Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100101

On the Application of Multicomplex Algebras in Numerical Integration

S. Shirinkam [1,*], *A. Alaeddini*[2] *and H. R. Millwater*[2]

[1] Department of Mathematics, University of Texas at San Antonio, TX 78249, USA
[2] Department of Mechanical Engineering, University of Texas at San Antonio, TX 78249, USA

Received: 4 Jun. 2015, Revised: 2 Aug. 2015, Accepted: 3 Aug. 2015
Published online: 1 Jan. 2016

Abstract: In this paper, we propose a methodology to numerically integrate functions using multicomplex algebras and their corresponding matrix representations. The methodology employs multicomplex Taylor series expansion (MCTSE) to adaptively approximate and integrate a function using sufficiently small number of points. We investigate this methodology by presenting three different algorithms for various approximation strategies. We also use numerical studies to demonstrate the performance of the proposed methodology.

Keywords: Numerical Integration, Multicomplex Algebra, Taylor Series Expansion.

1 Introduction

Numerical integration, also referred to as numerical quadrature, constitutes a broad family of algorithms for calculating the numerical value of a definite integral [13]. There are several reasons for carrying out numerical integration. The integrand function may not be known at some points, it may be difficult or impossible to find an antiderivative, or it may be easier to compute a numerical approximation than to compute the antiderivative. Numerical integration methods can be generally described as combining evaluations of the integrand to get an approximation of the integral [1]. The integrand is evaluated at a finite set of integration points and a weighted sum of these values is used to approximate the integral. The integration points and weights depend on the specific method used and the accuracy required from the approximation.

There has been a large body of literature around numerical integration [9]. A large class of methods uses Newton-Cotes formulas, also known as quadrature formulas, which approximate the function with various degrees of polynomials evaluated at equally spaced points, of which the trapezoidal rule and Simpson's rule are among common examples [13]. Some of these methods have been integrated with Taylor series approximation as proposed in [3]. In addition, a generalization of the trapezoidal rule is Romberg integration, which can yield more accurate results for many fewer function evaluations [1]. Another group of quadrature formulas allow intervals between interpolation points to vary, which includes Gaussian quadrature formulas [5]. When the integrand is smooth, a Gaussian quadrature rule is typically more accurate than a NewtonCotes rule. Other quadrature methods with varying intervals include GaussKronrod and ClenshawCurtis quadrature methods [2] and [4].

The other group of quadrature, known as adaptive quadrature, approximates the function using static quadrature rules on adaptively refined subintervals of the integration domain [7]. Generally, adaptive algorithms are just as efficient and effective as traditional algorithms for "well behaved" integrands, but are also effective for "badly behaved" integrands for which traditional algorithms tend to fail. There are also other numerical integrations methods based on information theory, which have been developed to simulate information systems such as computer controlled systems, communication systems, and control systems [11].

An important part of the analysis of any numerical integration method is to study the behavior of the approximation as a function of the number of integrand evaluations. Generally, a method that yields a small error

* Corresponding author e-mail: sara.shirinkam@utsa.edu

for a small number of evaluations is considered superior, because reducing the number of evaluations of the integrand typically reduces the number of arithmetic operations involved, and therefore reduces the total round-off error.
In this paper, we propose a methodology to numerically integrate functions based on multicomplex Taylor series expansion (MCTSE) method [8,12] and [10]. The proposed framework adaptively increases the order of approximations and refines subintervals of the integration domain as necessary to reduce the number of function evaluations. Our methodology can be applied to the problems where the number of possible function evaluations is limited or evaluations are expensive, such as the finite element method (FEM). The methodology also has the convenient property of nesting, where integrand values can be re-used. In addition, the proposed framework has a modular and flexible structure, which allows its different components to be combined and integrated in several ways. In particular, we present three different methods based on proposed methodology and compare their performance. In the proposed framework, we begin with dividing the integration interval into n subintervals. Next, we use Taylor series expansion based on the MCTSE to approximate the function values at the endpoints and compare them with the actual values. Then, if the error is greater than a pre-specified threshold, we first consider higher degrees of the Taylor series and then make additional points as necessary. Finally, we use the information of the approximating Taylor series to integrate the function over the entire interval. Figure 1, shows the general scheme of the proposed framework.

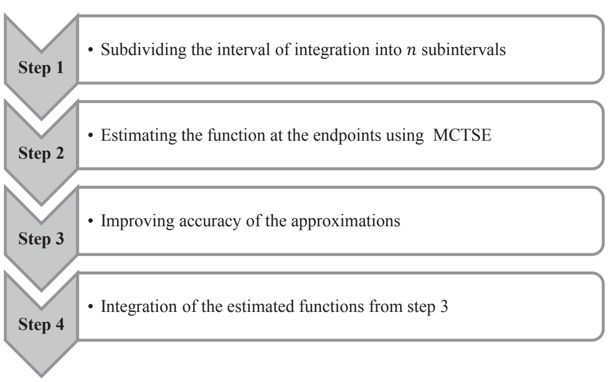

Fig. 1: The outline of the proposed methodology.

The rest of the paper is organized as follows: In section 2, we provide some necessary information about the MCTSE method for the calculation of various order of derivatives. We also describe the implementation of the MCTSE using corresponding matrix representation of multicomplex numbers. In section 3, we investigate our proposed methodology for computing numerical integration and we provide detail description of the three proposed methods. In section 4, we first provide an illustration example showing the steps of the proposed methods. Next, we evaluate the performance of the proposed methods in comparison with some of the existing methods in the literature using several numerical examples. Finally, in section 5, we present the conclusion and direction for future research.

2 Multicomplex Algebras for The Calculation of High order Numerical Derivatives

In this section, we describe the multicomplex Taylor series expansion (MCTSE) method for the calculation of high order derivatives[8]. MCTSE method uses the Taylor series expansion of $f(x+ih)$ around x as,

$$f(x+hi) = f(x) + hf'(x)i + O(h^2),$$

Then, the limit of the imaginary part of the Taylor series divided by h as h approaches zero gives the first derivative as follows,

$$\frac{\partial f(x)}{\partial x} = \lim_{h\to 0} \frac{Im_1[f(x+hi)]}{h} = \lim_{h\to 0} \frac{f'(x)h}{h},$$

The procedure can be generalized for calculating the n^{th} derivative using multicomplex numbers by perturbing the function in n directions of $i_1, \ldots, i_n$ as given,

$$\frac{\partial f^n(x)}{\partial x^n} = \lim_{h\to 0} \frac{Im_{1\ldots n}(f[x+hi_1+\ldots+hi_n)])}{h^n}. \quad (1)$$

MCTSE employs the matrix representation of a multicomplex number to calculate high order derivatives of analytic functions. The implementation of the method for the calculation of the first three derivatives has been shown here. Computation of the first three derivatives is possible through perturbing the function in all three directions of i_1, i_2 and i_3 in the tricomplex space. To find the first three derivatives, x is replaced with its perturbation, $\xi_x^{123} = x + hi_1 + hi_2 + hi_3$. The matrix representations of the tricomplex number ξ_x^{123} is given below:

$$\xi_x^{123} = x + hi_1 + hi_2 + hi_3 \leftrightarrow N_x^{123}$$
$$= \begin{Bmatrix} x & -h & -h & 0 & -h & 0 & 0 & 0 \\ h & x & 0 & -h & 0 & -h & 0 & 0 \\ h & 0 & x & -h & 0 & 0 & -h & 0 \\ 0 & h & h & x & 0 & 0 & 0 & -h \\ h & 0 & 0 & 0 & x & -h & -h & 0 \\ 0 & h & 0 & 0 & h & x & 0 & -h \\ 0 & 0 & h & 0 & h & 0 & x & -h \\ 0 & 0 & 0 & h & 0 & h & h & x \end{Bmatrix}$$

Then, we calculate $f(N_x^{123})$ which will be a $2^3 \times 2^3$ matrix. Based on the MCTSE, the first three derivatives can be calculated using equations 2-4.

$$\frac{\partial f(x)}{\partial x} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{21}}{h} = \frac{[f(N_x^{123})]_{31}}{h} = \frac{[f(N_x^{123})]_{51}}{h}, \tag{2}$$

$$\frac{\partial^2 f(x)}{\partial x^2} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{41}}{h^2} = \frac{[f(N_x^{123})]_{61}}{h^2} = \frac{[f(N_x^{123})]_{71}}{h^2}, \tag{3}$$

$$\frac{\partial^3 f(x)}{\partial x^3} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{81}}{h^3}. \tag{4}$$

The general formulation for the position and number of appearances of different order of derivatives in the MCTSE resulted matrix has been discussed in [8]. One may also use algorithm 2, to find the position of derivatives.

Algorithm. Position of derivatives in MCTSE matrix

Input: Order of derivative (d)
Output: Matrix position of derivatives (D)

Step1. $[D] = [\mathbb{0}]_{2\times d}$
Step 2. $[D]_{2,1} = 1$
For $i = 2$ *to* d
Step 3. $[D] = \begin{bmatrix} D \\ D \end{bmatrix}_{2^i \times d}$
Step 4. $[D]_{2^{i-1}+1\,:\,2^i,\ i} = 1$
End
Step 5. $[D_{sum}]_{2^d\times 1} = \underset{row-wise}{\text{sum}}[D]$

Fig. 2: Algorithm for the position of derivatives.

In the above algorithm, $D_{(j:l,i)}$, are the components in rows j to l and column i of matrix D. Also $SUM_{row-wise}(D)$ represents the summation of each row of D.
Example 2, describes the MCTSE for calculating the first three derivatives using algebraic properties of tricomplex numbers.

To calculate the first three derivatives of $f(x) = x^3$ using MCTSE and the equation 1, we have,

$$f'''(x) = \lim_{h\to 0} \frac{Im_{123}(f[x + hi_1 + hi_2 + hi_3])}{h^3},$$

Using the matrix representation of ξ_x^{123} and evaluating $f(N_x^{123})$ we have,

$$f(N_x^{123}) = \begin{bmatrix} x^3-9h^2x & 3hx^2 & 3hx^2 & 6h^2x & 3hx^2 & 6h^2x & 6h^2x & -6h^3 \\ 3hx^2-7h^3 & x^3 & -6h^2x & 3hx^2 & -6h^2x & 3hx^2 & 6h^3 & 6h^2x \\ 3hx^2-7h^3 & -6h^2x & x^3 & 3hx^2 & -6h^2x & 6h^3 & 3hx^2 & 6h^2x \\ 6h^2x & 3hx^2 & 3hx^2 & x^3 & -6h^3 & -6h^2x & -6h^2x & 3hx^2 \\ 3hx^2 & -6h^2x & -6h^2x & 6h^3 & x^3 & 3hx^2 & 3hx^2 & 6h^2x \\ 6h^2x & 3hx^2 & -6h^3 & -6h^2x & 3hx^2 & x^3 & -6h^2x & 3hx^2 \\ 6h^2x & -6h^3 & 3hx^2 & -6h^2x & 3hx^2 & -6h^2x & x^3 & 3hx^2 \\ 6h^3 & 6h^2x & 6h^2x & 3hx^2 & 6h^2x & 3hx^2 & 3hx^2 & x^3 \end{bmatrix}$$

Based on the equations 2-4, the elements in the second, third and fifth rows, first column of the $f(N_x^{123})$ divided by h show the first derivatives of the function as follow:

$$f'(x) = \lim_{h\to 0} \frac{[f(N_x^{123})]_{21}}{h} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{31}}{h} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{51}}{h} = \lim_{h\to 0} \frac{3hx^2}{h} = 3x^2$$

The elements in the fourth, sixth and seventh rows, first column of the $f(N_x^{123})$ divided by h^2 show the second derivatives of the function as follow:

$$f''(x) = \lim_{h\to 0} \frac{[f(N_x^{123})]_{41}}{h^2} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{61}}{h^2} = \lim_{h\to 0} \frac{[f(N_x^{123})]_{71}}{h^2} = \lim_{h\to 0} \frac{6h^2x}{h^2} = 6x$$

Finally, the element in the eighth row, first column of the $f(N_x^{123})$ divided by h^3 shows the third derivatives of the function as follow:

$$f'''(x) = \lim_{h\to 0} \frac{[f(N_x^{123})]_{81}}{h^3} = \lim_{h\to 0} \frac{6h^3}{h^3} = 6$$

3 Proposed Methodology

In this section we investigate numerical integration using multicomplex Taylor series expansion (MCTSE) method[5,6] and [8]. In particular, we present three different methods and compare them by given numerical examples investigating which method is more accurate and costs less. In all of the methods, we divide the interval of integration (a,b) into n subintervals. In the first method, n is considered an even positive integer, while in the second and third methods n can be any positive integer. Next, we approximate the function in each interval using the Taylor series of degree d about a point, namely c, in that interval. In the first method, c is the midpoint of each subinterval, while in the second and third methods c is the left endpoint of each subinterval. Also, in the first and third methods d starts from 2 and will be increased in each step using MCTSE, while in the second method d is a predetermined fixed number, e.g. 10. After that, we check if the difference between $f(x_i)$ and the approximated value of $\hat{f}(x_i)$, for all $0 \le i \le n$ is greater than a pre-specified threshold ε, e.g. 10^{-5}. If so, in the first and third method we increase d up to a specified number, e. g. 10, to make $|f(x_i) - \hat{f}(x_i)| \le \varepsilon$. If the difference is still greater than ε, we add a point at the center of the associated subinterval until the condition is satisfied. But in the second method, we only consider adding midpoints iteratively to make $|f(x_i) - \hat{f}(x_i)| \le \varepsilon$. Finally, to calculate $\int_a^b f(x)dx$ we integrate the resulting

Taylor series from the previous steps. Bellow we will discuss the details of each method one step at a time. Figure 3 also summarizes the major differences between the three proposed methods.

Method	Number of subintervals (n)	Degree of Taylor series (d)	Center of approximation (c), where error $> \varepsilon$, $d = 10$
Method 1	$n = 2k; k \in \mathbb{Z}^+$	Varies from 2 to 10	Midpoints
Method 2	$n \in \mathbb{Z}^+$	Fixed at 10	Right endpoints
Method 3	$n \in \mathbb{Z}^+$	Varies from 2 to 10	Right endpoints

Fig. 3: General view of the three proposed methods.

Method 1:
Method 1, improves the accuracy of the integral approximation using two strategies: (1) adaptively increasing the degree of Taylor series approximation, and (2) adaptively refining the interval of integration. It also employs the concept of Simpson's rule [9] which uses three points to define the subintervals. Consequently, it uses the Taylor series approximation based on MCTSE method around the midpoints to approximate the function at the left and right endpoints and estimate the integral in each subinterval. Figure 4, provides the framework of method 1 followed by the detailed description of each step.

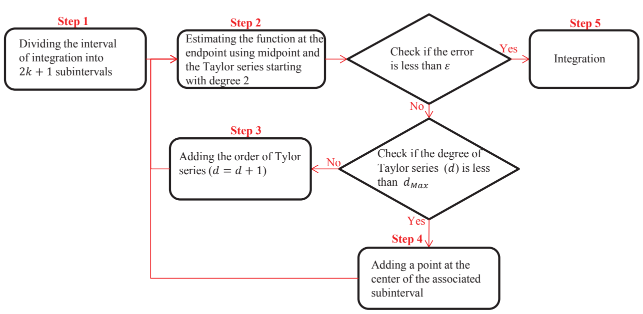

Fig. 4: The outline of method 1.

Step 1. Divide the interval of integration into $n = 2k$ subintervals. We divide the interval of integration (a,b) into $n = 2k, k \in Z^+$ equally spaced subintervals, $(x_0,x_2),(x_2,x_4),\ldots,(x_{2n-2},x_{2n})$.
Step 2. Estimate the function at the endpoint using midpoint and the Taylor series of degree 2. We employ the Taylor Series expansion about the center of (x_{i-2},x_i), $2 \leq i \leq n$, denoted by x_{i-1}, to approximate $f(x_{i-2})$ and $f(x_i)$, as given,

$$\hat{f}(x_{i-2}) \cong f(x_{i-1}) + f'(x_{i-1})(x_{i-2}-x_{i-1}) + \frac{f''(x_{i-1})}{2!}(x_{i-2}-x_{i-1})^2, \hat{f}(x_i) \cong f(x_{i-1}) + f'(x_{i-1})(x_i - x_{i-1}) + \frac{f''(x_{i-1})}{2!}(x_i - x_{i-1})^2,$$

where $f'(x)$ and $f''(x)$ are calculated using MCTSE. Figure 5, provides a graphical representation of method 1 step 2.

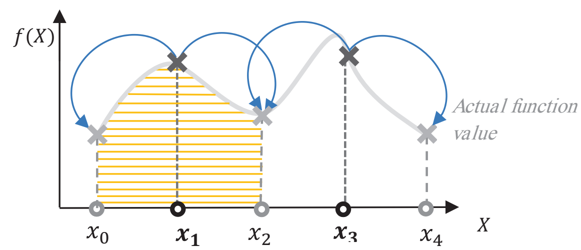

Fig. 5: An illustration of method 1 step 2: Estimate the function at the endpoints using the midpoint.

Step 3. Improve the accuracy of the approximation by increasing the degree of the Taylor series expansion. If at least one of $|\hat{f}(x_{i-2}) - f(x_{i-2})| > \varepsilon$ or $|\hat{f}(x_i) - f(x_i)| > \varepsilon$ is true, and the degree of the Taylor series, d, is less than d_{Max}, e.g. $d_{Max} = 10$. We increase d by 1 to make the errors less than ε, subjected to $d \leq 10$. Notably,we use MCTSE to calculate the various order of the derivatives .
Step 4. Improve the accuracy of the approximation by adding new points. If at least one of $|\hat{f}(x_{i-2}) - f(x_{i-2})|$ or $|\hat{f}(x_i) - f(x_i)|$ is greater than ε, we add a point r at the center of the associated subinterval and approximate the function around r. For instance if $|\hat{f}(x_{i-2}) - f(x_{i-2})| > \varepsilon$, we have the following expressions,

$$\hat{f}(x_{i-2}) \cong f(r) + f'(r)(x_{i-2}-r) + \frac{f''(r)}{2!}(x_{i-2}-r)^2 + \cdots + \frac{f^{(20)}(r)}{20!}(x_{i-2}-r)^{20}$$

We follow the preceding process for m times, $m < m_{max}$, where $m < m_{max}$ is the maximum number of added points, to make the difference between the estimated and actual value (error) less than ε. Figure 6, provides an illustration of method 1 step 4.

Step 5. Integrate. Considering the subinterval (x_{i-2},x_i) and the Taylor series approximation with only the linear term we have,

$$\int_{x_{i-2}}^{x_i} f(x_{i-1}) + f'(x_{i-1})(x - x_{i-1})dx = 2hf(x_{i-1}), \quad d = 1$$

adding the quadratic term to the Taylor series approximation we have,

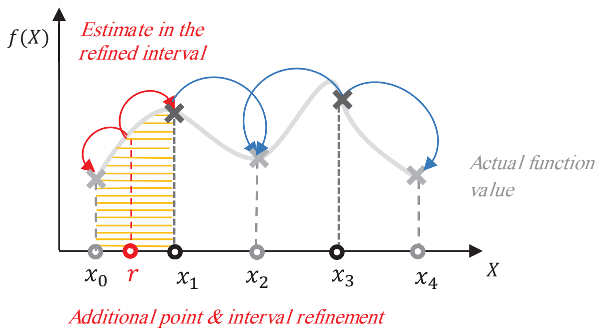

Fig. 6: An illustration of Method 1 step 4: Estimate the function at the endpoint using the midpoint.

$$\int_{x_{i-2}}^{x_i} f(x_{i-1}) + f'(x_{i-1})(x - x_{i-1}) + \frac{f''(x_{i-1})}{2!}(x - x_{i-1})^2 dx = 2hf(x_{i-1}) + \frac{f''(x_{i-1})}{(2!/2)} \times \frac{h^3}{3},\ d = 2$$

and with the cubic term we will get,

$$\int_{x_{i-2}}^{x_i} f(x_{i-1}) + f'(x_{i-1})(x - x_{i-1}) + \ldots + \frac{f'''(x_{i-1})}{3!}(x - x_{i-1})^3 dx = 2hf(x_{i-1}) + \frac{f''(x_i - 1)}{(2!/2)} \times \frac{h^3}{3}, \quad d = 3$$

continuing this process we can derive the general formulation of the integral for the interval (x_{i-2}, x_i) as,

$$\int_{x_{i-2}}^{x_i} f(x)dx = \sum_{k=1}^{\left\lfloor \frac{d_{max}}{2}+1 \right\rfloor} 2\frac{f^{(2k-2)}(r)}{(2k-2)!} \times \frac{h^{2k-1}}{2k-1},\ d \le d_{max} \tag{5}$$

One can use equation 5, for each subinterval and calculate the overall integral as, $\sum_{i=2}^{n} \int_{x_{i-2}}^{x_i} f(x)dx$. If only one of $|\hat{f}(x_{i-2}) - f(x_{i-2})|$ or $|\hat{f}(x_i) - f(x_i)|$ is less than ε, that is one subinterval has a midpoint but the other one does not, see Figure 6, we add a hypothetical midpoint at the center of the associated subinterval and approximate the function around that point using the Taylor series and employ the same procedure described above to integrate the function.

Method 2:

Method 2, starts with a Taylor series approximation of high degree to estimate the integral, but adaptively refines the interval of integration (by increasing the number of points) to improve the accuracy of integration. Method 2 defines its interval simply based on pairs of consecutive point, while using MCTSE around the right endpoints to approximate the function values at the left endpoints and consequently estimate the integral in each subinterval. Figure 7, provides the general framework of method 2 followed by detailed explanation of each step.

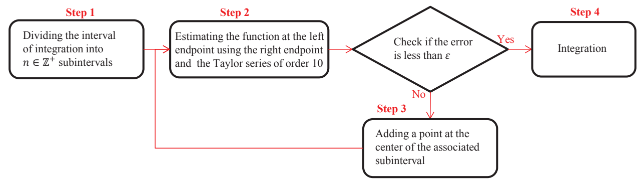

Fig. 7: The outline of method 2.

Step 1. Divide the interval of integration into $n \in Z^+$ subintervals. We divide the interval of integration (a, b) into $n \in Z^+$ equally spaced subintervals, $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$.

Step 2. Estimate the function at the left endpoint using the right endpoint and the Taylor series of degree 10. We employ the Taylor Series expansion about the right endpoint x_{i+1}, to approximate $f(x_i)$, as given,

$$\hat{f}(x_i) \cong f(x_{i+1}) + f'(x_{i+1})(x_i - x_{i+1}) + \ldots + \frac{f^{(10)}(x_{i+1})}{10!}(x_i - x_{i+1})^{10}$$

Note that, MCTSE is used to calculate the different order of derivatives.

Step 3. Improve the accuracy of the approximation by adding new points. If $|\hat{f}(x_i) - f(x_i)| > \varepsilon$ we add a point c at the center of (x_i, x_{i+1}) and approximate the function about c, as given,

$$\hat{f}(x_i) \cong f(c) + f'(c)(x_i - c) + \ldots + \frac{f^{(10)}(c)}{10!}(x_i - c)^{10}$$

We follow the preceding process for m times to make the difference between the estimated and actual values of the function less than ε.

Step 4. Integrate. To calculate $\int f(x)dx$ we integrate the Taylor series in step 2. So we have:

$$F(x) = f(c)(x_i - c) + f'(c)\frac{(x_i - c)^2}{2} + \ldots + \frac{f^{(10)}(c)}{10!}\frac{(x_i - c)^{11}}{11}$$

Therefore, $\int_{x_i}^{x_{i+1}} f(x)dx = F(x_{i+1}) - F(x_i)$.

We can repeat this process for each subinterval and calculate the overall integral as, $\sum_{i=2}^{n} \int_{x_{i-2}}^{x_i} f(x)dx$.

Method 3:

Method 3 has some commonalities with both Methods 1 and 2. Similar to Method 1, it uses both of the two strategies, increasing the order of Taylor series approximation, and refining the interval of integration. Meanwhile, like method 2 it uses a two point strategy to define and integrate the subintervals. Figure 8, provides the general framework of method 2 followed by detailed explanation of each step.

Step 1. Divide the interval of integration into n positive integer subintervals: Same as step 1 of method 2.

Step 2. Estimate the function at the left endpoint using the right endpoint and the Taylor series of degree 2. Same as step 2 of method 2, except that we start approximating using the second order Taylor series. Figure 9, provides a graphical representation of method 3 step 2.

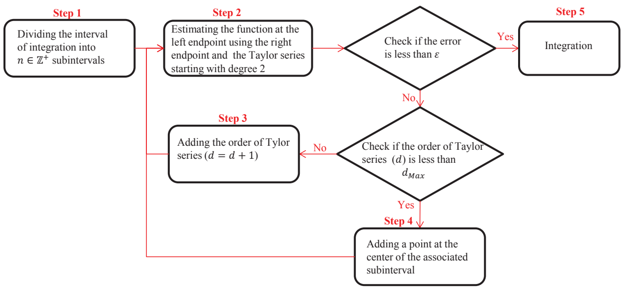

Fig. 8: The outline of method 3.

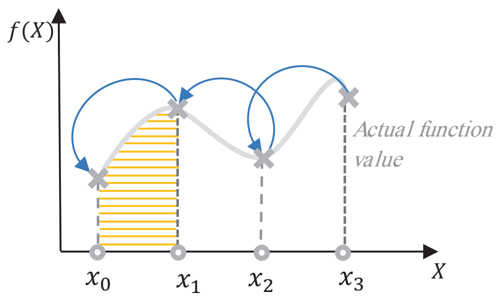

Fig. 9: An illustration of Method 3 step 2: Estimate the function at the left endpoint using the right endpoint.

Step 3. Improve the accuracy of the approximation by increasing the degree of the Taylor series expansion. If $|\hat{f}(x_i) - f(x_i)| > \varepsilon$ and the degree of the Taylor series, d, is less than d_{Max}, e.g. $d_{Max} = 10$, we increase d by 1 to make the error less than ε, subjected to $d \leq 10$. Notably, we use MCTSE to calculate the various order of the derivatives .

Step 4. Improve the accuracy of the approximation by adding new points. Same as step 3 of method 2. Figure 10, provides an illustration of method 3 step 4.

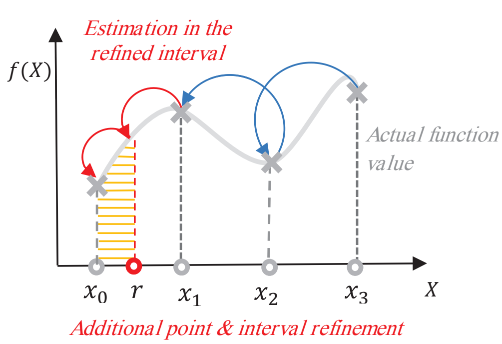

Fig. 10: An illustration of Method 3 step 4: Improve the accuracy of the approximation by adding new points.

Step 5. Integrate. Same as step 4 of method 2.

3.1 Error analysis and computational complexity

It can be easily shown that the error in approximating the integrals by the proposed framework and consequently the three methods discussed above is directly related to the predefined Taylor series approximation error, which is an input parameter for all the three methods. In other words, the accuracy of the integral can be easily determined by the setting the desirable approximation error, e.g. $\varepsilon = 10^{-5}$, which is also demonstrated in Section 4.2. In addition, since the error is related to the degree of Taylor series approximation and the number of subintervals, the proposed methods can be used to make an efficient trade-off between the two choices based on the practical limitations.

The proposed methodology adaptively uses high order derivatives and if necessary more points to improve the accuracy of integration, especially when the number of possible function evaluations is limited. Consequently, the proposed methods make a trade-off between the computational effort and number of function evaluations. For the computational complexity, MCTSE is the major contributor to the complexity of the proposed algorithm. While the authors are not aware of any analytical analysis of the complexity of MCTSE algorithm, using the corresponding matrix representations of the multicomplex number as proposed in this paper can significantly improve the speed of the algorithm specially for high order derivatives. In addition, the matrix form of multicomplex number, which includes the information of various order of derivatives, can be derived parametrically which eliminates the requirement of calling MCTSE algorithm multiple times for each points. Finally, since the MCTSE matrix is typically very sparse in high dimensions (high order derivatives), sparse matrix manipulation techniques can be used to further improve the processing time of the algorithm. Applying these tricks will make the computation time of the proposed methodology (specifically at high order derivatives) comparable to other numerical integration methods in the literature.

4 Numerical Examples

In this section we first show the steps of the three proposed methods using a simple example. Next, we evaluate the performance of the methods against some of the major existing numerical integration methods using different types of functions.

4.1 Illustrative Example

Here we present a simplified version of the proposed method steps for integrating $f(x) = x^4 + 2x$ between 0 and 1. For ease of illustration, we consider the following parameters: $m = 5$, $d_{Max} = 3$, $\varepsilon = 10^{-3}$ and $m_{max} = 5$. Table 1, illustrates the major steps for method 1. Section *I* of the table illustrates step 1 of method 1, which simply shows the function values at the sampled points along with other preliminary information. Section *II* presents the results for step 2 of method 1, estimating the function values at the endpoints using the midpoints. In particular, columns 2-6 of this section show the midpoints, i.e. 0.25 and 0.75 (*col. 2*), and the estimated function values at their left and right endpoints (*col. 5 and 6*) based on Taylor series approximation of degree 2(*col 4*). In addition, column 7-12 illustrates the actual function values at the endpoints (*col. 7 and 8*), the observed error (*col. 9 and 10*), and weather they satisfy the minimum allowed error (*col. 11 and 12*). Section *III* tabulates the step 3 of method 1, which increases the degree of Taylor series expansion. This section provides the same information as the preceding section based on Taylor series approximation of degree 3 (*col 4*). Section *IV* shows the step 4 of method 1, which refines subintervals of high approximation error by adding new (mid) points presented in *col 2* and recheck the error (*col. 11 and 12*). Note that approximation errors at this section satisfy the predefined minimum error limit. Finally, Section *V* illustrates the step 5 of the method 1, including the integral values at each subinterval along with their sum showing the integral over the interval $(0,1)$ (*col 13*).

Table 1: An illustrative example of method 1 steps

Sec.\Col ID.	2	3	4	5	6	7	8	9	10	11	12	13
I	**Step 1: Subdividing the interval of integration**											
	m	**x**	**f(x)**	**epsilon**	**d_max**							
	1	0	0.0000	10^-3	3							
	2	0.25	0.5039									
	3	0.5	1.0625									
	4	0.75	1.8164									
	5	1	3.0000									
II	**Step 2: estimating function values at endpoints using the midpoints**											
	x_m	**Step size**	**d**	**f^(x_m-step)**	**f^(x_m+step)**	**f(x-step)**	**f(x+step)**	**|f^-f|**	**|f^-f|**	**error > epsilon**	**error > epsilon**	
	0.25	0.25	2	0.0117	1.0430	0	1.0625	0.0117	0.0195	1	1	
	0.75	0.25	2	1.1055	2.9492	1.0625	3	0.0430	0.0508	1	1	
III	**Step3: increasing the degree of Taylor series expansion**											
	x_m	**Step size**	**d**	**f^(x_m-step)**	**f^(x_m+step)**	**f(x-step)**	**f(x+step)**	**|f^-f|**	**|f^-f|**	**error > epsilon**	**error > epsilon**	
	0.25	0.25	3	-0.0039	1.0586	0	1.0625	0.0039	0.0039	1	1	
	0.75	0.25	3	1.0586	2.9961	1.0625	3	0.0039	0.0039	1	1	
IV	**Step 4: Adding new points**											
	x_m	**Step size**	**d**	**f^(x_m-step)**	**f^(x_m+step)**	**f(x-step)**	**f(x+step)**	**|f^-f|**	**|f^-f|**	**error > epsilon**	**error > epsilon**	
	0.125	0.125	3	-0.0002	0.5037	0.0000	0.5039	0.000	0.000	0	0	
	0.375	0.125	3	0.5037	1.0623	0.5039	1.0625	0.000	0.000	0	0	
	0.625	0.125	3	1.0623	1.8162	1.0625	1.8164	0.000	0.000	0	0	
	0.875	0.125	3	1.8162	2.9998	1.8164	3	0.000	0.000	0	0	
V	**Step 5: Integration**											
	x_m	**Step size**	**d**	**f^(x_m-step)**	**f^(x_m+step)**	**f(x-step)**	**f(x+step)**	**|f^-f|**	**|f^-f|**	**error > epsilon**	**error > epsilon**	**integral**
	0.125	0.125	3	-0.0002	0.5037	0.0000	0.5039	0.000	0.000	0	0	0.0626
	0.375	0.125	3	0.5037	1.0623	0.5039	1.0625	0.000	0.000	0	0	0.1926
	0.625	0.125	3	1.0623	1.8162	1.0625	1.8164	0.000	0.000	0	0	0.3510
	0.875	0.125	3	1.8162	2.9998	1.8164	3	0.000	0.000	0	0	0.5848
											Sum	**1.1910**

Similarly, Table 2, illustrates the major steps of method 2 for the example discussed above. Section *I* illustrates the step 1 of the method 2, which includes subdividing the interval of integration. Section *II* presents the results for step 2 of method 2, estimating function values at each point (*col 5*) using the right endpoint and the Taylor series of maximum order, i.e. 3 (col 4). Additionally section II of the table shows the actual function values at the endpoints (*col. 6*), the observed error (*col. 7*), and weather they satisfy the minimum allowed error (*col.8*). Section *III* illustrates the step 3 of method 2, which refines subintervals of high approximation error by adding new (mid) points presented in *col 2* and check the approximation error (*cols. 7-8*). Note that approximation errors at this section satisfy the predefined minimum error limit. Finally, Section *IV* illustrates the step 4 of the method 2, integrating the function over the entire interval $(0,1)$ based on the sum of integrals at individual subintervals (*col 8*). Finally, Table 3, shows the major

Table 2: An illustrative example of Method 2 steps

Sec.\Col	2	3	4	5	6	7	8
I	**Step1:Subdividing the interval of integration**						
	m	**x**	**f(x)**	**epsilon**	**d**		
	1	0	0.0000	10^-3	3		
	2	0.25	0.5039				
	3	0.5	1.0625				
	4	0.75	1.8164				
	5	1	3.0000				
II	**Step2: estimating function values at each points using the adjacent point on the right**						
	x	**Step size**	**d**	**f^(x-step)**	**f(x-step)**	**|f^-f|**	**error >epsilon**
	0.25	0.25	3	-0.0039	0.0000	0.0039	1
	0.5	0.25	3	0.5000	0.5039	0.0039	1
	0.75	0.25	3	1.0586	1.0625	0.0039	1
	1	0.25	3	1.8125	1.8164	0.0039	1
III	**Step3: Adding points to the intevrals with error greater than epsilon**						
	x	**Step size**	**d**	**f^(x-step)**	**f(x-step)**	**|f^-f|**	**error >epsilon**
	0.125	0.125	3	-0.0002	0.0000	0.0002	0
	0.25	0.125	3	0.2500	0.2502	0.0002	0
	0.375	0.125	3	0.5037	0.5039	0.0002	0
	0.5	0.125	3	0.7695	0.7698	0.0002	0
	0.625	0.125	3	1.0623	1.0625	0.0002	0
	0.75	0.125	3	1.4023	1.4026	0.0002	0
	0.875	0.125	3	1.8162	1.8164	0.0002	0
	1	0.125	3	2.3359	2.3362	0.0002	0
IV	**Step 4: Integration**						
	x	**Step size**	**d**	**error >epsilon**	**F(x-step)**	**F(x)**	**integral**
	0.125	0.125	3	0	-0.0039	0.0079	0.0118
	0.25	0.125	3	0	-0.0196	0.0239	0.0435
	0.375	0.125	3	0	-0.0356	0.0403	0.0758
	0.5	0.125	3	0	-0.0525	0.0579	0.1104
	0.625	0.125	3	0	-0.0714	0.0777	0.1491
	0.75	0.125	3	0	-0.0937	0.1013	0.1950
	0.875	0.125	3	0	-0.1210	0.1304	0.2514
	1	0.125	3	0	-0.1555	0.1673	0.3229
						Sum	**1.1599**

steps of method 3. As for the preceding tables Section *I* presents the step 1 of the method 3, which includes subdividing the interval of integration. Section *II* shows the results for step 2 of method 2, estimating function values at each point (*col 5*) using the right endpoints and the Taylor series of order 2 (*col 4*). Similar to Table 2, section *II* of Table3, shows the actual function values at the endpoints (*col. 6*), the observed error (*col. 7*), and weather they satisfy the minimum allowed error (*col. 8*) as well. Section *III* tabulates the step 3 of method 3, which increases the degree of Taylor series expansion. In particular, this section provides the same information as the preceding sections of the table based on Taylor series approximation of degree 3 (*col 4*). Section *IV* illustrates the step 4 of method 2, which refines subintervals of high approximation error by adding new (mid) points presented in *col 2* and check the approximation error (*cols. 7-8*). Note that approximation errors at this section satisfy the predefined minimum error limit. Finally, Section *V* illustrates the step 5 of the method 2, integrating the function over the entire interval $(0,1)$ based on the sum of integrals at individual subintervals (*col 8*).

Table 3: An illustrative example of method 3 steps

Sec.\Col	2	3	4	5	6	7	8
	Step1:Subdividing the interval of integration						
	m	x	f(x)	epsilon	d		
	1	0	0.0000	10^-3	3		
	2	0.25	0.5039				
I	3	0.5	1.0625				
	4	0.75	1.8164				
	5	1	3.0000				
	Step 2: estimating function values at each points using the adjacent point on						
	x	Step size	d	f^(x-step)	f(x-step)	|f^-f|	error >epsilon
	0.25	0.25	2	0.0117	0.0000	0.0117	1
II	0.5	0.25	2	0.5313	0.5039	0.0273	1
	0.75	0.25	2	1.1055	1.0625	0.0430	1
	1	0.25	2	1.8750	1.8164	0.0586	1
	Step 3: increase the order of Taylor series and reestimating function values at						
	x	Step size	d	f^(x-step)	f(x-step)	|f^-f|	error >epsilon
	0.25	0.25	3	-0.0039	0.0000	0.0039	1
III	0.5	0.25	3	0.5000	0.5039	0.0039	1
	0.75	0.25	3	1.0586	1.0625	0.0039	1
	1	0.25	3	1.8125	1.8164	0.0039	1
	Step 4:Adding points to the intevrals with error greater than epsilon						
	x	Step size	d	f^(x-step)	f(x-step)	|f^-f|	error >epsilon
	0.125	0.125	3	-0.0002	0.0000	0.0002	0
	0.25	0.125	3	0.2500	0.2502	0.0002	0
	0.375	0.125	3	0.5037	0.5039	0.0002	0
IV	0.5	0.125	3	0.7695	0.7698	0.0002	0
	0.625	0.125	3	1.0623	1.0625	0.0002	0
	0.75	0.125	3	1.4023	1.4026	0.0002	0
	0.875	0.125	3	1.8162	1.8164	0.0002	0
	1	0.125	3	2.3359	2.3362	0.0002	0
	Step 5: Integration						
	x	Step size	d	error >epsilon	F(X)	F(X-step)	integral
	0.125	0.125	3	0	0.0079	-0.0039	0.0079
	0.25	0.125	3	0	0.0239	-0.0196	0.0239
	0.375	0.125	3	0	0.0403	-0.0356	0.0403
V	0.5	0.125	3	0	0.0579	-0.0525	0.0579
	0.625	0.125	3	0	0.0777	-0.0714	0.0777
	0.75	0.125	3	0	0.1013	-0.0937	0.1013
	0.875	0.125	3	0	0.1304	-0.1210	0.1304
	1	0.125	3	0	0.1673	-0.1555	0.1673
						Sum	**1.1599**

4.2 Performance Comparison

Here we study the performance of the proposed methods based on six different functions, including polynomial, rational, exponential, Trigonometric, hyperbolic and oscillating functions, and compare with five other integration methods, namely trapezoidal, Romberg, adaptive Simpson quadrature, adaptive Gauss-Kronrod quadrature and adaptive Lobatto quadrature from the literature.

For method 1 and 3, we start the Taylor series approximation with $d = 2$, and let it increase till $d_{max} = 10$. For method 2, we use $d = 10$ throughout the integration process. For all of the three proposed methods we use $m = 50$, and set maximum number of points as $m_{max} = 75$. In addition, we set the Taylor series approximation error limit as $\varepsilon = 10^{-5}$ for all of the three methods. For the trapezoidal method we set the number of points as $m = 100$ and for the Romberg method we use 7 rows. For all of the adaptive quadrature methods we use the Tolerance level of $\varepsilon = 10^{-6}$. For the adaptive Gauss-Kronrod quadrature methods in particular we also set the maximum interval count as 650.

Table 4, presents the specific functions, related interval of integration and the level of error achieved by each of the comparing methods. As illustrated in the table, even though the number of function evaluations in the proposed methods is limited to $m_{max} = 75$, in all cases they achieved an error of less than 10^{-5} (on average 10^{-6}) which is competitive with other methods. Notably, in several cases the number of function evaluations for the proposed methods has been equal or very close to initial 50 points, because it already satisfied the predefined Taylor series approximation error of $\varepsilon = 10^{-5}$. In other words, the predefined Taylor series approximation error of, i.e. $\varepsilon = 10^{-5}$, set the upper bound of the error performance of the proposed methods. Meanwhile, as investigated by the authors, the error performance of the proposed methods improves and converges to the performance of the best performing methods by increasing the maximum degree of Taylor series approximation and/or number of points. In summery, the adaptive nature of the proposed framework makes it an efficient and effective numerical integration tool for many practical applications which requires both accuracy and minimum function evaluations.

Table 4: The error performance of the comparing methods

Function	exp(x)	1/(x+1)	cosh(x)-cos(x)	x^5-x	Sin(1/x)	e^(-x^2/2)
range	(0 , 0.5) *Error*	(1,2) *Error*	(0 , 0.5) *Error*	(0 , 0.5) *Error*	(0.1, 1) *Error*	(0 , 0.5) *Error*
Method 1 *Max Derv. Order= 10* *No. of init points= 50* *Max add. points= 25*	< 10^-6	< 10^-6	< 10^-6	<10^-14	< 10^-6	< 10^-6
Method 2 *Taylor ser. order=10*	< 10^-6	< 10^-6	< 10^-6	<10^-8	< 10^-6	< 10^-6
Method 3 *Taylor ser. order=10*	< 10^-6	< 10^-6	< 10^-6	<10^-8	< 10^-5	<10^-6
Trapozeidal *m=100*	<10^-2	< 10^-6	< 10^-6	< 10^-6	< 10^-5	< 10^-6
Romberg *No. of rows = 7*	< 10^-6	< 10^-12	< 10^-15	< 10^-15	< 10^-9	< 10^-15
Adapt. Simpson quad. *Tolerance=1.0e-6*	< 10^-6	< 10^-9	< 10^-11	< 10^-15	< 10^-8	< 10^-10
Adapt. GK quad. *Rel. Tol. =1.0e-6* *Max Int. Count= 650*	< 10^-6	< 10^-15	< 10^-15	< 10^-15	< 10^-15	< 10^-15
Adapt. Lobatto quad. *Tol.=1.0e-6*	< 10^-6	< 10^-11	< 10^-15	< 10^-15	< 10^-8	< 10^-14

5 Conclusion

In this paper we proposed an adaptive numerical integration framework for application, where the number of possible function evaluations are limited but the high level of accuracy is needed. The proposed framework uses the basic idea of Taylor series for function approximation and integration, while adaptively increasing the degree of the Taylor series and refines the integration area as necessary to reduce the integration error to the desired level. To calculate the high order derivatives efficiently, the proposed methodology uses multicomplex algebras and their corresponding matrix representations through multicomplex Taylor series expansion (MCTSE). In addition, it uses the predetermined error of the MCTSE to control the accuracy of the results. The framework has a modular strategy and its components can be used in different combinations. In particular, we present three different methods derived from the proposed framework and demonstrate their competitive performance against other methods in the literature. Method 1 is computationally more expensive than Methods 2 and 3, but provides the best results. Method 2 is suitable for applications with higher order differentiability. Finally, Method 3 is the most computationally efficient algorithm among the three. The proposed framework can be effectively applied to scientific and engineering problems such as Finite Element method (FEM) to calculated integrals with limited number of function evaluations. For the future work we plan on extending the proposed idea to numerical optimization.

References

[1] P. J. Davis and P. Rabinowitz, Methods of Numerical Integration. Courier Corporation, (2007).
[2] F. B. Hildebrand, Introduction to Numerical Analysis. Courier Corporation, (1987).
[3] H. Hirayama, Fast Numerical Integration Method Using Taylor Series. In Integral Methods in Science and Engineering, Birkhauser Boston, 135-140, (2008).
[4] P. K. Kythe and M. R. Schaferkotter, Handbook of Computational Methods for Integration, Chapman and Hall CRC Press, New York (2004).
[5] G. Lantoine, R. P. Russell and T. Dargent, Using Multicomplex Variables for Automatic Computation of High-order Derivatives, ACM Transactions on Mathematical Software (TOMS) **38** (2012).
[6] G. Lantoine and R. P. Russell, A Methodology for Robust Optimization of Low-Thrust Trajectories in Multi-Body Environments, Dissertation in the School of Aerospace Engineering, Georgia Institute of Technology (2010).
[7] W. M. McKeeman, Algorithm 145: Adaptive Numerical Integration by Simpson's Rule, Communications of the ACM **5**(12), 604 (1962).
[8] H. R. Millwater and S. Shirinkam, Multicomplex Taylor Series Expansion for Computing High-order Derivatives, Int. J. Appl. Math. **27**(4),311–334 (2014). DOI: 10.12732/ijam.v27i4.2.
[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in FORTRAN: The Art of Scientific Computing. Cambridge: Univ. press. (1992).
[10] G. B. Price, An Introduction to Multicomplex Spaces and Functions, Marcel Dekker INC, New York (1991).
[11] J. M. Smith, Recent Developments in Numerical Integration. Journal of Dynamic Systems, Measurement, and Contro **96**(1), 61-70, (1974).
[12] V. Vittaldev, R. P. Russell, N. Arora and D. Gaylor, Second-order Kalman Filter Using Multicomplex Step Derivatives, American Astronomial Society **204** (2012).
[13] C. W. Ueberhuber, Numerical Computation 1: Methods, Software, and Analysis, Springer Science & Business Media **16**, (1997).

Sara Shirinkam is a faculty at the Department of Mathematics at University of Texas at San Antonio. She received her PhD degree in Mathematics from K.N. Toosi University of Technology. She spent the last two years of her PhD as a Visiting Scholar at University of Tennessee, Knoxville. She also did her Post Doc at University of Texas at San Antonio. Her research areas are in sensitivity analysis, numerical analysis and commutative rings.

Adel Alaeddini is an Assistant Professor of Mechanical Engineering at University of Texas at San Antonio. He obtained his Ph.D. in Industrial and Systems Engineering from Wayne State University. He also did a Post Doc at University of Michigan, Ann Arbor. His main research interests include data analytics, operations research and their applications in engineering.

Harry Milwater received his PhD degree in Engineering Mechanics from the University of Texas at Austin. His research areas are in uncertainty quantification methods as applied to structural engineering systems. His research efforts are funded by the U.S. Air Force Office of Scientific Research, the Office of Naval Research, the Air Force Research Laboratory, and the Federal Aviation Administration