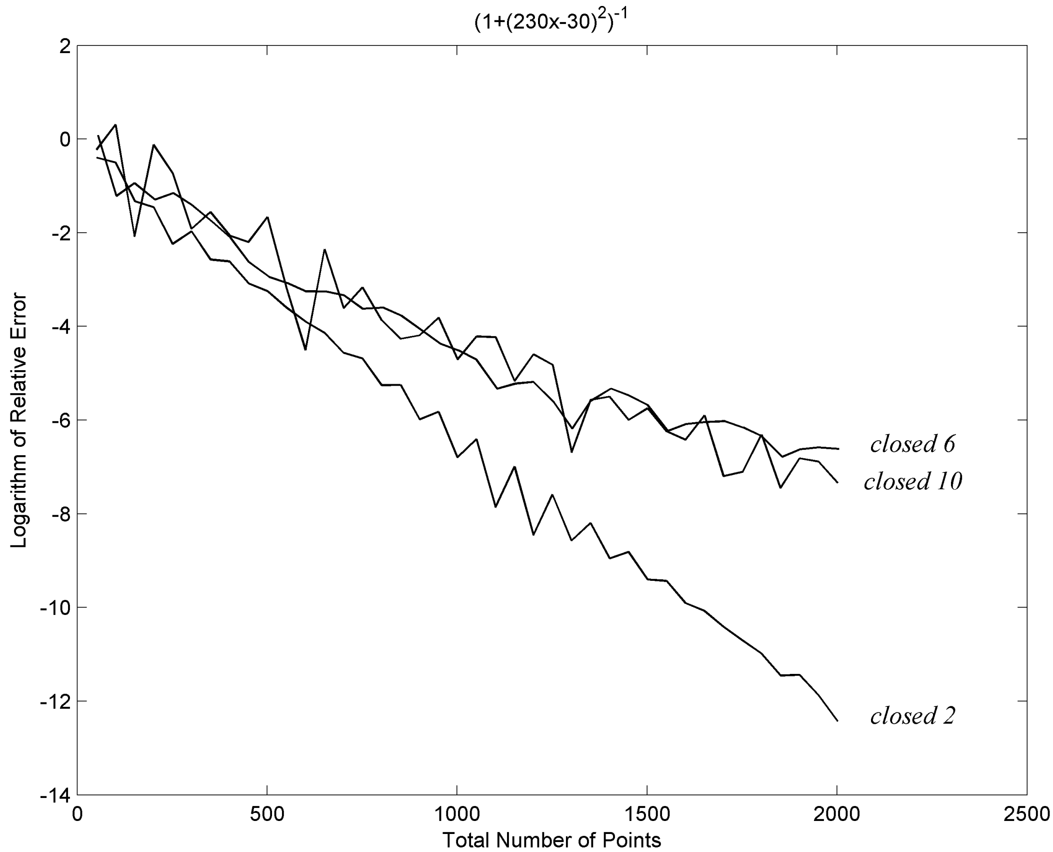


Figure 4: Test Results for $\int_{0.1}^{1} \frac{\sin(100\pi x)}{\pi x}\,dx$ and $\int_{0}^{1} \frac{1}{1+(230x-30)^2}\,dx$

References

[1] E. Sermutlu, Comparison of Newton–Cotes and Gaussian methods of quadrature, *Applied Mathematics and Computation*, **171** (2005), 1048–1057.

[2] T.H.Fay and P.G.Webster, Lagrange interpolation and Runge's example, *International Journal of Mathematical Education in Science and Technology*, **27(6)** (1996), 785–795.

[3] M. El-Mikkawy, A unified approach to Newton–Cotes quadrature formulae, *Applied Mathematics and Computation*, **138** (2003), 403–413.

[4] M. El-Mikkawy, On the error analysis associated with the Newton–Cotes formulae, *International Journal of Computer Mathematics*, **79(9)** (2002), 1043–1047.

[5] M. Dehghan, M. Masjed-Jamei, M.R. Eslahchi, On numerical improvement of closed Newton–Cotes quadrature rules, *Applied Mathematics and Computation*, **165** (2005), 251–260.

[6] M. Dehghan, M. Masjed-Jamei, M.R. Eslahchi, On numerical improvement of open Newton–Cotes quadrature rules, *Applied Mathematics and Computation*, **175** (2006), 618–627.

[7] T.E.Simos, High–order closed Newton–Cotes trigonometrically - fitted formulae for long-time integration of orbital problems, *Computer Physics Communications*, **178** (2008), 199–207.

[8] M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions, Dover Publications, 1965.

[9] D.R. Hayes and L. Rubin, A proof of the Newton–Cotes quadrature formulas with error term, *The American Mathematical Monthly*, **77(10)** (1970), 1065–1072.

[10] G.A. Evans, The estimation of errors in numerical quadrature, *International Journal of Mathematical Education in Science and Technology*, **25(10)** (1994), 727–744.

[11] K.N. Melnik, R.V.N. Melnik, Optimal-by-order quadrature formulae for fast oscillatory functions with inaccurately given a priori information, *Journal of Computational and Applied Mathematics*, **110** (1999), 45–72.

[12] K.N. Melnik, R.V.N. Melnik, Optimal cubature formulae and recovery of fast-oscillating functions from an interpolational class, *BIT*, **41(4)** (2001), 748–775.

[13] K.N. Melnik, R.V.N. Melnik, Optimal-by-accuracy and optimal-by-order cubature formulae in interpolational classes, *Journal of Computational and Applied Mathematics*, **147** (2002), 233–262.

[14] E. Sermutlu, H.T. Eyyuboğlu, A new quadrature routine for improper and oscillatory integrals, *Applied Mathematics and Computation*, **189** (2007), 452–461.

[15] A.Ullah and K.Shah, Numerical analysis of Lane EmdenâĂŞ-Fowler equations, *Journal of Taibah University for Science*, **12(2)** (2018), 180–185.

[16] W.Kumam, M.B.Zada, K.Shah and R.A.Khan, Investigating a Coupled Hybrid System of Nonlinear Fractional Differential Equations, *Discrete Dynamics in Nature and Society*, (2018). doi: 10.1155/2018/5937572.

[17] K. Shah and J.Wang, A numerical scheme based on non-discretization of data for boundary value problems of fractional order differential equations, *Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. Serie A. Matemáticas*, (2018). doi:10.1007/s13398-018-0616-7.