Jordan Journal of Civil Engineering, Volume **18**, No. **3**, **2024**
DOI: https://doi.org/10.14525/JJCE.v18i3.04

Jordan Journal of Civil Engineering

Journal homepage: https://jjce.just.edu.jo

A Modified Numerical Integration Method: Superior Accuracy for Hard-Exponential Functions

Karim Numayr [1]***, Hashem Almashaqbeh*** [2]***, Madhar Haddad*** [3]****, Shehab Bani Hani*** [4]

[1] *Department of Civil Engineering, Isra University, Amman, Jordan. E-Mail: karim.s.numayr@gmail.com*
[2] *Department of Civil Engineering, Isra University, Amman, Jordan. E-Mail: H.almashaqbeh@iu.edu.jo*
[3] *Department of Architectural Engineering, United Arab Emirates University, Al Ain, UAE.*
** Corresponding Author. E-Mail: madhar@uaeu.ac.ae*
[3] *Department of Civil Engineering, Isra University, Amman, Jordan. E-Mail: Shehabahmedbanihani@gmail.com*

ARTICLE INFO

Article History:
Received: 16/3/2024
Accepted: 24/4/2024

ABSTRACT

A novel numerical integration method is presented, which uses the tangents at the beginning of the intervals in addition to the ordinates of the function to estimate the integral. The trapezoidal part of the area under the curve between tangent and abscissa is calculated exactly, while the area bounded between the curve and the tangent is estimated. This major reduction of the approximated part yields more accurate numerical-integration results than current methods. The approximation is carried out using a mapped simple power function and a correction function. The method is tested using several functions including hard exponentials, where most numerical-integration methods fail to attain accurate results with limited numbers of intervals, since the ordinates of such functions increased rapidly within a short boundary region. The proposed method is compared with the well-known Simpson's 1/3 rule, Gauss quadrature, in addition to the MATLAB quad function. The numerical results demonstrate that the proposed method is superior to these methods.

Keywords: Numerical integration, Quadrature, Hard-exponential functions, Newton–Cotes, MATLAB quad.

INTRODUCTION

Numerical integration, also known as quadrature, is an approximate solution to a convoluted integration that cannot be solved analytically (Savage and Peterson, 1996). It is one of the basic tools used to solve problems in many engineering fields, including chemical, mechanical, electrical and civil engineering (Savage and Peterson, 1996; Fister et al., 2018). The most popular numerical-integration methods are the Newton-Cotes and the Gaussian quadrature formulae. Both quadrature formulae replace the complicated mathematical functions, which are hard to be integrated, with a polynomial which is easy to be integrated (El-Mikkawy, 2003; Dehghan et al., 2005; Sermutlu, 2005). In any Newton-Cotes formula, the space between the abscissas must be equal, while in Gaussian quadrature, it is not necessary that the space between the abscissas be equal (Fister et al., 2018; Kress, 1998). The most common Newton-Cotes formulae are the trapezoidal rule, the Simpson 1⁄3 rule and the Simpson 3/8 rule (Chapra, 2015; Mettle et al., 2016). Simpson's rule and its adaptive variations are used ubiquitously in computing finite integration. These rules employ higher-order polynomials to connect the points used to calculate the integration. This results in a more accurate estimation of the integrals (Lyness, 1969; Davis and Rabinowitz, 2007; Chapra, 2015). An n-point Gaussian quadrature rule can provide the exact integration for a polynomial of 2n-1 order over a finite interval. The number and the width of the sub-intervals determine the level of accuracy. Moreover, more recent versions can provide additional accuracy due to the orthogonality properties of the Legendre polynomials. Generally, Gaussian quadrature can provide better accuracies, because the rules depend on the roots of the orthogonal polynomials. The most common Gaussian quadrature formulae include Gauss-Legendre, Gauss-Hermite and Clenshaw-Curtis quadratures (Fister et al., 2018).

The nature of the integrated function can play a significant role in the accuracy and precision of several numerical-integration methods. For instance, in some boundary-layer problems, a highly non-linear exponential function would be involved and many numerical-integration methods fail to attain accurate solutions, since the ordinates of such functions increase rapidly within a short boundary region. Hence, there is a pressing need to improve numerical-integration techniques to obtain more accurate results and improve computational time (Davis and Rabinowitz, 2007; Fister et al., 2018).

Over the past few years, several schemes have been presented to improve the efficiency of numerical-integration methods by either enhancing the available techniques or developing a new method. Mettle et al. (2016) proposed a new method based on the trapezoidal rule. At first, the main area is divided into several segments to mimic the trapezoidal rule. Then, the top-part area of each segment is sub-divided into rectangles and/or squares and triangles. The numerical value of the integration is the sum of the areas of these geometrical shapes, in addition to the area of the down part of each segment, which is usually a rectangle. Compared to some Newton-Cotes formulae, this proposed method can provide better estimates and reduce the total error. Dehghan et al. (2005) presented a new technique to improve the Newton-Cotes formulae. They considered the lower and the upper limits of the integral as additional two unknowns. As a result, the precision degree of the numerical-integration formula is improved to be (n+2) instead of (n+1) which is well-known for the Newton-Cotes formulae.

Another way to improve numerical-integration is the use of adaptive quadrature techniques. These form a numerical-integration approach that adaptively refines the sub-intervals of the region of integration. As a result, such techniques can remedy the coarseness in regions of high variability by adjusting the step size. Gander and Gautschi (2000) reviewed the basic principles of adaptive quadrature and presented two MATLAB quadrature programs. The first is an implementation of the adaptive recursive Simpson rule, while the second is based on a four-point Gauss-Lobatto formula and two successive Kronrod extensions. Their presented "adaptsim" function has been included in MATLAB since then as the quadrature function (MathWorks, 2005). Many researchers have studied or developed numerical-integration algorithms using simple or adaptive techniques (Kahaner, 1972; Rice, 1975; Genz, 1991; Genz and Kass, 1997; Gerstner and Griebel, 1998; Krack and Köster, 1998; Mastroianni and Milovanovi, 2002; Kim et al., 2003; Kzaz and Prévost, 2003; Boal and Sayas, 2004; Woodford and Phillips, 2011; Galdino, 2012). Others used special or modified numerical-integration schemes in finite-element codes or to solve structural dynamics problems (Alnahhal and Aref, 2019; Numayr et al., 2022).

The MATLAB function quad, the main target of comparison, is an adaptive Simpson's quadrature scheme built in the MATLAB to evaluate the definite integral of a function f(x) over the limits (a,b). The ordinate values of the function ($y_i = f(x_i)$) are evaluated at points ($x = x_i$) in the abscissa. For each sub-interval (x_i, x_{i+2}) between the limits, this advanced algorithm compares the difference between the value of the integral using Simpson's 1/3 rule for the whole interval and the value of the integral using Simpson's 1/3 rule for combined half intervals (x_i, x_{i+1}) and (x_{i+1}, x_{i+2}) for a certain desired tolerance (ϵ). Then, if the desired accuracy is met, the value of the integral for the interval (x_i, x_{i+2}) is accepted and the recursion is stopped. Otherwise, the process is repeated for the left-half (x_i, x_{i+1}) and right-half (x_{i+1}, x_{i+2}) intervals. This recursion process continues until either the accuracy or termination criteria are met. The MATLAB (quad) function uses relative and absolute difference errors for accuracy criteria with an absolute-error tolerance of (TOL = 1.0E-6). The function also has a termination requirement, which is that the number of recursions should not exceed (1.0E+4) (Woodford and Phillips, 2011; Chapra, 2015).

Based on literature, most methods, both the classical and the modified ones, avoid discussion of the functions that have exponential growth at their boundaries, primarily the functions the asymptotic boundaries of which are a straight line at a certain value of x parallel to the y-axis. These functions either can be integrated with a great margin of error or cannot be integrated at all. Hence, the problems with integration of such functions can be summarized as follows: a) the higher the order the bigger the weights, b) the weights' signs alternate and lead to problems with rounding errors and c) an increasing-order sequence of Newton-Cotes approximations does not guarantee convergence to the true integral. For instance, it has been numerically proven that extending the Newton-Cotes formula is not necessarily fruitful. Moreover, using polynomials of much higher orders to approximate the integrated function does not necessarily improve the corresponding quadrature-rule approximation to the integral.

A novel numerical-integration method is therefore presented, which uses the tangents at the beginning of the intervals in addition to ordinate values at the beginning, end and middle of the intervals. The method calculates exactly a major part of the area under the curve, which is the trapezoid bounded by the tangent and the abscissa. This results in an enormous reduction of the approximated area under the curve. The estimation of the integral is limited to the area bounded by the curve and the tangent. For this purpose, the coordinates are rotated and scaled and a simple one-term power function is used, where the power is real and not an integer. The proposed method is used to estimate the integrals of several functions, such as polynomials, hard-exponential and trigonometric functions. The numerical-integration results are compared with estimates using Simpson's 1/3 rule, Gauss quadrature and the MATLAB quad function in order to evaluate the performance of the proposed method.

There are many applications for numerical-integration methods in many fields of applied sciences, such as engineering. Many applications of civil engineering require evaluated limited integrals. Most of the functions used in these applications have no exact integrals; hence, they have to be evaluated numerically. Moment-area theorem requires the evaluation of integrals of the following form to find the angle θ_{ab} between tangents at points (a) and (b) on the elastic curve of deformation of beams.

$$-\theta_{ab} = \int_a^b \frac{M(x)}{E(x)I(x)}dx \tag{1}$$

where, $M(x)$ is the equation depending on the load-function boundary conditions. The elastic modulus $E(x)$ could be a power, exponential or segmental functions in the case of functionally graded beams. The moment of inertia function, $I(x)$, is most likely cubic or higher-order polynomial for beams with variable cross-sections. Second moment – area theorem is used to evaluate the vertical distance $\Delta_{a/b}$ from point (a) on the elastic curve of deformation to the tangent from point (b).

$$\Delta_{a/b} = \int_a^b \frac{\bar{y}(x)M(x)}{E(x)I(x)}dx \tag{2}$$

where $\bar{y}(x)$ is the distance from centroid of area under $\frac{M(x)}{E(x)I(x)}$ diagram to point (a), being added to the integrand, which makes it more complicated to be exactly integrable.

Using the column-analogy method to solve problems of isotropic beams with variable cross-sections or functionally graded beams in the axial direction, the analogous force (F) for example is:

$$F = \int_0^L \frac{M(x)}{E(x)I(x)}dx. \tag{3}$$

Also, when using the virtual-work method or Castigliano's second theorem to solve indeterminate beams and frames, similar integrals are involved. For example, according to the virtual-work method, the deflection at redundant (i), Δ_i, due to the applied load on primary structure is:

$$\Delta_i = \int_0^L \frac{m(x)M(x)}{E(x)I(x)}dx \tag{4}$$

where the integral is over the length of beam or frame and $M(x)$ and $m(x)$ are the moment equations due to applied and unit loads, respectively, acting on the primary structure. In the analysis of arches with variable cross-sections using the virtual-work method,

$$\Delta_i = \int_0^L \frac{m(x)M(x)}{E(x)I(x)\cos(\tan^{-1}(\frac{dy}{dx}))}dx \tag{5}$$

where $y(x)$is the centerline of the arch equation that can be parabolic, circular or of another shape.

It is obvious that the integrand is complicated to be exactly evaluated. Fourier integrals in Navier solution of simply supported plates should be numerically evaluated for complicated loading functions, isotropic plates with variable cross-sections and plates functionally graded in the membrane directions.

Finite-element codes use quadrature numerical-integration which might lead to less accuracy in boundary-layer problems with very narrow boundary region. Therefore, to determine edge effect in composite laminate plate problems, a special exponential-shape function may be used and therefore a special numerical-integration method can be adopted to detect the sudden exponential growth in shear and normal stresses at the free edge.

The proposed numerical-integration method can be used with a relatively high accuracy compared to other numerical-integration schemes, for the above-mentioned applications, in structural mechanics. It can also be adopted in other fields where numerical integration is required.

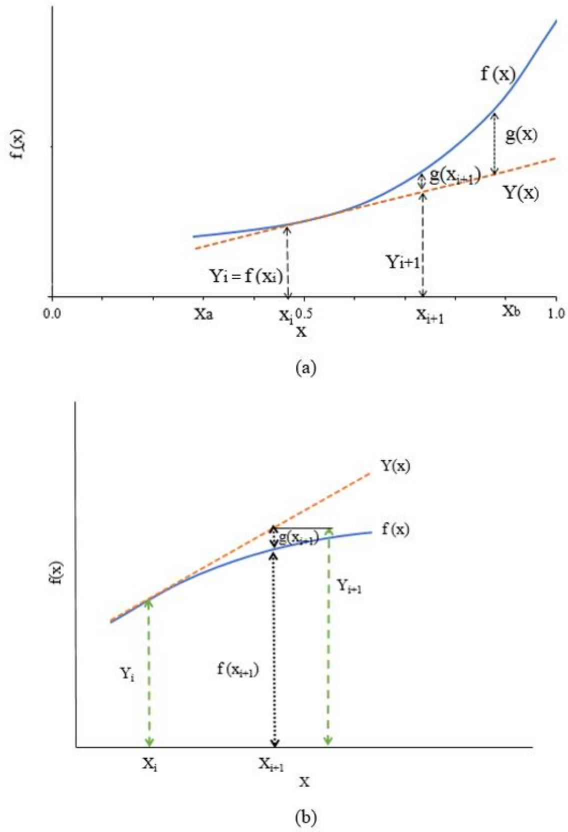


Figure (1): (a) Function *f (x)* defined in the domain (x_a, x_b). The curve is concave upward and (b) Function *f (x)* defined in the domain (x_a, x_b). The curve is concave downward

METHODOLOGY

As shown in Figure 1(a), if the curve is concave upward, the approximated area bounded by the curve and the tangent line is added to the trapezoid area to evaluate the definite integral of the function *f(x).* In Figure 1(b), the curve is concave downward and therefore, the area bounded by the tangent line and the curve is subtracted from the trapezoid area. This is done automatically from the sign of the difference function *g(x).* The tangent line at x_i, Y(x), is defined using the function and its derivative as follows:

$$Y(x) = \frac{df}{dx}(x - x_i) + Y_i \, . \tag{6}$$

The definite integral of the function *f(x)* in the limits (x_a,, x_b) is:

$$\int_{x_a}^{x_b} f(x)dx = \sum_{i=1}^{n} \int_{x_i}^{x_{i+1}} f(x)dx, \;\; x_1 = x_a \, , x_{n+1} = x_b \tag{7}$$

$$\int_{x_i}^{x_{i+1}} f(x)dx = \frac{\Delta x}{2}(Y(x_i) + Y(x_{i+1}) + \int_{x_i}^{x_{i+1}} g(x)dx \tag{8}$$

where $\Delta x = x_{i+1} - x_i$, $g(x) = f(x) - Y(x)$ and n

is the number of equally-spaced intervals between x_a and x_b.

The function *g(x)* is shown in Figure 2(a) *g(x)* on the original curve, (b) *g(x)* on its own. The integration will be approximated by adding the sum of three main terms for each interval.

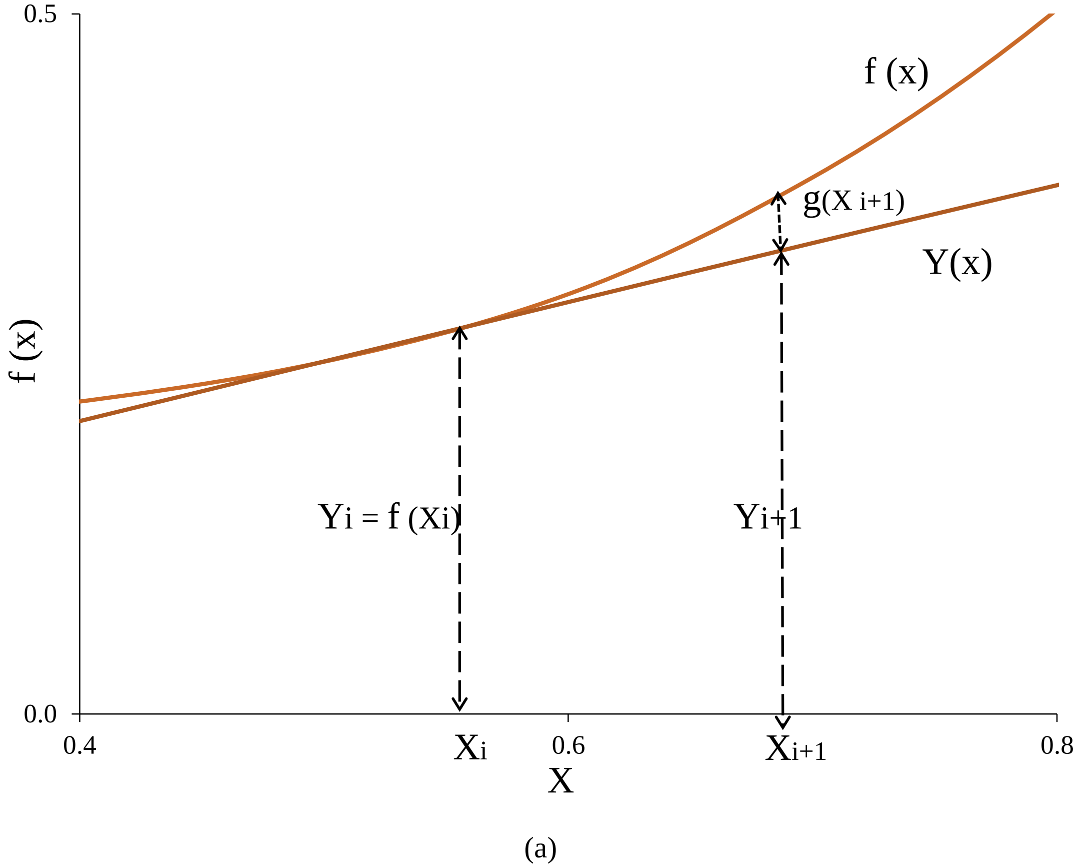


Figure (2): (a) *g(x)* in the original function *f(x)* and (b) *g(x)*

The first term is the exact value of trapezoid area below the tangent. The second term is the area bounded between the curve and the tangent line. The third term is a correction term that will be defined later. It should be noted here that the trapezoidal area is simply the integral of the first two terms of Taylor series expansion using the exact value of the derivative at x_i, while the trapezoidal rule uses approximate values of the first derivative. The second term evaluates the integral of the difference function that is the area bounded by the function and the tangent. It uses one higher-order term (x^r), where (r) is not an integer value, unlike Simpson's 1/3 rule which uses the third quadratic term of Taylor's series expansion and unlike other methods that use higher-order interpolation. Figure 1(a) shows the trapezoid and the approximated term of the integral that is:

$$\int_{x_i}^{x_{i+1}} g(x)dx = \Delta x * g(x_{i+1}) . \int_0^1 Q(z)dz. \quad (9)$$

The function $g(x)$ is divided by its value at $x = x_{i+1}$ so as $\frac{g(x)}{g(x_{i+1})} = 1$ at $x = x_{i+1}$. A transformation function, *Q(z)*, is defined in the domain $(0 \le z \le 1)$ using $g(x)$; that is:

$$Q(z) = \frac{g(z*\Delta x + x_i)}{g(x_{i+1})} \quad (10)$$

where $z = \frac{x - x_i}{\Delta x}$, $x = z * \Delta x + x_i$ and $dx = \Delta x . dz$.

The purpose of using *Q(z)* is to simplify its approximation and to facilitate the calculation of the correction term. Therefore,

$$\int_{x_i}^{x_{i+1}} g(x)dx = \Delta x . g(x_{i+1}) . \int_0^1 \frac{g(z*\Delta x + x_i)}{g(x_{i+1})} dz. \quad (11)$$

$\bar{Q}(z)$ is an interpolated function that satisfies *Q(z)* at the beginning, middle and end of the interval *(0,1)*, in addition to that both functions have a zero derivative at (*z=0*) and can be defined as:

$$\bar{Q}(z) = z^r \quad (12)$$

Integration of both sides yields:

$$\int_0^1 \bar{Q}(z)dz = \int_0^1 z^r\, dz = \frac{z^{r+1}}{r+1}\Big|_0^1 = \frac{1}{r+1} \quad (13)$$

where *r* can be evaluated using midpoint (*z=0.5*), since $\bar{Q}(z)$ equals *Q(z)* at *z=0, z=0.5, z=1 and* $\frac{dQ}{dz} = 0$ at *z=0* for any value of *r*. Therefore,

$$\bar{Q}(0.5) = Q(0.5) \quad (14)$$

$$0.5^r = \frac{g(0.5*\Delta x + x_i)}{g(x_{i+1})} \quad (15)$$

$$r = \frac{\ln(g(0.5*\Delta x + x_i)/\, g(x_{i+1}))}{\ln(0.5)}. \quad (16)$$

$\bar{Q}(z)$ and $Q(z)$ do not match in the left interval $(0 < z < 0.5)$ and the right interval $(0.5 < z < 1)$, as shown in Figure 3. The maximum difference $Q(z) - \bar{Q}(z)$ occurs in the vicinity of z = *0.75* for the right interval, while the maximum difference is not so close to the middle, as shown in difference function plot, Figure 4. A reasonable interpolation function for the left half (0,0.5) is a cubic polynomial that satisfies the end conditions of zero-value of the function at 0 and 0.5, zero slope at 0 and the known value at the middle: $Q(0.25) - \bar{Q}(0.25)$. Another reasonable quadratic polynomial is used for the interval (0.5,1) that satisfies conditions of zero-value of the function at 0.5 and 1, in addition to the known value of the difference function at the middle of this interval: $Q(0.75) - \bar{Q}(0.75)$. Evaluation of definite integrals of both polynomials yields the same factor of (2/3). Therefore, a correction term can be estimated from the differences $Q(0.25) - \bar{Q}(0.25)$ and $Q(0.75) - \bar{Q}(0.75)$ as follows.

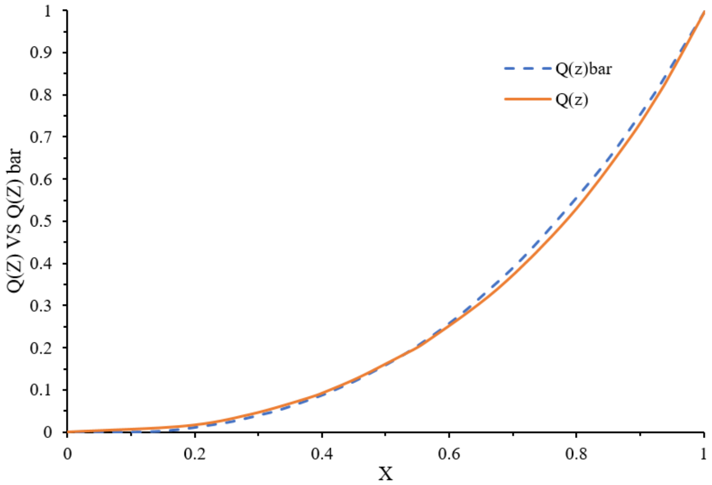


Figure (3): Functions *Q(z) and* $\bar{Q}(z)$

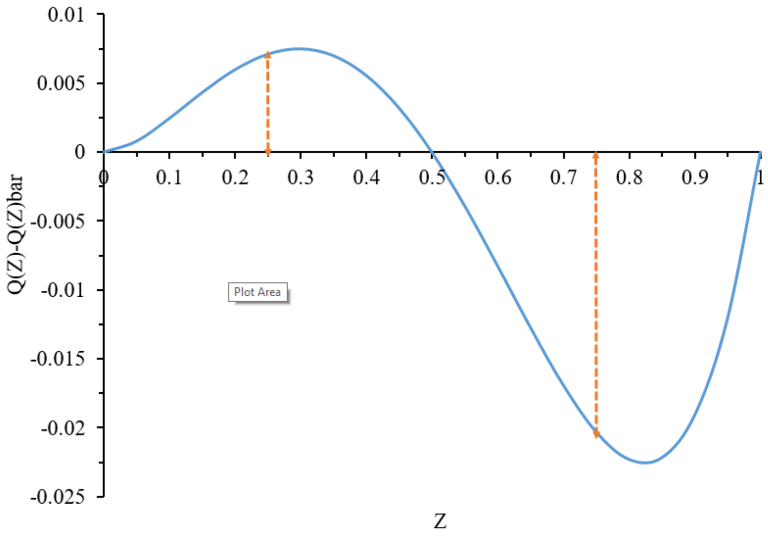


Figure (4): Difference function $Q(z) - \bar{Q}(z)$

The last term below is the correction term, which provides a minor contribution to the accuracy for small intervals, while it should be accounted for in large-size intervals. Including the correction term, the integral becomes:

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{\Delta x}{2}\left[Y(x_i) + Y(x_{i+1}) + g(x_{i+1})\left(\frac{2}{r+1} + \frac{2}{3}\left(Q(0.25) - \bar{Q}(0.25) + Q(0.75) - \bar{Q}(0.75)\right)\right)\right] \quad (17)$$

NUMERICAL EXAMPLE

A numerical example to estimate $\int_2^4 f(x)\,dx$ for f(x)= e^x in the limits (2,4) is carried out as follows.

$x_i = 2$, $x_m = 3$, $x_{i+1} = 4$, at these points $f(x_i) = 7.3890$, $f(x_m) = 20.0855$, $f(x_{i+1}) = 54.5982$ and

$Y(x_i) = 7.3890$, $Y(x_m) = 14.7781$, $Y(x_{i+1}) = 22.1672$ respectively. The difference between the function and the tangent line is $g(x_i) = 0$, $g(x_m) = 5.3074$, $g(x_{i+1}) = 32.4310$. Therefore, $Q(0.5) = \bar{Q}(0.5) = 0.1636$ and $r = 2.6113$. The value of the integral without and including the correction term are 47.5170 and 47.2316, respectively. Since the exact value is 47.2091, the proposed-method relative error is 0.0004767.

RESULTS AND DISCUSSION

The proposed method is compared to Simpson's 1/3 rule in that it yields a much better accuracy with a smaller number of intervals, especially for higher-order polynomials, trigonometric and exponential functions. It yields accurate results while Simpson's 1/3 rule fails to provide acceptable accuracy. It is compared to Gauss-quadrature integration, since the latter is used in finite-element codes and yields a better accuracy than other methods. It is demonstrated that quadrature integration yields inaccurate results for certain functions. The proposed numerical-integration method is also compared to the adoptive Simpson's rule function (quad) of the MATLAB. Even though the quad function is very powerful compared to other numerical schemes, it fails to provide accurate results for hard-exponential functions, while the proposed method yields accurate results.

Several functions are used to examine the proposed numerical-integration method. These functions are divided into five main groups. The first group, Table 1, includes the most used functions, such as polynomials, exponentials and trigonometric functions. This group is prepared to compare the proposed method with Simpson's 1/3 rule and Gauss quadrature with two points. In the second group, Figure 5, the limits of the integrals are kept fixed, while the number of intervals is changed from 2 to 10 to investigate the accuracy and the computational time of the proposed method. The third group, Figure 6, compares the proposed method with Newton-Cotes methods of orders n = 2, 4 and 6 and Gauss quadratures of orders 2, 4 and 6 by changing the number of points from 0 to 2500 to evaluate the performance of the proposed method with higher-order numerical-integration formulae. The fourth group, Table

2, compares the results of the proposed method with Simpson's 1/3 rule and Gauss quadrature with two points for exponential functions which are analytically integrable. The fifth group, Table 3, aims to test the proposed method with the hard-exponential functions, which are analytically non-integrable. For this purpose, the MATLAB numerical-integration function (quad) is employed.

Table 1. Integration of x^7, e^x, cos(x) and sin(x) using the proposed method, Simpson's 1/3 rule and Gauss quadrature

2 intervals & 2-points Gauss quadrature (2 intervals)						
Function	Limits		Exact integral	Simpson's 1/3 rule	Gauss quadrature	The proposed method
	a	b		Absolute relative error	Absolute relative error	Absolute relative error
x^7	5	10	1.245117E+07	3.186275E-02	2.113290E-02	6.653484E-05
e^x	5	10	2.187805E+04	1.201122E-01	7.625087E-02	3.255055E-04
sin(x)	0	π/4	2.928932E-01	2.115925E-07	1.409920E-07	6.366071E-08
sin(x)	π/4	π/2	7.071068E-01	2.115276E-07	1.410570E-07	1.348701E-08
cos(x)	0	π/4	7.071068E-01	2.115276E-07	1.410570E-07	1.506228E-08
cos(x)	π/4	π/2	2.928932E-01	2.115925E-07	1.409920E-07	9.750016E-08

In Table 1, the proposed method is compared with the well-known Simpson's 1/3 rule and Gauss quadrature by integrating the most common functions including x^7, e^x, cos(x) and sin(x). The integral limits for x^7 and e^x are selected to be from 0 to 5, while for the trigonometric functions, the integral limits are selected to be from 0 to $\pi/4$ and from $\pi/4$ to $\pi/2$. Two intervals were used for all the methods to provide a fair comparison. Based on the absolute relative error, the numerical results clearly show that the proposed method is superior to Simpson's 1/3 rule and Gauss quadrature for all the selected functions. For example, the absolute relative error in the proposed method for the integral of x^7 from 5 to 10 is 6.653484E-05, while for the same integral, the absolute relative error is found to be 3.186E-02 and 2.11E-02 for Simpson's 1/3 rule and Gauss quadrature, respectively.

Moreover, the proposed method is tested by estimating the integral values for two other functions (x^{10} and e^x). This time, the integral limits are fixed, while the number of intervals is changed from 2 to 6.

The absolute relative errors in estimating a) $\int_5^{10} x^{10}\,dx$ and b) $\int_5^{10} e^x\,dx$ using the proposed method, Simpson's 1/3 rule and Gauss quadrature and by changing the number of intervals from 2 to 6, are shown in Figure 5. As may be seen, the proposed method achieves a better accuracy (a lower absolute relative error) than Simpson's 1/3 rule and Gauss quadrature regardless of the number of intervals used. The proposed method has been compared with a previous study that used Newton-Cotes methods of orders n = 2, 4 and 6 and Gauss quadratures of orders 2, 4 and 6 to estimate $\int_0^{100} \frac{1}{x}\,dx$ (Sermutlu, 2005). This comparison aims to evaluate the performance of the proposed method with higher-order numerical-integration formulae. In common practice, lower-order interpolation formulae are preferred, such as Simpson's, second-order, trapezoidal and linear, compared to higher-order Newton-Cotes or Gauss quadrature. When a piece-wise simpler interpolation function is used for the sake of evaluation of a limited integral, a larger number of intervals or ordinate points would compensate for the lost accuracy. The integral value is estimated using the two methods mentioned (Sermutlu, 2005), with different Newton-Cotes orders and Gauss quadrature points in addition to the proposed method using different numbers of points. To compare the different integration methods fairly, the following should be considered: accuracy, reliability, efficiency, ease of application and number of function evaluations necessary for a given tolerance. As a main comparison indicator, relative error is used instead of absolute error due to the possibility of large differences in the results that are not easily compared, especially when the ordinate of the function and its limited integral values are extremely high. Relative error, E, is computed using the following formula:

$$E = log_{10}|\frac{I-Q}{Q}| \tag{18}$$

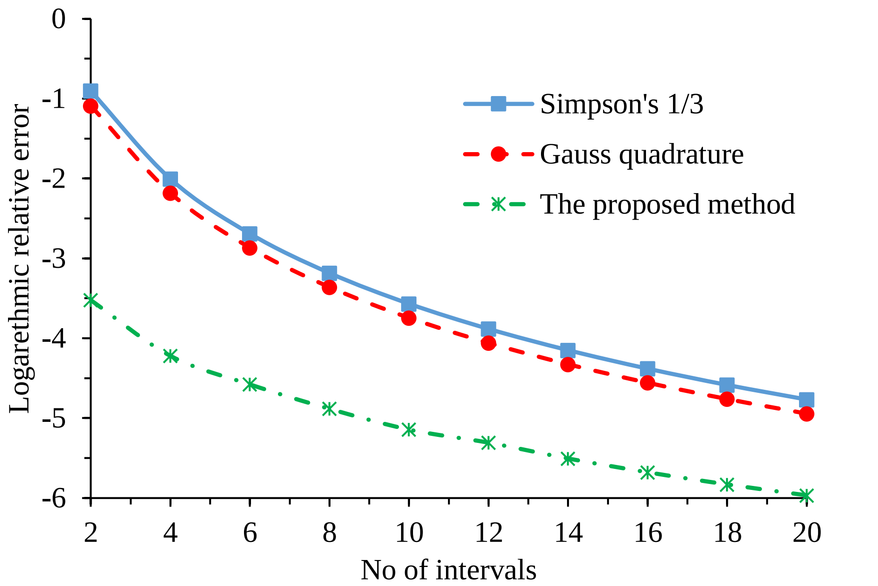
where, I is the estimate and Q is the exact value (Sermutlu, 2005).

Table 2. Numerical-integration results of analytically integrable exponential functions using the proposed method, Simpson's 1/3 rule and Gauss quadrature

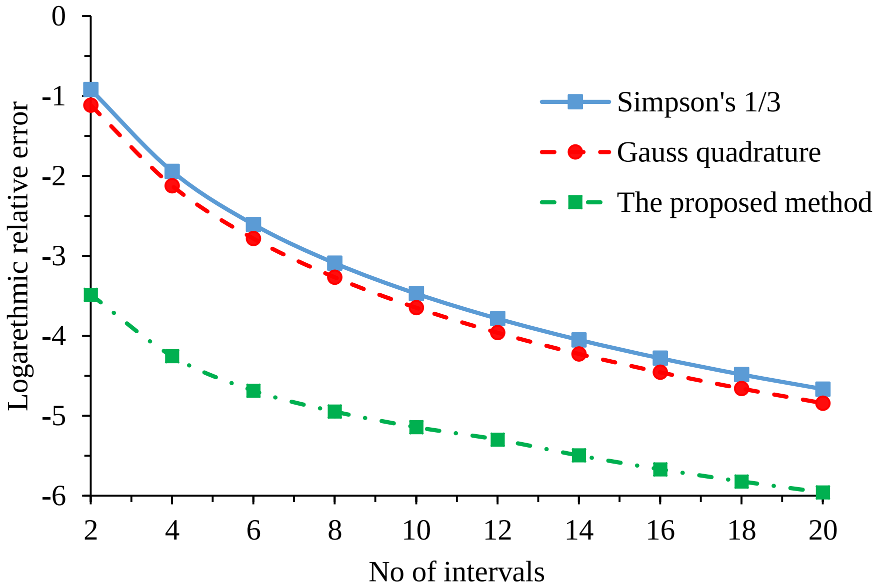
10 intervals & 2-point Gauss quadrature (10 intervals)						
Function	**Limits a**	**Limits b**	**Exact integral**	**Simpson's 1/3 rule Absolute relative error**	**Gauss quadrature Absolute relative error**	**The proposed method Absolute relative error**
$x*e^x$	0	10	1.996348E+05	7.039995E-03	4.636783E-03	4.651017E-05
$x*e^x$	0	20	9.858245E+09	6.943991E-02	4.473319E-02	6.166150E-05
$x*e^x$	0	30	3.794403E+14	2.243652E-01	1.388645E-01	1.094321E-03
$x*e^x$	0	40	1.338859E+19	4.584484E-01	2.689458E-01	3.244738E-03
$x*e^x$	0	50	4.425708E+23	7.420580E-01	4.078682E-01	5.370926E-03
$x*e^x$	0	60	1.382651E+28	1.052068E+00	5.368248E-01	5.600480E-03
x^2*e^x	0	10	1.823659E+06	9.683909E-03	6.388232E-03	5.660991E-05
x^2*e^x	0	20	1.898067E+11	8.072037E-02	5.184624E-02	1.172842E-04
x^2*e^x	0	30	1.120217E+16	2.449603E-01	1.509358E-01	1.113131E-03
x^2*e^x	0	40	5.321762E+20	4.854618E-01	2.831468E-01	3.488412E-03
x^2*e^x	0	50	2.207582E+25	7.726368E-01	4.217564E-01	5.608109E-03
x^2*e^x	0	60	8.288554E+29	1.084400E+00	5.490590E-01	5.570485E-03
x^3*e^x	0	10	1.682330E+07	1.296630E-02	8.540400E-03	8.463031E-05
x^3*e^x	0	20	3.666457E+12	9.301880E-02	5.955914E-02	1.565523E-04
x^3*e^x	0	30	3.312061E+17	2.663649E-01	1.633690E-01	1.240870E-03
x^3*e^x	0	40	2.116605E+22	5.129536E-01	2.974138E-01	3.746031E-03
x^3*e^x	0	50	1.101420E+27	8.034570E-01	4.355209E-01	5.780879E-03
x^3*e^x	0	60	4.969162E+31	1.116842E+00	5.610863E-01	5.442664E-03

In Figure 6, the logarithm of relative error (E) is shown as a function of the number of ordinate points used in estimating $\int_1^{100} 1/x\,dx$ using Newton-Cotes method of orders (2,4,6), Gauss quadrature with the number of points for each interval (2, 4, 6) (Sermutlu, 2005) and the proposed method. As may be seen from this figure, the results obtained using the proposed method are more accurate than those obtained using Newton-Cotes and Gauss quadrature, especially for fewer intervals or number of points from 0 to 400 (Figure 6(a)). It may also be seen that even for a large number of points, from 2000 to 2500 (Figure 6(b)), the proposed method yields a much better accuracy than the other methods. For large numbers of points, 6-point Gauss quadrature yields undesired oscillation in the error estimate and the continuity of slope of the error function was lost.

Table 2 shows the numerical-integration results of hard-exponential functions, which have known exact values, such as xe^x, x^2e^x *and* x^3e^x, using the proposed method, Simpson's 1/3 rule and Gauss quadrature. For the results presented in this table, the number of intervals was fixed at 10, while the upper integral limits were increased to demonstrate the effectiveness of the proposed method relative to other methods when the value of the function grows rapidly close to the upper limit (boundary region). In all the cases shown, the proposed method succeeds in achieving a much lower relative error than Simpson's 1/3 rule and Gauss quadrature. When the integration values become very large, Simpson's 1/3 rule and Gauss quadrature are not capable of estimating accurate results. The reason behind the accurate estimations obtained by the proposed method is because of the pronounced reduction in the estimated area under the curve, using a simple and reliable interpolation of the function relative to the tangent at the beginning of each interval, as well as proposing an appropriate correction term. Moreover, the problem of shooting of the function (sudden growth or decay of the function) in a very narrow boundary region is not a problem using the proposed method, whereas it is a major problem when using other methods. Hence, using other methods can yield high errors and a major loss of accuracy when the boundary width is less than the interval size. For example, the absolute relative error in the proposed method for the integral of x^3e^x from 0 to 60 is 5.44E-03, while for the same integral, the absolute relative error is found to be 5.61E-01 and 1.11 for Simpson's 1/3 rule and Gauss quadrature, respectively.

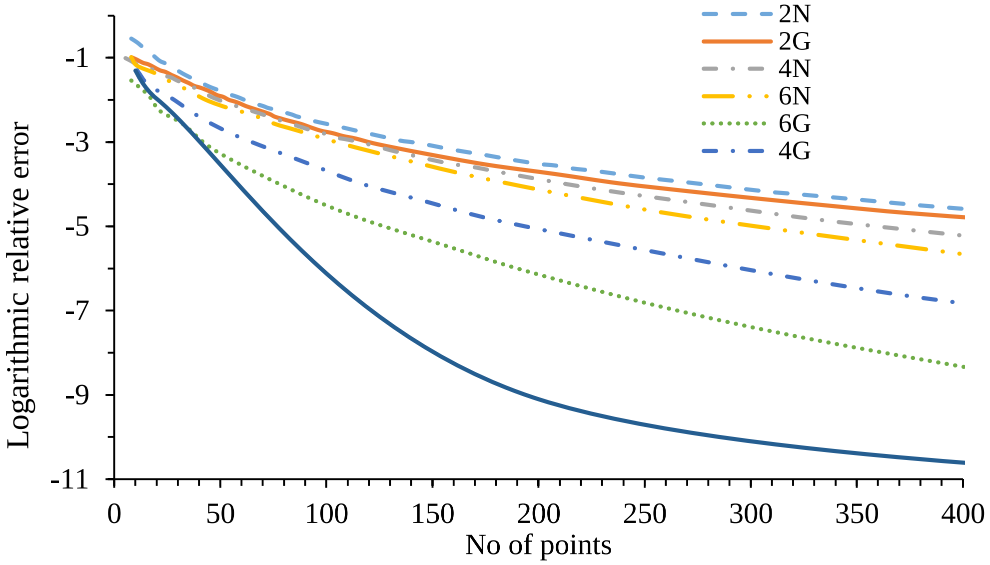


(a)

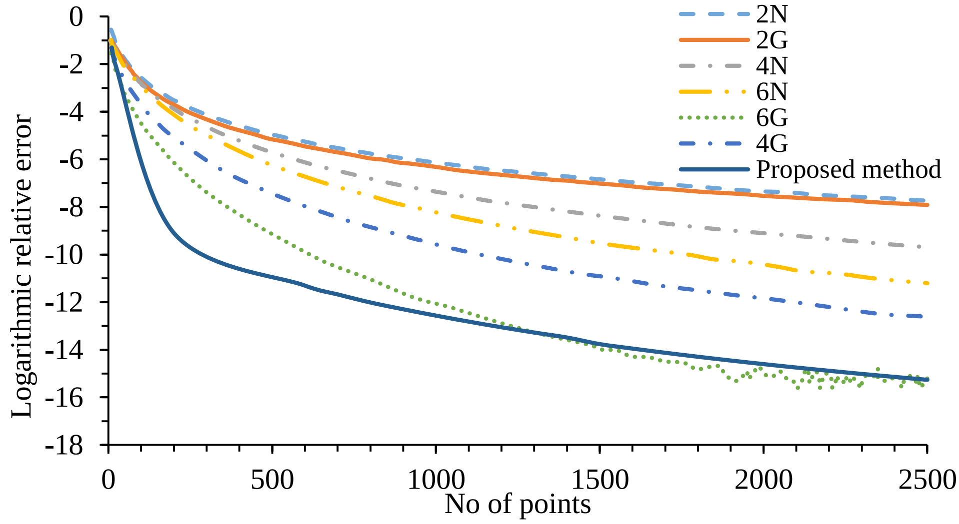


(b)

Figure (5): The logarithmic relative error in estimating (a) $\int_5^{10} x^{10}\,dx$ and (b) $\int_5^{10} e^{x}\,dx$ using the proposed method, Simpson's 1/3 rule and Gauss quadrature



(a)



(b)

Figure (6): The logarithmic relative error in estimating $\int_1^{100} 1/x\,dx$ using Newton-Cotes method of orders 2,4 and 6, Gauss quadrature method with numbers of points of 2, 4 and 6 for each interval and the proposed method for (a) number of points from 0 to 500 and (b) number of points from 500 to 2500

Table 3. Numerical-integration results of hard-exponential functions using the proposed method and MATLAB quad

Function	Limits		Power-series solution	Number of intervals	Quad	The proposed method
	a	b			Absolute relative error	Absolute relative error
e^{x^2}	0	10	1.350880E+42	5009	3.203284E+00	1.692766E-06
e^{x^2}	10	20	1.307000E+172	5039	7.438602E+00	4.081443E-06

e^{x^2}	0	26.5	1.816912E+303	5003	2.864594E+01	1.239677E-06
$x * e^{x^2}$	0	10	1.344059E+43	5004	3.224614E+00	4.539177E-09
$x * e^{x^2}$	10	20	2.610735E+173	5034	7.449156E+00	3.495117E-08
$x * e^{x^2}$	0	26.5	4.811383E+304	5008	2.866710E+01	1.416917E-06
$x^2 * e^{x^2}$	0	10	1.337304E+44	5009	3.245950E+00	1.238883E-07
$x^2 * e^{x^2}$	10	20	5.214935E+174	5034	7.459743E+00	2.948013E-08
$x^2 * e^{x^2}$	0	26.4	6.399472E+303	5005	2.846477E+01	1.382112E-06
e^{x^3}	0	2	2.768529E+02	88	1.728716E-10	1.258682E-06
e^{x^3}	2	4	1.312800E+26	5004	1.023341E+00	6.741032E-05
e^{x^3}	4	8.88	5.385400E+301	5036	2.336176E+01	6.698561E-06
$x^2 * e^{x^3}$	0	2	9.933193E+02	112	5.895989E-11	5.160159E-04
$x^2 * e^{x^3}$	2	4	2.078383E+27	5010	1.043713E+00	1.188839E-06
$x^2 * e^{x^3}$	4	8.8	3.036742E+295	5027	2.255489E+01	4.089096E-04
$x^3 * e^{x^3}$	0	2	1.895021E+03	126	1.416555E-10	4.959510E-08
$x^3 * e^{x^3}$	2	4	8.269700E+27	5004	1.054001E+00	8.365212E-06
$x^3 * e^{x^3}$	4	8.88	3.765600E+304	5028	9.999998E-01	2.634837E-03

Next, the results of the proposed method are compared to those of the MATLAB quad function. A summary of results of numerically integrating hard-exponential functions including e^{x^2}, xe^{x^2}, $x^2e^{x^2}$, e^{x^3}, $x^2e^{x^3}$, $x^3e^{x^3}$ is presented in Table 3. The MATLAB quad function is used with the default tolerance. Then, the number of intervals of the MATLAB quad function are found for each integration to use it with the proposed method. In most of the cases shown, the proposed method achieved a much lower relative error value compared to the MATLAB quad function. The proposed method is able to calculate a large integration value within a relatively short range (x_b-x_a). Comparatively, the quad function only succeeds in evaluating limited integration values, reaching singularity without getting close to the actual integration value and with a huge relative error. The quad function can achieve more accuracy and a low relative error only before the limits are suddenly changing into infinite. For example, for the integration of $x^3e^{x^3}$ from 0 to 2 using 126 intervals, the absolute relative error with the MATLAB quad function is found to be 1.416E-10, while the absolute relative error with the proposed method is found to be 4.95E-08 for the same integration. However, the proposed method is still capable of estimating accurate results for those cases. Thus, the proposed method is superior to the MATLAB quad function, especially when the value of the integrated function is changing suddenly into infinite. For example, integrating e^{x^2} from 0 to 26.5 using 5003 intervals from the default tolerance of the quad, the MATLAB function achieves an absolute relative error of 2.864E+01, while the proposed method gets a distinguished accuracy with an absolute relative error of 1.23968E-06.

The success of this method is due to the fact that the method follows the tangent of the function at different points, which allows the algorithm to lead where the function is heading and get closer to the function that needs to be approximated, while a big portion of the area is being exactly calculated using the trapezoid method. The interpolating-function calculation adds the majority of the resultant integration area. The correction term then tries to append any modifications needed to this value. These results are of extreme significance. The boundary-layer problem exists in numerous fields and applications. These applications would benefit from finding a solution to the actual integration with a low margin of error. These include finite-element codes for mechanical-engineering problems and various important stress and strain boundary-layer cases in civil engineering. Furthermore, numerous differential equations and partial differential equations with such boundary-layers would have better approximate solutions using this method.

CONCLUSIONS

A novel numerical-integration method is presented to estimate definite integrals of power, exponential and trigonometric functions. It is compared to other well-known numerical-integration methods; Simpson's quadrature and adaptive Simpson's quadrature (MATLAB quad), higher-orders Newton-Cotes and Gauss quadrature. The proposed method uses a simple sequence of calculations without the need for condition statements, except those required as stopping criteria. Therefore, a simple MATLAB code or even an excel sheet could do the job. However, MATLAB quad requires checking accuracy several times for each interval and therefore condition statements are used so many times. Adaptive Simpson's quadrature requires calling a sub-routine within itself; therefore, the MATLAB requires an external compiler, such as Fortran, C++ or others.

The proposed method succeeds in obtaining low relative error using fewer intervals or data points, while other prominent methods need many ordinate values and a much greater number of recursions for the same accuracy. The main aspect of the proposed method is estimating definite integrals of hard-exponential functions accurately, especially those that have high sudden growth or decay of ordinate values in a small boundary region. For those functions, MATLAB quad failed to get acceptable accuracy in some cases and failed to reach an answer in others. With the proposed method, the correction term is important in cases when few intervals are used. Future work would focus on improving the approximating function and/or the correction term. To improve accuracy, an adaptive technique can also be considered where tangents at the beginnings and ends of the intervals are employed with successive reach to the curve. This process would drastically reduce the approximated part of the area under the curve and improve accuracy. There are several real-world applications that would benefit from more accurate determination of boundary-layer problems.

Acknowledgements

The authors would like to express their gratitude to the civil-engineering departments of both Isra University and the UAE University for making this research possible.

REFERENCES

Alnahhal, W., and Aref, A. (2019). “Numerical evaluation of dynamic response by using modified Newmark’s method”. Jordan Journal of Civil Engineering, 13 (1), 30-43.

Boal N., and Sayas, F.-J. (2004). “Adaptive numerical integration on spherical triangles”. In: M.C. López de Silanes et al. (Eds.), Proceedings of VIII International Zaragoza-Pau Conference on Applied Mathematics and Statistics, Monografías Sem. Mat. G. Galdeano, 31, 61-69. Universidad de Zaragoza.

Chapra, S. (2015). “Applied numerical methods: With MATLAB for engineers and scientists”. McGraw Hill Education.

Davis, Oh.J., and Rabinowitz, Ph. (2007). “Methods of numerical integration”. Dover Publications, Second Edition. ISBN-10: 0486453391

Dehghan, M., Masjed-Jamei, M., and Eslahchi, M.R. (2005). “On numerical improvement of closed Newton-Cotes quadrature rules”. Applied Mathematics and Computation, 165 (2), 251-260. doi:10.1016/j.amc.2004.07.009

El-Mikkawy, M. (2003). “A unified approach to Newton-Cotes quadrature formulae”. Applied Mathematics and Computation, 138 (2-3), 403-413. doi:10.1016/S0096-3003(02)00144-3

Fister, M.W., Thapa, M., and Mulani, S.B. (2018). “Adaptive higher-order integration method and its application in uncertainty quantification”. In: AIAA Non-deterministic Approaches Conference, 2018, no. 209969. doi:10.2514/6.2018-1408

Galdino, S. (2012). “Interval integration revisited”. Open Journal of Applied Sciences, 2, 108-111. doi:10.4236/ojapps.2012.24B026

Gander, W., and Gautschi, W. (2000). “Adaptive quadrature-revisited”. BIT Numerical Mathematics, 40 (1), 84-101. doi:10.1023/A:1022318402393

Genz A., and Kass, R.E. (1997). "Sub-region-adaptive integration of functions having a dominant peak". Journal of Computational and Graphical Statistics, 6 (1), 92-111. doi:10.1080/10618600.1997.10474729

Genz, A. (1991). "An adaptive numerical-integration algorithm for Simplices". In: Lecture Notes in Computer Science (Including Sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 507 LNCS, 279-285. doi:10.1007/BFb0038504

Gerstner, T., and Griebel, M. (1998). "Numerical integration using sparse grids". Numerical Algorithms, 18 (3), 209-232, doi:10.1023/A:1019129717644

Kahaner, D.K. (1972). "Numerical quadrature by the ε-algorithm". Mathematics of Computation, 26 (119), 689. doi:10.1090/s0025-5718-1972-0329210-x

Kim, K.J., Cools, R., and Ixaru, L.G. (2003). "Extended quadrature rules for oscillatory integrands". Applied Numerical Mathematics, 46 (1), 59-73. doi:10.1016/S0168-9274(03)00009-6

Krack, M., and Köster, A.M. (1998). "An adaptive numerical integrator for molecular integrals". Journal of Chemical Physics, 108 (8), 3226-3234. doi:10.1063/1.475719

Kress, R. (1998). "Integral equations". Springer, New York, NY, 287-316.

Kzaz M., and Prévost, M. (2003). "Convergence acceleration of Gauss-Chebyshev quadrature formulae". Numerical Algorithms, 34 (2-4), 379-391. doi:10.1023/B:NUMA.0000005352.81665.7c

Lyness, J.N. (1969). "Notes on the adaptive Simpson quadrature routine". Journal of the ACM (JACM), 16 (3), 483-495. doi:10.1145/321526.321537

Mastroianni, G., and Milovanovi, G.V. (2002). "Weighted integration of periodic functions on the real line". Applied Mathematics and Computation, 128 (2-3), 365-378. doi:10.1016/S0096-3003(01)00080-7

MathWorks, Inc. (2005). "MATLAB: The language of technical computing." MathWorks, Accessed on: Apr. 02, 2021. [Online]. Available at: https://www.mathworks.com

Mettle, F., Quaye, E., Asiedu, L., and Darkwah, K. (2016). "A proposed method for numerical integration". British Journal of Mathematics & Computer Science, 17 (1), 1-15. doi:10.9734/bjmcs/2016/23048

Numayr, K.S., Ababneh A.N., and Almashaqbeh, H.K. (2022). "A finite-element method for modeling edge-effects in composite laminates with different ply-orientation". Jordan Journal of Civil Engineering, 16 (4), 549-566. doi:10.14525/JJCE.v16i4.39

Rice, J.R. (1975). "A metalgorithm for adaptive quadrature". Journal of the ACM (JACM), 22 (1), 61-82. doi:10.1145/321864.321870

Savage, J.S., and Peterson, A.F. (1996). "Quadrature rules for numerical integration over triangles and tetrahedra". IEEE Antennas and Propagation Magazine, 38 (3), 100-102. doi:10.1109/74.511963

Sermutlu, E. (2005). "Comparison of Newton-Cotes and Gaussian methods of quadrature". Applied Mathematics and Computation, 171 (2), 1048-1057. doi:10.1016/j.amc.2005.01.102

Woodford, C., and Phillips, C. (2011). "Numerical methods with worked examples: MATLAB edition". Springer Science & Business Media.