Katarína Tvrdá; Peter Novotný

Modifications of Newton-Cotes formulas for computation of repeated integrals and derivatives

Czechoslovak Mathematical Journal, Vol. 73 (2023), No. 4, 1175–1188

Persistent URL: `http://dml.cz/dmlcz/151953`

Czechoslovak Mathematical Journal, 73 (148) (2023), 1175–1188

MODIFICATIONS OF NEWTON-COTES FORMULAS FOR COMPUTATION OF REPEATED INTEGRALS AND DERIVATIVES

Katarína Tvrdá, Bratislava, Peter Novotný, Žilina

Received October 2, 2022. Published online March 24, 2023.

Abstract. Standard algorithms for numerical integration are defined for simple integrals. Formulas for computation of repeated integrals and derivatives for equidistant domain partition based on modified Newton-Cotes formulas are derived. We compare usage of the new formulas with the classical quadrature formulas and discuss possible application of the results to solving higher order differential equations.

Keywords: repeated integral; Cauchy formula for repeated integration; quadrature; cubature; numerical differentiation

MSC 2020: 65D32

1. Introduction

The initial motivation for derivation of the formulas discussed in this paper originates from technical practice. One of the classic problems in construction engineering is to determine the effect of various factors (weather, load, etc.) on the structure deformations. The basic theory used to model such deformations is Euler-Bernoulli beam theory, see e.g., [2], [6], [7].

Consider a tall bridge pillar made of reinforced concrete, see Figure 1 (a). The pillar is affected by vertical forces N (weight of the bridge, traffic) and horizontal forces F (wind), which generate bending moment $M(x)$. Let L be the height of the pillar. Denote by $W(x)$ for $x \in [0, L]$ the deflection of the pillar at x caused by the forces. The initial deflection caused by the horizontal forces which generate linear

This paper was supported by VEGA Grant Agency, project no. 1/0453/20 and VEGA Grant Agency, project no. 1/0155/23.

DOI: 10.21136/CMJ.2023.0437-22

bending moment can be obtained from the classic Euler-Bernoulli equation

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\Big(EI(x)\frac{\mathrm{d}^2W(x)}{\mathrm{d}x^2}\Big) = q(x).$$

Here, $EI(x)$ is the flexural rigidity of the structure and $q(x)$ is the distributed load.

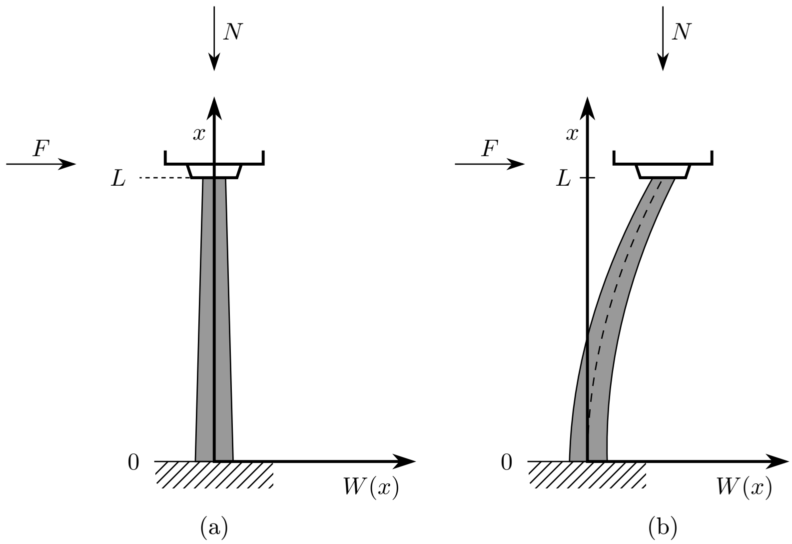


Figure 1. Bridge pillar.

The vertical forces do not affect the initial deflection, but this is no more true after the deformation occurs, see Figure 1 (b). An additional bending moment is generated which creates an additional deflection. This generates another moment and so on, until the equilibrium is reached. The deflection can be calculated iteratively using the formula

$$W_{k+1}(x) = \int_0^x \int_0^{x_1} \frac{M(x_2) + F(W_k(L) - W_k(x_2))}{EI(M, N, W_k, x_2)}\,\mathrm{d}x_2\,\mathrm{d}x_1 \quad \text{for } k = 0, 1, \ldots,$$

where W_0 is the initial deflection, see [8] for more details.

To be able to perform the above iterations, we need an effective way to evaluate the repeated integral. The expression under the integral is a complicated function and the integration can be accomplished only numerically using the values at discrete points.

We now introduce the concept more formally. Let $a \leqslant b$ be real numbers, $n \geqslant 2$ be an integer and

$$f\colon\ [\min\{0, a\}, \max\{0, b\}] \to \mathbb{R}$$

be a continuous function. We use the notation

$$f_\alpha^{(-n)}(\beta) := \int_\alpha^\beta \int_\alpha^{x_1} \ldots \int_\alpha^{x_{n-1}} f(x_n)\,\mathrm{d}x_n \ldots \mathrm{d}x_2\,\mathrm{d}x_1 \tag{1.1}$$

for the nth repeated integral of f based at α. Consider the repeated integral

$$D_{[a,b]}^{(-n)}(f) := \int_a^b \int_0^{x_1} \int_0^{x_2} \cdots \int_0^{x_{n-1}} f(x_n)\,\mathrm{d}x_n \ldots \mathrm{d}x_3\,\mathrm{d}x_2\,\mathrm{d}x_1.$$

In [9], the formula

$$D_{[a,b]}^{(-n)}(f) = \sum_{i=1}^{n-1} \frac{(b-a)^i}{i!} f_0^{(i-n)}(a) + f_a^{(-n)}(b) \tag{1.2}$$

is derived. Using this result, one can rewrite any repeated integral as a sum of repeated integrals with a fixed lower boundary in each of them. By the Cauchy formula for repeated integration (see [4], page 193) we have

$$f_\alpha^{(-n)}(\beta) = \frac{1}{(n-1)!} \int_\alpha^\beta (\beta - t)^{n-1} f(t)\,\mathrm{d}t. \tag{1.3}$$

With the help of this formula, one can rewrite the repeated integrals in (1.2) in terms of simple integrals. In [9], this approach is applied to obtain the weights for the cubature of Gaussian type. In the present work, we use it to derive the formulas for the computation of repeated integrals and derivatives in the Newton-Cotes fashion.

2. Results obtained by the Cauchy formula for repeated integration

By rewriting the integral in (1.3) using the Newton-Cotes formula (see, e.g., [3], Section 4.3) with k subintervals of length $h = (\beta - \alpha)/k$ and points $t_j = \alpha + jh$ for $j = 0, 1, \ldots, k$, we obtain

$$\begin{aligned} \int_\alpha^\beta (\beta - t)^{n-1} f(t)\,\mathrm{d}t &= Ah \sum_{j=0}^{k} w_j (\beta - t_j)^{n-1} f(t_j) + R \\ &= Ah \sum_{j=0}^{k} w_j (h(k-j))^{n-1} f(t_j) + R \\ &= Ah^n \sum_{j=0}^{k} w_j (k-j)^{n-1} f(t_j) + R. \end{aligned} \tag{2.1}$$

Here, w_j and A are standard Newton-Cotes weights and coefficient, and R is the error term. For instance, when we use the classical Simpson's rule, we have $k = 2$, $A = \frac{1}{3}$, $(w_0, w_1, w_2) = (1, 4, 1)$ and $R = -\frac{1}{90} h^5 f^{(4)}(\xi)$ for some $\xi \in (\alpha, \beta)$.

We denote by

$$w_{n,j} = w_j (k-j)^{n-1}, \quad A_n = \frac{A}{(n-1)!} \tag{2.2}$$

the new weights and coefficient for the nth repeated integral. Hence, we have

$$f_\alpha^{(-n)}(\beta) = h^n A_n \sum_{j=0}^{k} w_{n,j} f(t_j) + R. \tag{2.3}$$

When we use (2.1) to compute the nth repeated integral, we integrate the function $(\beta - t)^{n-1} f(t)$ instead of $f(t)$. This usually leads to a loss of precision, since Newton-Cotes formulas produce exact results for polynomials up to a certain degree and the term $(\beta - t)^{n-1}$ rises the degree of the integrand. On the other hand, the benefit of this approach is that in (2.2) the value of $w_{n,j}$ for $j = k$ is zero. Therefore, we do not need to know the value of f in β to compute (2.3). This is very useful when applying the result to stepwise solving of higher order differential equations – we can avoid using open Newton-Cotes formulas.

The weights and coefficients derived by (2.2) for some of the Newton-Cotes schemes are presented in Tables 1 and 2 in the appendix.

3. Results obtained by integration of Lagrange interpolation polynomial

The second method to calculate (1.1) is derived by replacing f with the Lagrange interpolation polynomial. We present here the details of the derivation with four interpolation nodes, the same technique can be applied to any number of nodes.

Let the notation f_j be a shorthand for $f(t_j)$ with $h = \frac{1}{3}(\beta - \alpha)$ and $t_j = \alpha + jh$ for $j = 0, 1, 2, 3$. Let L be the Lagrange interpolating polynomial satisfying $L(t_j) = f_j$, i.e.,

$$L(x) = f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x) + f_3 l_3(x),$$

where $\{l_0, l_1, l_2, l_3\}$ is the standard Lagrange basis. By integrating,

$$\begin{aligned} L_\alpha^{(-1)}(\beta) &= \int_{t_0}^{t_3} L(x)\,\mathrm{d}x = h \int_0^3 L(t_0 + uh)\,\mathrm{d}u \\ &= h \sum_{j=0}^{3} f_j \int_0^3 l_j(t_0 + uh)\,\mathrm{d}u = \frac{3}{8} h (f_0 + 3f_1 + 3f_2 + f_3), \end{aligned}$$

which is classical Simpson's $\frac{3}{8}$ rule, see, e.g., [1], equation 25.4.13.

Similarly, we have

$$L_\alpha^{(-2)}(\beta) = \int_{t_0}^{t_3} \int_{t_0}^{x_1} L(x_2)\,\mathrm{d}x_2\,\mathrm{d}x_1 = h^2 \int_0^3 \int_0^{u_1} L(t_0 + u_2 h)\,\mathrm{d}u_2\,\mathrm{d}u_1 \tag{3.1}$$
$$= h^2 \sum_{j=0}^{3} f_j \int_0^3 \int_0^{u_1} l_j(t_0 + u_2 h)\,\mathrm{d}u_2\,\mathrm{d}u_1$$
$$= \frac{3}{40} h^2 (13 f_0 + 36 f_1 + 9 f_2 + 2 f_3).$$

The last expression displays the weights $w_{2,j}$ and the coefficient A_2 for the second repeated integral. We can obtain $w_{n,j}$ and A_n for other values of n in the same fashion.

These values (except for closed formulas for $n = 2$ with an odd number of nodes) are different from the values derived by the Cauchy formula in the previous section. When using them, we do not lose precision when n increases – the method produces exact results for polynomials up to the degree equal to the number of subintervals, regardless of the value of $n \geqslant 2$. On the other hand, the weights may be negative, which affects the stability, and they are not zero at the end of the interval.

We can generalize the previous concept and notation by using it also for negative values of n, thus deriving the formulas for calculation of derivatives of f. The derivative of f of order n can be viewed as the integral of the derivative of f of order $n + 1$. When we replace f with the Lagrange interpolation polynomial, we need to select the degree of the polynomial with respect to the order of derivative. For the second derivative with four interpolation nodes, using the same notation as above, we obtain

$$\int_{t_0}^{t_3} \frac{\mathrm{d}^2}{\mathrm{d}x^2} L(x)\,\mathrm{d}x = h \int_0^3 \frac{\mathrm{d}^2}{h^2\,\mathrm{d}u^2} L(t_0 + uh)\,\mathrm{d}u \tag{3.2}$$
$$= \frac{1}{h} \sum_{j=0}^{3} f_j \int_0^3 \frac{\mathrm{d}^2}{\mathrm{d}u^2} l_j(t_0 + uh)\,\mathrm{d}u$$
$$= \frac{3}{2h}(f_0 - f_1 - f_2 + f_3).$$

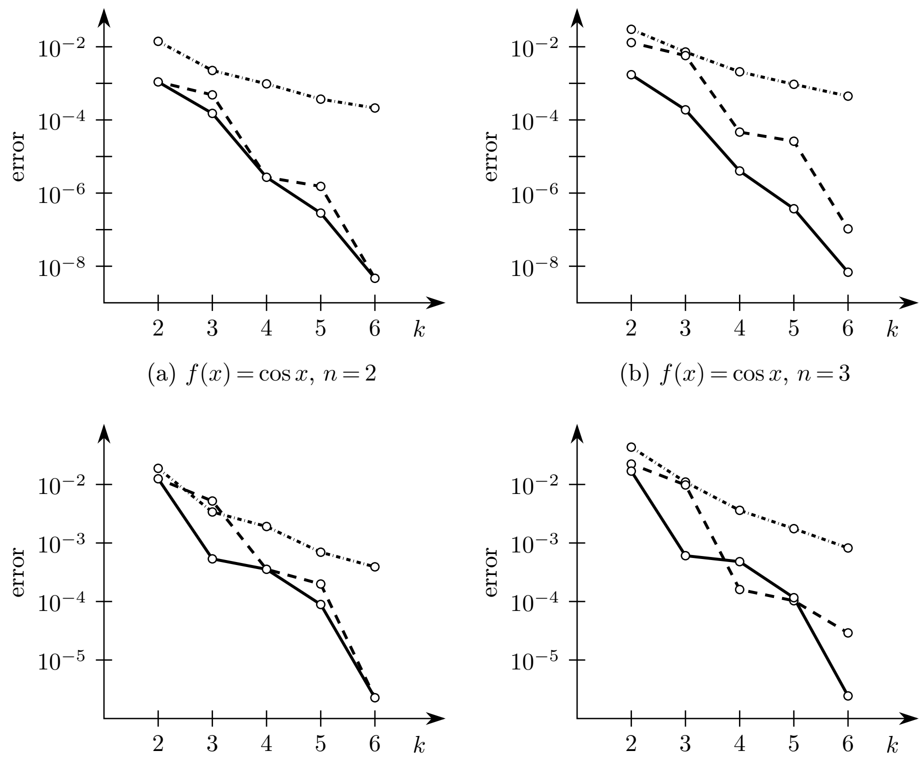
Denote by $w_{1-n,j}$ and A_{1-n} the derived weights and coefficient, respectively, where n is the order of the derivative. Hence, we have $w_{-1,0} = 1$, $w_{-1,1} = -1$, $w_{-1,2} = -1$, $w_{-1,3} = 1$ and $A_{-1} = \frac{3}{2}$.

The method used in this section may be applied for any number of nodes and works both for open and closed formulas. The weights and coefficients derived by this technique (both for repeated integrals and derivatives) are presented in Table 3 in the appendix.

4. Comparison with results obtained by repeated simple integration

Let us consider the situation as at the beginning of Section 1, where we need to evaluate (1.1) for some function f, integer $n \geqslant 2$ and endpoints α, β, with the values of f being known only at discrete equidistant points $t_j \in [\alpha, \beta]$ for $j = 0, \ldots, k$. Without the derived formulas and weights, we are left with the classical quadrature formulas for simple integrals applied repeatedly. During this process, when calculating the inner sub-integrals with bounds $[\alpha, t_j]$, only a portion of equidistant points lies between the bounds. Therefore, we need to use quadrature formulas with lower number of nodes, which leads to the loss of precision.

The plots on Figure 2 show the comparison of the errors obtained when using our method versus using the classical Newton-Cotes formulas for simple integrals. The calculations were performed for functions $\cos x$ and $1/(1+x^2)$ with endpoints $\alpha = 0$ and $\beta = 1$.



(a) $f(x) = \cos x$, $n = 2$

(b) $f(x) = \cos x$, $n = 3$

(c) $f(x) = 1/(1+x^2)$, $n = 2$

(d) $f(x) = 1/(1+x^2)$, $n = 3$

Figure 2. Integration error; dash-and-dotted—simple integrals, dashed—method from Section 2, solid—method from Section 3.

5. Application of the results to the solution of differential equations

The derived formulas can be used as a tool for solving higher-order differential equations. Consider an unknown function f with known values of its derivatives and integrals in the initial point t_0 up to some orders. More formally, let $f^{(i)}$ denote the ith derivative of f for $i \geqslant 0$ and the repeated integral (antiderivative) of order $|i|$ for $i < 0$, respectively. Suppose that the values of

$$f^{(-n)}(t_0), f^{(-n+1)}(t_0), \ldots, f^{(m-1)}(t_0), f^{(m)}(t_0) \tag{5.1}$$

are all known. For a given step h, let $t_j = t_0 + jh$ for $j = 0, 1, \ldots, k$. We can estimate the values of $f(t_j)$ from (5.1) using Taylor series approximations or other methods. Then we can use the weights and coefficients derived in the previous sections to calculate the values of

$$f^{(-n)}(t_k), f^{(-n+1)}(t_k), \ldots, f^{(m-1)}(t_k), f^{(m)}(t_k). \tag{5.2}$$

The process of obtaining (5.2) from (5.1) can be replicated variously: The role of $f = f^{(0)}$ in the list $f^{(-n)}, f^{(-n+1)}, \ldots, f^{(m-1)}, f^{(m)}$ is not special. We can select any function from this list as a basis function and consider the other functions as its derivatives and antiderivatives. This variability offers the opportunity to choose an optimal strategy based on the specific properties of the equation we are dealing with.

In technical practice, we often deal with the case, where both the values of (5.1) and (5.2) are known and our goal is to find $f(t_j)$ for $j = 1, 2, \ldots, k-1$. It is a special case of boundary value problem with known values playing the role of boundary conditions. The approach described above can be inverted. Since in our formulas (5.2) they depend linearly on $f(t_j)$, we can gain the information about $f(t_j)$ by solving a simple linear system. In Appendix C, we provide an example demonstrating this procedure.

6. Conclusions

In this paper, we present a new method for computation of repeated integrals and derivatives from known values of the function in equidistant points. Based on this, we propose a novel technique for dealing with some kinds of boundary value problems. It can be used to extract information about the unknown function in a very simple manner, which makes it an alternative approach to other methods, see Appendix C for demonstration.

The paper illustrates how to derive the weights for a small number of nodes from classical Newton-Cotes formulas. It is well-known that polynomial interpolation at equidistant nodes may introduce high interpolation error. Therefore, using the same approach with a higher number of nodes may not be optimal. In such cases, one may consider to modify a different kind of formulas in a similar way, for example stable Newton-Cotes formulas introduced in [5].

Appendix A. Derived weights and coefficients

Tables 1 and 2 contain the weights and coefficients derived by the technique described in Section 2. Table 3 contains the weights and coefficients derived by the technique described in Section 3. In all the tables, k is the number of subintervals and m is the highest polynomial degree for which the formula produces exact results.

k	n	A_n	$w_{n,0}$	$w_{n,1}$	$w_{n,2}$	$w_{n,3}$	$w_{n,4}$	$w_{n,5}$	$w_{n,6}$	m
2	1	1/3	1	4	1					3
	2	1/3	2	4	0					2
	3	1/6	4	4	0					1
	4	1/18	8	4	0					0
3	1	3/8	1	3	3	1				3
	2	3/8	3	6	3	0				2
	3	3/16	9	12	3	0				1
	4	3/48	27	24	3	0				0
4	1	2/45	7	32	12	32	7			5
	2	2/45	28	96	24	32	0			4
	3	2/90	112	288	48	32	0			3
	4	2/270	448	864	96	32	0			2
5	1	5/288	19	75	50	50	75	19		5
	2	5/288	95	300	150	100	75	0		4
	3	5/576	475	1200	450	200	75	0		3
	4	5/1728	2375	4800	1350	400	75	0		2
6	1	1/140	41	216	27	272	27	216	41	7
	2	1/140	246	1080	108	816	54	216	0	6
	3	1/280	1476	5400	432	2448	108	216	0	5
	4	1/840	8856	27000	1728	7344	216	216	0	4

Table 1. Weights and coefficients derived by the Cauchy theorem from closed Newton-Cotes formulas.

k	n	A_n	$w_{n,1}$	$w_{n,2}$	$w_{n,3}$	$w_{n,4}$	$w_{n,5}$	$w_{n,6}$	$w_{n,7}$	m
4	1	4/3	2	−1	2					3
	2	4/3	6	−2	2					2
	3	2/3	18	−4	2					1
	4	2/9	54	−8	2					0
5	1	5/24	11	1	1	11				3
	2	5/24	44	3	2	11				2
	3	5/48	176	9	4	11				1
	4	5/144	704	27	8	11				0
6	1	3/10	11	−14	26	−14	11			5
	2	3/10	55	−56	78	−28	11			4
	3	3/20	275	−224	234	−56	11			3
	4	1/20	1375	−896	702	−112	11			2
7	1	7/1440	611	−453	562	562	−453	611		5
	2	7/1440	3666	−2265	2248	1686	−906	611		4
	3	7/2880	21996	−11325	8992	5058	−1812	611		3
	4	7/8640	131976	−56625	35968	15174	−3624	611		2
8	1	8/945	460	−954	2196	−2459	2196	−954	460	7
	2	8/945	3220	−5724	10980	−9836	6588	−1908	460	6
	3	8/1890	22540	−34344	54900	−39344	19764	−3816	460	5
	4	8/5670	157780	−206064	274500	−157376	59292	−7632	460	4

Table 2. Weights and coefficients derived by the Cauchy theorem from open Newton-Cotes formulas.

The presented weights and coefficients can be used with the formula similar to (2.3), that is,

$$f_\alpha^{(-n)}(\beta) \approx h^n A_n \sum_{j=0}^{k} w_{n,j} f(t_j) \quad \text{for } n > 0$$

and

$$\int_\alpha^\beta \frac{\mathrm{d}^{1-n}}{\mathrm{d}x^{1-n}} f(x)\,\mathrm{d}x \approx h^n A_n \sum_{j=0}^{k} w_{n,j} f(t_j) \quad \text{for } n < 0,$$

where $h = (\beta - \alpha)/k$ and $t_j = \alpha + jh$ for $j = 0, 1, \ldots, k$.

Note that when we use the formula to calculate (1.2), we have to take into account also the contribution of the terms $f_0^{(i-n)}(a)$, not only the single term $f_a^{(-n)}(b)$.

k	n	A_n	$w_{n,0}$	$w_{n,1}$	$w_{n,2}$	$w_{n,3}$	$w_{n,4}$	$w_{n,5}$	$w_{n,6}$	m
2	−1	2	1	−2	1					3
	1	1/3	1	4	1					3
	2	2/3	1	2	0					2
	3	1/15	9	12	−1					2
	4	2/45	8	8	−1					2
3	−2	3	−1	3	−3	1				4
	−1	3/2	1	−1	−1	1				3
	1	3/8	1	3	3	1				3
	2	3/40	13	36	9	2				3
	3	9/80	12	27	0	1				3
	4	27/280	13	24	−3	1				3
4	−2	2	−1	2	0	−2	1			4
	−1	1/3	7	−16	18	−16	7			5
	1	2/45	7	32	12	32	7			5
	2	8/45	7	24	6	8	0			4
	3	8/315	93	272	12	48	−5			4
	4	32/945	88	224	−24	32	−5			4
5	−2	5/12	−11	43	−74	74	−43	11		6
	−1	5/12	5	−9	4	4	−9	5		5
	1	5/288	19	75	50	50	75	19		5
	2	25/2016	122	475	100	250	50	11		5
	3	125/8064	233	815	10	310	−35	11		5
	4	125/72576	3346	10525	−1400	3350	−850	149		5
6	−2	3/4	−5	16	−17	0	17	−16	5	6
	−1	1/60	157	−432	675	−800	675	−432	157	7
	1	1/140	41	216	27	272	27	216	41	7
	2	3/70	41	180	18	136	9	36	0	6
	3	9/350	198	792	−45	480	−90	72	−7	6
	4	9/175	191	702	−135	380	−135	54	−7	6

Table 3. Weights and coefficients derived by the Lagrange polynomial integration

We provide an example for illustrative purposes. Consider the function $\cos x$ and the integrals

$$\text{(B.1)}\quad \int_0^{\pi/2}\int_0^{x_1}\cos x_2\,\mathrm{d}x_2\,\mathrm{d}x_1 = 1, \qquad \int_0^{\pi/2}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\cos x\,\mathrm{d}x = -1,$$
$$\int_0^{\pi/2}\int_0^{x_1}\int_0^{x_2}\cos x_3\,\mathrm{d}x_3\,\mathrm{d}x_2\,\mathrm{d}x_1 = \frac{\pi}{2}-1, \qquad \int_0^{\pi/2}\frac{\mathrm{d}^3}{\mathrm{d}x^3}\cos x\,\mathrm{d}x = 1$$

with known exact values. Using the notation (1.1), we can rewrite the expressions in the first column of (B.1) as $\cos_0^{(-2)}(\frac{1}{2}\pi)$ and $\cos_0^{(-3)}(\frac{1}{2}\pi)$, respectively.

With four equidistant nodes $t_0 = 0$, $t_1 = \frac{1}{6}\pi$, $t_2 = \frac{1}{3}\pi$ and $t_3 = \frac{1}{2}\pi$, using (2.2) and classical Simpson's $\frac{3}{8}$ rule with $A = \frac{3}{8}$ and $(w_0, w_1, w_2, w_3) = (1, 3, 3, 1)$, we have

$$w_{2,0} = 3,\quad w_{2,1} = 6,\quad w_{2,2} = 3,\quad w_{2,3} = 0,\quad A_2 = \frac{3}{8},$$
$$w_{3,0} = 9,\quad w_{3,1} = 12,\quad w_{3,2} = 3,\quad w_{3,3} = 0,\quad A_3 = \frac{3}{16}.$$

By applying the formula (2.3) we obtain

$$\cos_0^{(-2)}\left(\frac{\pi}{2}\right) \approx \frac{\pi^2}{6^2}\frac{3}{8}\left(3\cos 0 + 6\cos\frac{\pi}{6} + 3\cos\frac{\pi}{3}\right) \approx 0.99685,$$
$$\cos_0^{(-3)}\left(\frac{\pi}{2}\right) \approx \frac{\pi^3}{6^3}\frac{3}{16}\left(9\cos 0 + 12\cos\frac{\pi}{6} + 3\cos\frac{\pi}{3}\right) \approx 0.56232$$

with relative errors 0.31543% and 1.48501%, respectively.

We may estimate the same values with the weights and coefficients

$$w_{2,0} = 13,\quad w_{2,1} = 36,\quad w_{2,2} = 9,\quad w_{2,3} = 2,\quad A_2 = \frac{3}{40},$$
$$w_{3,0} = 12,\quad w_{3,1} = 27,\quad w_{3,2} = 0,\quad w_{3,3} = 1,\quad A_3 = \frac{9}{80},$$

derived as in (3.1). We obtain

$$\cos_0^{(-2)}\left(\frac{\pi}{2}\right) \approx \frac{\pi^2}{6^2}\frac{3}{40}\left(13\cos 0 + 36\cos\frac{\pi}{6} + 9\cos\frac{\pi}{3} + 2\cos\frac{\pi}{2}\right) \approx 1.00088,$$
$$\cos_0^{(-3)}\left(\frac{\pi}{2}\right) \approx \frac{\pi^3}{6^3}\frac{9}{80}\left(12\cos 0 + 27\cos\frac{\pi}{6} + \cos\frac{\pi}{2}\right) \approx 0.5714$$

with relative errors 0.08789% and 0.10552%, respectively.

The expressions in the second column of (B.1) can be estimated with the weights and coefficients

$$\begin{array}{lllll} w_{-1,0}=1, & w_{-1,1}=-1, & w_{-1,2}=-1, & w_{-1,3}=1, & A_{-1}=\dfrac{3}{2}, \\ w_{-2,0}=-3, & w_{-2,1}=9, & w_{-2,2}=-9, & w_{-2,3}=3, & A_{-2}=1, \end{array}$$

derived as in (3.2). We obtain

$$\int_0^{\pi/2} \frac{\mathrm{d}^2}{\mathrm{d}x^2}\cos x\,\mathrm{d}x \approx \frac{6}{\pi}\frac{3}{2}\Big(\cos 0-\cos\frac{\pi}{6}-\cos\frac{\pi}{3}+\cos\frac{\pi}{2}\Big) \approx -1.04859,$$
$$\int_0^{\pi/2} \frac{\mathrm{d}^3}{\mathrm{d}x^3}\cos x\,\mathrm{d}x \approx \frac{6^2}{\pi^2}\Big(-3\cos 0+9\cos\frac{\pi}{6}-9\cos\frac{\pi}{3}+3\cos\frac{\pi}{2}\Big) \approx 1.07322$$

with relative errors 4.85855% and 7.32174%, respectively.

Appendix C. Boundary value problem example

Consider the situation described at the end of Section 5. Put $t_0=0$, $h=\frac{1}{10}\pi$ and $t_j=t_0+jh$ for $j=0,1,\dots,5$. Let f be an unknown function with known values

$$\begin{array}{lllll} f^{(-2)}(t_0)=-1, & f^{(-1)}(t_0)=0, & f(t_0)=1, & f^{(1)}(t_0)=0, & f^{(2)}(t_0)=-1, \\ f^{(-2)}(t_5)=0, & f^{(-1)}(t_5)=1, & f(t_5)=0, & f^{(1)}(t_5)=-1, & f^{(2)}(t_5)=0. \end{array}$$

Our goal is to estimate the values $f(t_j)$ for $j=1,2,3,4$. Typically, one would solve this problem by setting $g:=f^{(-2)}$ and estimating g with Hermite interpolation polynomial H of degree 9. The desired values $f(t_j)$ would be approximated as $H''(t_j)$.

Since we are only interested in the values of f at the specific nodes t_j, instead of generating a (potentially unstable) high degree polynomial over the entire interval $[t_0,t_5]$, we can use the approach described in Section 5. Using the weights and coefficients from Table 3 (case $k=5$), with the shorthand notation f_j for $f(t_j)$, we have

$$\int_{t_0}^{t_5} \frac{\mathrm{d}^3}{\mathrm{d}x^3} f(x)\,\mathrm{d}x \approx \frac{5}{12h^2}(-11f_0+43f_1-74f_2+74f_3-43f_4+11f_5),$$
$$\int_{t_0}^{t_5} \frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x)\,\mathrm{d}x \approx \frac{5}{12h}(5f_0-9f_1+4f_2+4f_3-9f_4+5f_5),$$
$$f_{t_0}^{(-1)}(t_5) \approx \frac{5h}{288}(19f_0+75f_1+50f_2+50f_3+75f_4+19f_5),$$
$$f_{t_0}^{(-2)}(t_5) \approx \frac{25h^2}{2016}(122f_0+475f_1+100f_2+250f_3+50f_4+11f_5).$$

On the other hand, we can express the above as

$$\begin{aligned}\int_{t_0}^{t_5} \frac{\mathrm{d}^3}{\mathrm{d}x^3} f(x)\,\mathrm{d}x &= f^{(2)}(t_5) - f^{(2)}(t_0) = 1,\\ \int_{t_0}^{t_5} \frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x)\,\mathrm{d}x &= f^{(1)}(t_5) - f^{(1)}(t_0) = -1,\\ f_{t_0}^{(-1)}(t_5) &= f^{(-1)}(t_5) - f^{(-1)}(t_0) = 1,\\ f_{t_0}^{(-2)}(t_5) &= f^{(-2)}(t_5) - f^{(-2)}(t_0) - f^{(-1)}(t_0)(t_5 - t_0) = 1.\end{aligned}$$

This leads to the system

$$\begin{pmatrix} 43 & -74 & 74 & -43\\ -9 & 4 & 4 & -9\\ 75 & 50 & 50 & 75\\ 475 & 100 & 250 & 50 \end{pmatrix}\begin{pmatrix} f_1\\ f_2\\ f_3\\ f_4\end{pmatrix} \approx \begin{pmatrix} \dfrac{12h^2}{5} + 11f_0 - 11f_5\\ -\dfrac{12h}{5} - 5f_0 - 5f_5\\ \dfrac{288}{5h} - 19f_0 - 19f_5\\ \dfrac{2016}{25h^2} - 122f_0 - 11f_5 \end{pmatrix}$$

with the solution $(f_1, f_2, f_3, f_4) \approx (0.95108, 0.80899, 0.58777, 0.30904)$. Note that the function $f(x) = \cos x$ meets the original boundary conditions and we have

$$(\cos t_1, \cos t_2, \cos t_3, \cos t_4) \approx (0.95106, 0.80902, 0.58779, 0.30902),$$

so the error is around $2 \cdot 10^{-5}$.

References

[1] *M. Abramowitz, I. A. Stegun* (eds.): Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. U.S. Department of Commerce, Washington, 1964. zbl MR

[2] *O. A. Bauchau, J. I. Craig*: Euler-Bernoulli beam theory. Structural Analysis. Solid Mechanics and Its Applications 163. Springer, Dordrecht, 2009, pp. 173–221. doi

[3] *R. L. Burden, J. D. Faires*: Numerical Analysis. PWS Publishing Company, Boston, 1993. zbl

[4] *G. B. Folland*: Advanced Calculus. Prentice Hall, Hoboken, 2001.

[5] *P. Holoborodko*: Stable Newton-Cotes Formulas. Available at `http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/stable-newton-cotes-formulas/`.

[6] *A. Janečka, V. Průša, K. R. Rajagopal*: Euler-Bernoulli type beam theory for elastic bodies with nonlinear response in the small strain range. Arch. Mech. *68* (2016), 3–25. zbl MR

[7] *V. K. M. Selvam, K. R. Bindhu*: Application of double integration method and the Maxwell-Betti theorem for finding deflection in determinate flexural frames: A supplement note. J. Struct. Eng. *41* (2014), 420–431.

[8] *K. Tvrdá*: Solution of a high bridge pillar under wind effects taking into account the real properties of reinforced concrete. MATEC Web Conf. *313* (2020), 6 pages. doi
[9] *K. Tvrdá, M. Minárová*: Computation of definite integral over repeated integral. Tatra Mt. Math. Publ. *72* (2018), 141–154. zbl MR doi

Authors' addresses: Katarína Tvrdá, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 810 05 Bratislava, Slovakia, e-mail: `katarina.tvrda@stuba.sk`; Peter Novotný (corresponding author), University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia e-mail: `peter.novotny@fri.uniza.sk`.