Int. J. Nonlinear Anal. Appl. 12 (2021) No. 1, 830-837
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.4938

New optimization algorithm to improve numerical integration method

Inaam Rikan Hassan[a]

[a] *University of Information Technology and Communications, Baghdad, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

This paper introduces a new proposed algorithm of numerical integration evaluation regarded as optimization problem solution. The new method is characterized to have superiority features such as attractive, accurate and rapid. An improvement of polynomial regression has been done by selecting nearest neighbors points being searched around of the values of regression coefficients which calculated by using least squares method. Furthermore, Trapezoidal and Simpson methods were considered as traditional methods in numerical integration. In this regard, comparison has been done among all four methods used in simulation application via MATLAB program that have been performed to achieve the desired numerical results for the four methods. As conclusion, the proposed algorithm approved its superiority.

Keywords: Trapezoidal method, Simpson method, Optimization problem, Polynomial regression, Least squares method.

1. Introduction

The tremendous development made researchers compete in a great way in various fields, and all these fields were based on the mathematical equations used in order to implement these modern technologies, whether artificial intelligence or other technologies used [1]-[12]. In general, integral results can be got from mathematics books specializing in integration and differentiation; however, this cannot be always done, because some functions do not have the "antiderivative". Such functions can be expressed in other famous functions, such as polynomials, exponentials. Fortunately, these definite integrations can be approximated by using many methods, for instance: Trapezoidal or Simpson.

*Corresponding author
Email address: drinrh@uoitc.edu.iq (Inaam Rikan Hassan)

Received: September 2020 *Revised:* November 2020

Suppose f to be function defined on the segment [a, b], assuming that: $S = \{x_0, x_1, x_2, \ldots, x_n\}$ be systematic partition of $[a, b]$, i.e., equidistances on $[a, b]$. This partition will produce "n" of sub-segments on the total segment [a, b] whose length of each is fixed and equally which is called: "h", where is: $h = \frac{(b-a)}{n}$.
Suppose that we want to calculate the following integration result:

$$I = \int_a^b f(x)dx \tag{1.1}$$

Much attention has been given to the application of numerical integration due to its interesting features. Many literatures have been written in this topic [13].

A straightforward numerical integration procedure that can be used for both unequally spaced data and equally spaced data is based on fitting the data by a direct fit polynomial and integrating that polynomial. Thus,

$$I \cong \int_a^b P_n(x)dx = a_0 + a_1x + a_2x^2 \tag{1.2}$$

Polynomial regression interpolation regarded as a useful and reliable method in obtaining numerical integration in this case [14]. The need to evaluate numerical integration for nonlinear difficult functions in a modern fast method based on seeking the most accurate solution as a target point. This approach is done by new technique, which makes the achievement to the required estimate faster and slighter. In this paper, a new algorithm of numerical integration is described. This method aims to solve an optimization problem may be similar to genetic algorithm, therefore, we can call it as: semi-genetic algorithm. Genetic algorithm is based on the original data x_i space themselves, while the proposed algorithm is based on polynomial regression coefficient space instead.

2. Materials and Methods

I.) Traditional Methods

Many authors have introduced traditional methods such as Trapezoidal (Tp.) by [15].

$$I_{(Tp)} = \frac{h}{2}\left[f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)\right] \tag{2.1}$$

Furthermore, Simpson method with 3n (Sp.) by [16], where is n is even.

$$I_{(Sp)} = \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{n-1}) + 4f(x_{n-1}) + f(x_n)\right] \tag{2.2}$$

II. Polynomial Regression (PR)

New algorithm has been suggested by Muangchan [17], to describe employing polynomial regression method in terms of quadratic polynomial which is characterized by three main coefficients called (α, β and γ). This method can be explained as the following algorithm below.

Step 1: Putting $x_i = x_0 +$ ih ; $(i = 1, 2, \ldots, n)$ where is: $x_0 = a, x_n = b$ and $h = \frac{(b-a)}{n}$

Step 2: Putting $x_1 = x_{i-1}$, $x_2 = x_i$ and $x_3 = \frac{x_i + x_{i-1}}{2}, (\forall i, \quad i = 1, 2, \ldots, n)$

Using then we write the function as a quadratic polynomial:

$$f^*(x_j) = \alpha + \beta x_j + \gamma x_j^2 \quad , \quad j = 1, 2, 3 \tag{2.3}$$

Step3: Calculate $\Delta_i = \left[x_i - x_{i-1} \frac{x_i^2 - x_{i-1}^2}{2} \quad \frac{x_i^3 - x_{i-1}^3}{3} \right]'$
which is (1×3) vector , i = 1, 2, 3, ..., n
to construct the following square matrix

$$\Delta = \begin{bmatrix} \Delta_1 & & & \\ & \Delta_2 & & \\ & & \ddots & \\ & & & \Delta_n \end{bmatrix} \quad ; \quad (\text{n} \times 3\text{n}) \text{ matrix}$$

Step 4: Find the polynomial regression coefficients (α, β and γ) by solving the normal equations system [14], the following solution can be obtained:
$\underline{B} = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n & \beta_1 & \beta_2 & \ldots & \beta_n & \gamma_1 & \gamma_2 & \ldots & \gamma_n \end{bmatrix}'$
where is $\underline{\mathbf{B}}$ of order $(3\text{n} \times 1)$ vector

$$\underline{B} = (Z'Z)^{-1} Z' \underline{f} \tag{2.4}$$

Z: $(3n \times 3n)$ Block-diagonal (Z_i) matrix

$$Z = \begin{bmatrix} Z_1 & & & \\ & Z_2 & & \\ & & \ddots & \\ & & & Z_n \end{bmatrix}$$

$$Z_i = \begin{bmatrix} 1 & x_{i-1} & x_{i-1}^2 \\ 1 & x_i & x_i^2 \\ 1 & \frac{x_{i-1}+x_i}{2} & \left(\frac{x_{i-1}+x_i}{2}\right)^2 \end{bmatrix} \quad ; \quad (3 \times 3) \text{ matrix }, \quad \text{i} = 2, 3, \ldots, \text{n}+1$$

f: $(3n \times 1)$ vector

$$\underline{f} = \begin{bmatrix} \underline{f}(j)1 \\ \frac{f}{-(j)2} \\ \vdots \\ \underline{f}_{(j)n} \end{bmatrix} \quad ; \quad j = 2, 3, \ldots, \text{n}+1$$

$$\underline{f}_{(j)} = \begin{bmatrix} f\left(x_{j-1}\right) \\ f\left(x_j\right) \\ f\left(\frac{x_{ji-1}+x_j}{2}\right) \end{bmatrix} \quad ; \quad (3 \times 1) \text{ vector} ; \quad j = 2, 3, \ldots, n+1$$

Step 5: Definite integration result can be approximated by using polynomial regression:

$$I_{(PR)} = \underline{1} \cdot \Delta \cdot \underline{B} \tag{2.5}$$

where is: $\underline{1} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$, $(1 \times n)$ vector

III.) **Proposed Optimization Method (Pm.)**

The main goal is to improve the accuracy of approximated integration. New iterative algorithm can be proposed by the following steps using MATLAB program to achieve the accurate polynomial regression coefficients (α, β and γ) and get better solution.

Step 1: Initialize with space search, which is represented by polynomial regression coefficients vector ($\underline{B}$) from PR as shown previously in equation (6) for each subsegment with length "h".

Step 2: Picking some points of coefficient vector ($\underline{B}$) for each sub-segment described in previous (PR) method. Consequently, we will get (3n) coefficients according to the equation (5). Starting with these set of coefficients, we construct region search by finding new values around these coefficients as neighbors in distance boundaries about half-step (1/2) to the right and left of old initial values, i.e., ($\underline{B}' = \underline{B} \pm 0.5$).
Step 3: Working on the previous segments described above, we can generate **r** of values of coefficients vector ($\underline{B}'$) , for (k $= 1, 2, 3, \ldots,$ r) and r depends on MATLAB program loop that lies along with the distance lengths ($\underline{B} - \underline{B}$) to generate a very large combination of these values to be updated so that each combination gives a certain error calculated by mean absolute error: MAE is associated with the corresponding numerical integration calculating according to the equation (7) between the exact value f(x) and the approximate f′(x)
Step 4: The best (optimal) combination that corresponding to the lowest MAE is selected according to this basis, optimal vector $\underline{B}^*$ will be obtained that are not less than PR accuracy but even have often overtake them, i.e.,

$$I^*_{(Pm)} = \underline{1} \cdot \Delta \cdot \underline{B}^* \tag{2.6}$$

Step 5: Otherwise, the algorithm will stop or return to step (2) above.

Error Criterion Expression

It is can be said that we have a comparison of exact values f $(\mathrm{x_i})$ along with f^* $(\mathrm{x_i})$ as numerical result. Therefore, the error criterion that used widely to do this is:
$E = |f(x_i) - f^*(x_i)|$
Which has to be minimized. Where is Mean Absolute Error: MAE can expressed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |I_i^* - I_i|$$

where is:
I: is the exact value of integration.
I* : is the result obtained by proposed algorithm.

3. Numerical Results

Variety of five functions has been taken into account to compute their numerical integration to the corresponding segments [18],[3] for n $= 4, 6, 8,$ 12. Numerical results were obtained by MATLAB program to test the proposed method performance compared with other methods. The functions are as below.

$f_1(x) = \frac{1}{x}$,	$[2, 4]$	$f_4(x) = \frac{4-\sin^2(6+2x)}{e^{x+1}}$,	$[-0.75, 3]$
$f_2(x) = (2 + \cos(x))^2$,	$[0.75, 2]$	$f_5(x) = \frac{1}{x^3+1}$,	$[1, 2]$
$f_3(x) = \frac{e^{-x^2}}{x} quad,$	$[0.1, 1.1]$		

Table 1: Numerical results when n = 4

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.697024	0.693254	0.693155	0.693155
$f_2(x)$	6.104008	6.096719	6.097040	6.097063
$f_3(x)$	2.336885	2.074415	1.965115	1.941004
$f_4(x)$	2.794311	2.728517	2.607984	2.600888
$f_5(x)$	0.257500	0.254312	0.254350	0.254353

Table 2: Error when n = 4

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.003877	0.000107	7.35E − 06	7.35E − 06
$f_2(x)$	0.006947	0.000341	2.05E − 05	2.44E − 06
$f_3(x)$	0.395880	0.133411	0.024111	2.66E − 15
$f_4(x)$	0.193423	0.127629	0.007096	7.27E − 13
$f_5(x)$	0.003147	4.09E − 05	3.06E − 06	6.17E − 07

Table 3: Numerical results when n = 6

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.694877	0.69317	0.693149	0.693149
$f_2(x)$	6.100128	6.096995	6.097056	6.097060
$f_3(x)$	2.137803	1.992287	1.948441	1.941004
$f_4(x)$	2.694725	2.646692	2.601720	2.600888
$f_5(x)$	0.255749	0.254344	0.254352	0.254353

Table 4: Error when n = 6

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.001730	2.26E − 05	1.48E − 06	1.48165E − 06
$f_2(x)$	0.003067	6.54E − 05	4.02E − 06	4.72172E − 07
$f_3(x)$	0.196799	0.051283	0.007437	1.12831E − 11
$f_4(x)$	0.093837	0.045804	0.000831	1.11121E − 16
$f_5(x)$	0.001396	9.32E − 06	6.18E − 07	1.01784E − 07

Table 5: Numerical results when n = 8

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.694122	0.693155	0.693148	0.693148
$f_2(x)$	6.098782	6.09704	6.097059	6.097060
$f_3(x)$	2.058057	1.965115	1.94399	1.941004
$f_4(x)$	2.654566	2.607984	2.60111	2.600888
$f_5(x)$	0.255137	0.25435	0.254353	0.254353

Table 6: Error when n = 8

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.000975	7.35E − 06	4.72E − 07	4.72E − 07
$f_2(x)$	0.001722	2.05E − 05	1.27E − 06	5.36E − 07
$f_3(x)$	0.117053	0.024111	0.002986	4.44E − 16
$f_4(x)$	0.053678	0.007096	0.000222	8.88E − 16
$f_5(x)$	0.000784	3.06E − 06	1.97E − 07	1.97E − 07

Table 7: Numerical results when n = 12

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.693581	0.693149	0.693147	0.693147
$f_2(x)$	6.097824	6.097056	6.097060	6.09706
$f_3(x)$	1.995782	1.948441	1.941751	1.941004
$f_4(x)$	2.624971	2.601720	2.600927	2.600888
$f_5(x)$	0.254701	0.254352	0.254353	0.254353

Table 8: Error when n = 12

Function	Tp	Sp	PR	Pm
$f_1(x)$	0.000434	1.48E − 06	9.38E − 08	9.38E − 08
$f_2(x)$	0.000764	4.02E − 06	2.52E − 07	2.51E − 07
$f_3(x)$	0.054778	0.007437	0.000747	1.21E − 16
$f_4(x)$	0.024083	0.000831	3.92E − 05	8.88E − 16
$f_5(x)$	0.000348	6.18E − 07	3.91E − 08	3.91E − 08

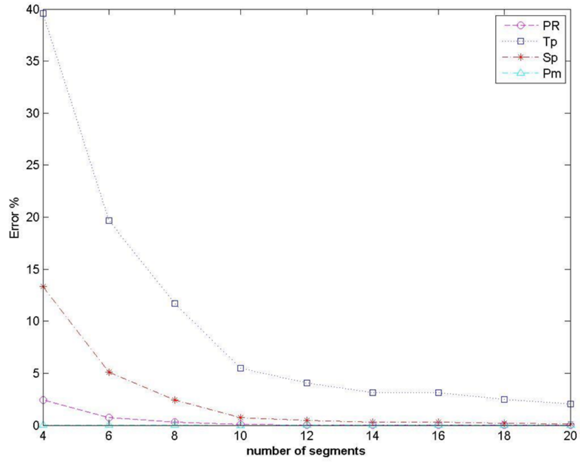

Figure 1: Accuracy results of all methods

4. Conclusion

From the previous numerical results, it has been explored that the proposed technique (Pm) has proven its superiority. It was obvious that the proposed method is the best compared to the other methods according to Error criterion and all sample sizes. The only exception has occurred when $(n = 8, 10)$ where (PR) method got the same result with (Pm) method as equivalence case for the 4^{th} function $f_4(x)$. Semi-genetic algorithm was the most accurate and very fast in operating time.

5. Future Work

The proposed method (Pm) can be applied for more complicated compound function in numerical integration.

References

[1] M. S. Mahdi, Y. M. Abid, A. H. Omran and A.-M. GH, *A Novel Aided diagnosis schema for covid 19 using convolution neural network*, 5th International Conference on Advanced Technology & Applied Sciences in conjunction with 6th Malaysia-Japan Joint International Conference, Malaysia, October 2020.

[2] Alaa Hamza Omran, Dalila M.S. et al. *A Novel intelligent detection schema of series arc fault in photovoltaic (PV) system based convolutional neural network*, Period. Engin. Natural Sci. 8(3) (2020) 1641–1653.

[3] A. H. Omran, D. M. Said, S. M. Hussin, S. Mirsaeidi and Y. M. Abid, *An intelligent classification method of series arc fault models using deep learning algorithm.* 2020 IEEE International Conference on Power and Energy (PECon), 7-8 Dec. 2020 Penang, Malaysia.

[4] A. Fadhel, A. H. Omran and Y.r M. Abid, *Novel intelligent traffic light system using PSO and ANN*, J. Adv. Res. Dyn. Cont. Syst. 11(4) 1528–1539.

[5] A. H. Omran, I. Raheem, and A. D. Hussein, *Minimizing the loses of PV panel generation by designing an intelligent controller based on FPGA*, Int. J. Engin. Tech. 7(4) (2018) 4846–4849.

[6] A. H. Omran and Y. M. Abid, *Design of smart chess board that can predict the next position based on FPGA*, Adv. Sci. Tech. Engin. Syst. J. 3(4) (2018) 193–200.

[7] A. H. Omran,Y. M. Abid, A. S. Ahmed, H. Kadhim and R. Jwad, *Maximizing the power of solar cells by using intelligent solar tracking system based on FPGA*. Advances in Science and Engineering Technology International Conferences (ASET), (2018) 1-5. https://doi.org/10.1109/ICASET.2018.8376786

[8] A. H. Omran,Y. M., Abid and H. Kadhim, *Design of artificial neural networks system for intelligent chessboard*, 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), (2017) 1-7. https://doi.org/10.1109/ICETAS.2017.8277882

[9] H. Badi, A. Hamza and S. Hasan, *New method for optimization of static hand gesture recognition*, Intelligent Systems Conference (IntelliSys), (2017) 542-544. https://doi.org/10.1109/IntelliSys.2017.8324347

[10] Y. M. Abid, A. H. Omran and R. Jwad *Supervised feed forward neural networks for Smart chessboard based on FPGA*, J. Engin. Appl. Sci. 13(11) (2018) 4093–4098.

[11] M. S. Mahdi, R. A. Azeez and N. F. Hassan, *A proposed lightweight image encryption using ChaCha with hyper-chaotic maps*, Period. Engin. Natural Sci. 8(4) (2020) 2138–2145.

[12] H. Samet, D. K. Asl, T. Ghanbari and A. H. Omran, *Optimal number and location of the required measurment units for fault detection of PV Arrays*, 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 1–5. https://doi.org/10.1109/EEEIC.2018.8493762

[13] J. Q. Howell, *A least-squares distance curve-fitting technique*, NASA Langley Research Center, L-7675 (1971) 1-22.

[14] C. Figueroa, R. Riera and G. Campoy, *Numerical interpolation with polynomials approximation and curve fitting, focus the MATLAB"*, IOSR J. Math. 12(1) (2016) 10–20.

[15] R. L. Burden and J. D. Faires, *Numerical analysis*, 9th edition, Brooks/Cole, Cengage Learning, USA, 2011.

[16] E. A. Costa, C. A. G Finger, P.R. Schneider and I. Müller, *Approximation of numerical integration applied to araucaria angustifolia Stem Taper models"*; Floresta Curit. 45(1) (2015) 31–40.

[17] S. Muangchan, P. Prasertsang, P. Mukwachi S. and Sompong, *A new algorithm of numerical integration by polynomial regression technique*, Appl. Math. Sci. 7(16) (2013) 76-772.

[18] R. Chiranjeevi and U. Dixit, *A note on Clamped Simpson's rule*, Glob. J. Pure Appl. Math. 13(9) (2017) 5275–5285.