International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

Solution of Numerical Integration using Fractal Interpolating Functions

Ali Hassan AL-Muhanna[1] and Adil AL-Rammahi[2]

University of kufa

Email: alialmuhanna45@gmail.com2, adilm.hasan@uokufa.edu.iq2

Abstract: *The main aim of this paper is to study a numerical integration via spline fractal interpolation functions. This thesis studies the solution of numerical integration of continuous function . In which a modified method is introduced for solving numerical integration via the composition of spline interpolation function and fractal interpolation function. To test the modified method, a comparison of this method with the famous update methods has been implemented. The error of modified method is the least than the other Newton – cotes methods represented as trapezoidal, Simpson and Bool. Gaussian quadrature and Bool are the nearest of least error with each of quadrature fractal spline and cubic fractal spline. Many examples were implemented via MATLAB*

keywords : Numerical Integration, Fractal Interpolation, Spline Interpolation ,Fractal Spline Method to Solution of Numerical Integration , Implementation and Comparison

1. Introduction

In this paper , we take the integration of the continuous function f(x) with x- independent variable , where f is dependent on the closed interval (a,b) ,

$$I = \int_a^b f(x)dx \qquad (1)$$

That we studied spline with fractal interpolation function quadratically one time , and cubically in another time , I have compared it with research AL_Sharifi [10] and AL_ Haq [11]

We have discussed numerical integration , and many those discussed are linear , quadratic and cubic spline and fractal interpolation function approach with illustrative examples it also has dealt with newton – cote approaches in solving numerical integration and Gaussian quadrature approach , and the question were presented in tables in that each example covered the integration approaches mentioned in the paper . and has solved numerical integration by using fractal interpolation function with the spline approaches represented in linear , quadratic , and cubic spline , and all the programmers are carried out through MATLAB.

2. Basic Concepts of Numerical Integration and Interpolation

To test the efficiency of our method , comparison with each of the method of Trapezoidal Rule , Simpson's 1/3-Rule , Simpson's 3/8-Rule , Boole's Rule and Gaussian Quadrature , is implemented all the miscellaneous examples have the least error via our modified method

2.1 Numerical Integration

We will deal here , with the rules and the main famous methods in the numerical integration [9]
Newton-Cotes Closed Formula

Trapezoid Rule

$$\int_a^b f(x)dx \approx \sum_{i=0}^{1} c_i f(x_i) = c_0 f(x_0) + c_1 f(x_1) \qquad (2)$$

$$= \frac{h}{2}[f(x_0) + f(x_1)]$$

Simpson's 1/3-Rule

$$\int_a^b f(x)dx \approx \sum_{i=0}^{2} c_i f(x_i) = c_0 f(x_0) + c_1 f(x_1) + c_2 f(x_2)$$
$$= \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] \quad (3)$$

Simpson's 3/8-Rule

$$\int_a^b f(x)dx \approx \sum_{i=0}^{3} c_i f(x_i) = c_0 f(x_0) + c_1 f(x_1) + c_2 f(x_2) + c_3 f(x_3)$$
$$= \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] \quad (4)$$

Boole's rule[9]

$$I = \frac{2h}{45}\{7f(a) - 32f(x_1) + 12f(x_2) - 32f(x_3) + 7f(b)\} \quad (5)$$

Gaussian Quadrature

$$I = \int_{-1}^{1} f(x)dx = f(\frac{-1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}}) \quad (6)$$

2.2 Spline Interpolation

We will deal here , the modern method in interpolation it is spline interpolation

2.2.1 Quadratic Interpolation

Given $(x_0,y_0)(x_1,y_1)\ldots(x_{n-1},y_{n-1})(x_n,y_n)$ fit Quadratic spline through the data the splines are given [2]

$$f(x) = a_1x^2 + b_1x + c_1 \qquad x_0 \geq x \leq x_1 \quad (7)$$

$$f(x) = a_2x^2 + b_2x + c_2 \qquad x_1 \geq x \leq x_2 \quad (8)$$

⋮

$$f(x) = a_nx^2 + b_nx + c_n \qquad x_{n-1} \geq x \leq x_n \quad (9)$$

Each Quadratic spline goes through two consecutive data point

$$f(x_0) = a_1 x_0^2 + b_1 x_0 + c_1 \quad (10)$$

$$f(x_2) = a_2 x_1^2 + b_2 x_1 + c_2 \quad (11)$$

$\vdots$

$$f(x_{i-1}) = a_i x_{i-1}^2 + b_i x_{i-1} + c_i \quad (12)$$

$$f(x_i) = a_i x_i^2 + b_i x_i + c_i \quad (13)$$

$\vdots$

$$f(x_{n-1}) = a_n x_{n-1}^2 + b_n x_{n-1} + c_n \quad (14)$$

$$f(x_n) = a_n x_n^2 + b_n x_n + c_n$$

The condition gives 2n equations (15)

The first derivatives of two Quadratic spline are continuous at the interior points

The first derivatives spline

$$a_1 x^2 + b_1 x + c_1 \quad \text{is} \quad 2a_1 x + b_1 \quad (16)$$

The second derivatives spline

$$a_2 x^2 + b_2 x + c_2 \quad \text{is} \quad 2a_2 x + b_2 \quad (17)$$

And the two are equal at x_1=x giving

$$2a_1 x_1 + b_1 = 2a_2 x_1 + b_2 \quad (18)$$

$$2a_1 x_1 + b_1 - 2a_2 x_1 + b_2 = 0 \quad (19)$$

Similarly at the other interior points

$$2a_2 x_2 + b_2 - 2a_3 x_2 + b_3 = 0 \quad (20)$$

$\vdots$

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

$$2a_i x_i + b_i - 2a_{i+1} x_i + b_{i+1} = 0 \qquad (21)$$

⋮

$$2a_{n-1} x_{n-1} + b_{n-1} - 2a_n x_{n-1} + b_n = 0 \qquad (22)$$

We have n-1such equation the total number of equation is 2n + n-1 =3n-1

We can assume that the first spline is liner that is $a_1=0$ this give us 3n equation and 3n unknown once we find the 3n constants we can find the function at any value of x using the splines[2]

2.2.2 Cubic spline

Given a function f defined on $[X_0 , X_n]$ and a set of node a= $X_0 < X_1 < \ldots < X_n$ =b , a cubic spline interpolated S for f is a function that satisfies the following conditions [3] :

1) s(x) is a cubic polynomial , denoted $s_i(x)$, on the subinterval $[X_i , X_{i+1}]$ for each i=0,1,2,…,n-1

2) $s(X_i)=f(X_i)$ for each i=0,1,…,n

3) $s_{i+1}(X_{i+1}) = s_i(X_{i+1})$ for each i=0,1,…,n-2

4) $S'_{i+1}(x_{i+1})= S'_i(x_{i+1})$ for each i=0,1,…,n-2

5) $S''_{i+1}(x_{i+1})= S''_i(x_{i+1})$ for each i=0,1,…,n-2

Natural Linear System [4]

In matrix form the system of n + 1 equations has the form
A**c** = **y** where

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & \ldots & 0 \\ 0 & h_1 & 2(h_0+h_1) & h_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

A is a tridiagonal matrix.

The vector **y** on the right-hand side is formed as

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

$$y = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) - \frac{3}{h_{n-3}}(a_{n-2} - a_{n-3}) \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix}$$

A is a tridiagonal matrix.

$$A\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) - \frac{3}{h_{n-3}}(a_{n-2} - a_{n-3}) \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix}$$

We solve this linear system of equations using a common algorithm for handling tridiagonal systems[4].

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(2c_i + c_{i+1}), i = 0,1, \dots, n-1 \quad (23)$$

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i), i = 0,1, \dots, n-1 \quad (24)$$

2.3 Fractal Interpolation

The set $G=\{(x,f(x):x\epsilon[x_0,x_N]\}$ is a graph of function $f:[x_0,x_N] \to [a,b]$ if there is only y-value which corresponds to each x-value , where $y\epsilon[a,b]$ For studing the usage of iterated function system in interpolation the mish points , first one can present the following example [5][6]

Example 1 [5][6] : the parabola defined by $f(x)=2x-x^2$ on the interval [0,2] is an interpolation function for the set of data {(0,0),(1,1),(2,0)} let G denoted the graph of f(x) that is $G=\{(x,2x-x^2):x\epsilon[0,2]\}$ from other hand one can take the hyperbolic IFS$\{R^2$; w_1 , $w_2\}$, where

$$w_1\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{4} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

$$w_2\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & 0\\ -\frac{1}{2} & \frac{1}{4}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}1\\1\end{pmatrix}$$

Let $\bar{G}$ denoted the attractor of the IFS . then G= $\bar{G}$

To prove this claim , one simply notes

$$w_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{2}x\\ \frac{1}{2}x+\frac{1}{4}y\end{pmatrix}=\begin{pmatrix}\frac{1}{2}x\\ \frac{1}{2}x+\frac{1}{4}(2x-x^2)\end{pmatrix}=\begin{pmatrix}\frac{1}{2}x\\ 2\left(\frac{1}{2}x\right)-\left(\frac{1}{2}x\right)^2\end{pmatrix}=\begin{pmatrix}\frac{1}{2}x\\ f(\frac{1}{2}x)\end{pmatrix}$$

In the same manner , one can show that $w_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}1+\frac{1}{2}x\\ f(1+\frac{1}{2}x)\end{pmatrix}$

Linear fractal interpolation function

$$w_n\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}a_n & 0\\ c_n & 0\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}e_n\\f_n\end{pmatrix}$$

With condition

$$w_n\begin{pmatrix}x_0\\y_0\end{pmatrix}=\begin{pmatrix}x_{n-1}\\y_{n-1}\end{pmatrix}\quad \text{and}\quad w_n\begin{pmatrix}x_N\\y_N\end{pmatrix}=\begin{pmatrix}x_n\\y_n\end{pmatrix}$$

So ,

$a_nx_0+e_n=x_{n-1}$ (25)

$a_nx_N+e_n=x_n$ (26)

$c_nx_0+d_n\,y_0+f_n=y_{n-1}$ (27)

$c_nx_N+d_n\,y_N+f_n=y_n$ (28)

therefore there are five unknown variables , four equations , and so one free (independent) will pick and choose d_n . looking at that acts vertically – stretches or compresses vertical line segments , acts like a shear one can study $0\le d_n<1$ and show that one has contraction maps , condition yields that[5][6][7] :

$a_n=\frac{(x_n-x_{n-1})}{(x_N-x_0)}$ (29)

$e_n=\frac{(x_Nx_{n-1}-x_0x_n)}{(x_N-x_0)}$ (30)

$$c_n = \frac{y_n - y_{n-1}}{(x_N - x_0)} - d_n \frac{y_N - y_0}{(x_N - x_0)} \quad (31)$$

$$f_n = \frac{(x_N y_{n-1} - x_0 y_n)}{(x_N - x_0)} - d_n \frac{x_N y_0 - x_0 y_N}{(x_N - x_0)} \quad (32)$$

$$L_n(x)=a_n x+e_n \quad (33)$$

$$F_n(x,y)=c_n x+d_n y \quad (34)$$

$$f(t)=q_n \circ l_n^{-1} \text{ for all } t \in I_n \quad (35)$$

$$x= 1/a_n-e_n \quad (36)$$

3 Fractal spline method to solution of numerical integration

In this section, we will engage the spline function with the aforementioned equation (34) to find approximate integral values that are better than the rest of the methods used in numerical analysis by assuming the function once squared and cubic times and put up examples and compare them with other methods as well as with other research using the MATLAB program

3.1 proposed numerical integration method using fractal interpolating and spline approximation

The following Procedure is to clear our proposed method for finding the definite numerical integration using fractal interpolating function with spline approximation

1) the equation will be employed in this section which is fractal interpolation function

   $F_n(x,y)=c_n x+d_n y$

2) This equation consists of displacing c_n and it can be obtained from the equation

$$c_n = \frac{y_n - y_{n-1}}{(x_N - x_0)} - d_n \frac{y_N - y_0}{(x_N - x_0)}$$

   multiplied by the one which represents the compensation value in the numerical points and can be obtained in the equation
   $x= 1/a_n-e_n$

3) The value of d_n can be chosen between 0 and 1

4) Interlocking the spline function in the equation of fractal interpolation function, so it could be quadratic one time and cubic in the second time and can be symbolized by y

5) Choosing the value n

6) We integration this equation to obtain the improved result which contains less errors or no errors

7) This equation is used through a programmer formulated in the MATLAB

3.2 Implementation and Comparison :

I will perform calculations of numerical integrals for several examples and by comparing our developed method with known methods which are trapezoidal, Simpson, Bool, Gaussian quadratic, Quadratic spline and Cubic spline as shown in the following tables.

Table (1) : Numerical Integration Implementation

$\int_0^4 \frac{dx}{\sqrt{1+x}} = 2.4721$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	2.8944	0.4223
Simpson	2.5044	0.0323
Simpson 3/8	2.4889	0.0168
Bool	2.4749	0.0028
Gaussian quadratic	2.4535	0.0186
Quadratic spline	2.5044	0.0323
Cubic spline	2.484	0.0119
Fractal of quadratic spline	2.4649	0.0072
Fractal of cubic spline	2.4721	0

Table (2) : Numerical Integration Implementation

$\int_1^2 \frac{dx}{x} = 0.6931$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	0.7500	0.0569
Simpson	0.6944	0.0013
Simpson 3/8	0.6938	0.0007
Bool	0.6932	0.0001
Gaussian quadratic	0.6923	0.0008
Quadratic spline	0.6944	0.0013
Cubic spline	0.6929	0.0002
Fractal of quadratic spline	0.6932	0.0001
Fractal of cubic spline	0.6930	0.0001

Table (3) : Numerical Integration Implementation

$\int_1^2 \frac{dx}{\ln x} = 1.9224$		
Method	**The Resulting**	**Absolute Error**

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

Trapezoidal	2.1640	0.2416
Simpson	1.9350	0.0126
Simpson 3/8	1.9286	0.0062
Bool	1.9231	0.0007
Gaussian quadratic	1.9147	0.0077
Quadratic spline	1.9350	0.0126
Cubic spline	1.9350	0.0126
Fractal of quadratic spline	1.9224	0
Fractal of cubic spline	1.9224	0

Table (4) : Numerical Integration Implementation

$\int_{0.1}^{0.3}\frac{\cos xdx}{x}=1.0787$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	1.3134	0.2347
Simpson	1.0912	0.0125
Simpson 3/8	1.0848	0.0061
Bool	1.0793	0.0006
Gaussian quadratic	1.0710	0.0077
Quadratic spline	1.0912	0.0125
Cubic spline	1.0912	0.0125
Fractal of quadratic spline	1.0689	0.0098
Fractal of cubic spline	1.0788	0.0001

Table (5): Numerical Integration Implementation

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

$\int_1^2 \ln(x)dx = 0.3862$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	0.3466	0.0396
Simpson	0.3858	0.0004
Simpson 3/8	0.3861	0.0001
Bool	0.3863	0.0001
Gaussian quadratic	0.3866	0.0004
Quadratic spline	0.3858	0.0004
Cubic spline	0.3858	0.0004
Fractal of quadratic spline	0.3715	0.0147
Fractal of cubic spline	0.3862	0

I will perform calculations of numerical integrals for several examples and by comparing our developed method with a method Abdul Sharifi [10] and Abdel-Haq[11] as shown in the following tables

Table (6): Numerical Integration Implementation

$\int_0^1 \frac{dx}{x+1} = 0.6931$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	0.7500	0.0569
Simpson	0.6944	0.0013
Simpson 3/8	0.6938	0.0007
Bool	0.6932	0.0001
Gaussian quadratic	0.6923	0.0008
Quadratic spline	0.6944	0.0013
Cubic spline	0.6929	0.0002
Rectangular	0.7187	0.0256
Fractal of quadratic spline	0.6931	0
Fractal of cubic spline	0.6924	0.0007

Table (7): Numerical Integration Implementation

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 4, Issue 5, May – 2020, Pages: 32-43

$\int_{-1}^{1} \frac{dx}{\sqrt{4-x^2}} = 1.0472$		
Method	**The Resulting**	**Absolute Error**
Trapezoidal	1.1547	0.1075
Simpson	1.0516	0.0044
Simpson 3/8	1.0493	0.0021
Bool	1.0474	0.0002
Gaussian quadratic	1.0445	0.0027
Quadratic spline	1.0516	0.0044
Cubic spline	1.0516	0.0044
Romberg	1.0644	0.0172
Fractal of quadratic spline	1.0472	0
Fractal of cubic spline	1.0475	0.0003

4. Conclusion

A modified method for solving numerical integration using spline and fractal interpolation function is introduced . So the following Tables were discussed with other integrated methods known at the Multiple function within the interval

The function mentioned in Table (10) is descending that the amount of error of fractal quadratic spline equals 0.0072 at the value contractively factor is 0.7 . and the amount of error of fractal cubic spline equals 0 at the value contractively factor is 0.34 and it is considered the best, and the Bool's method is also the nearest

The function mentioned in Table (11) is descending that the amount of error of fractal quadratic spline equals 0.0001 at the value contractively factor is 0.4658 . and the amount of error of fractal cubic spline equals 0.0001 at the value contractively factor is 0.7229 and it is considered in both cases the best, and the Bool's method is also the nearest

The function mentioned in Table (12) is descending that the amount of error of fractal quadratic spline equals 0 at the value contractively factor is 0.765 . and the amount of error of fractal cubic spline equals 0 at the value contractively factor is 0.6435 and it is considered in both cases the best, and the Bool's method is also the nearest

The function mentioned in Table (13) is descending that the amount of error of fractal quadratic spline equals 0.0098 at the value contractively factor is 0.4 . and the amount of error of fractal cubic spline equals 0.0001 at the value contractively factor is 0.6421 and it is considered the best, and the Bool's method is also the nearest

The function mentioned in Table (14) is ascending that the amount of error of fractal quadratic spline equals 0.0147 at the value contractively factor is 0.1 . and the amount of error of fractal cubic spline equals 0 at the value contractively factor is 0.463 and it is considered the best, and the Bool's and Simpson 3/8 method is also the nearest

The function mentioned in Table (15) is descending, is shown by comparing it with the methodology of the researcher by adopting the rectangle method by (shariffi) that the amount of error equals 0.0256 and by using the fractal function. It has been found out that the amount of error of fractal quadratic spline equals 0 at the value contractively factor is 0.7812 and considered the best, and the amount of error of fractal cubic spline equals 0.0007 at the value contractively factor is 0.51 and the Bool's method is also the nearest

The function mentioned in Table (16) by (Abdel-Haq) has used Romberg's integration acceleration that the amount of error equals 0.0172 and by using the fractal function, it has been found out that the amount of error of fractal quadratic spline equals 0 at the value contractively factor is 0.9952 which is considered the best, and the amount of error of fractal cubic spline equals 0.0003 at the value contractively factor is 0.931 and the Bool's method is also the nearest.

References

[1] M. K. Jain, S.R.K. Iyenger and R.K. Jain , ″ Numerical Methods for Scientific and Engineering Computations″ , New age International Publication (P)Ltd ,(2007) .

[2] A. k. Ghosh , " Introduction to Measurement and Instrumentation " , new delhi , International/Eastern Economy Edition, university of kalyani , (2012) .

[3] Richard L. Burden and J. Douglas Faires , " Numerical Analysis ", brooks / cole publishing company , USA, (1984).

[4] J. Robert Buchanan , " Cubic Spline Interpolation " , MATH 375, Numerical Analysis , Department of Mathematics , Spring , (2019) .

[5] barnsley , M. F. ," fractal function and interpolation constructive " approximation ,2,303-329, (1986) .

[6] barnsley , M. F. " fractals everywhere ", academic press ,(1988) .

[7] M.A.Navascues and M.V.Sebastian , (2006) . " Smooth fractal interpolation " , hindawi publishing corporation

[8] Curtis F.Gerald / Patrick O. Wheatley , " Applied Numerical Analysis " , Addison –wesley publishing company , California polytechnic state university , ISBN 0-201-11579-4 , (1984).

[9] P. V.Ubale , "Numerical Solution of Boole's Rule in Numerical Integration By Using General Quadrature Formula " , Sci Press Ltd, Switzerland , G.S.College of Science, Arts, Comm , ISSN: 2277-8020, Vol. 2, pp 1-4 ,(2012) .

[10] F. H. Abdul Sharifi , I. A. Al-Jazaery , s. h. malak , " Devising a Base for Numerical Integration", Babylon University Journal , Pure and applied , (2016).

[11] H. A. Alou , M. H. Abdel-Haq , " Integration using numerical methods " , Libya , (2017).