AN ALGORITHM APPROXIMATES THE HIGHER ORDER DERIVATIVE WITH HIGH ACCURACY

Vu Vinh Quang, Nguyen Dinh Dung
TNU - University of Information and Communication Technology

ARTICLE INFO

Received: 02/01/2024
Revised: 28/3/2024
Published: 29/3/2024

KEYWORDS

Derivative
Grid space
Mesh function
Set of neighboring points
Order of accuracy

ABSTRACT

When studying and solving practical problems in continuous environments, through modeling methods, the vast majority of problems lead to models described by partial differential equations, i.e. models that contain differential operators. For a very small class of problems corresponding to simple models and boundary conditions, we can obtain a direct solution of the problem through analytical methods, while the vast majority of complex problems can be obtained through analytical methods. Methods of discretizing differential operators to convert to systems of difference equations. Then the approximate solution will be obtained through solving the system of difference equations based on the tools of an electronic computer. With the need to obtain solutions with high accuracy, the issue of researching methods to discretize differential operators with high precision is a research area of special interest to mathematicians. In this paper, we propose an algorithm to discretize the n-th order derivative with high-order accuracy. Theoretical results and experimental calculations have confirmed the accuracy of the algorithm.

THUẬT TOÁN TÍNH GẦN ĐÚNG ĐẠO HÀM CÁC CẤP VỚI ĐỘ CHÍNH XÁC BẬC CAO

Vũ Vinh Quang, Nguyễn Đình Dũng[*]
Trường Đại học Công nghệ thông tin và Truyền thông - ĐH Thái Nguyên

THÔNG TIN BÀI BÁO

Ngày nhận bài: 02/01/2024
Ngày hoàn thiện: 28/3/2024
Ngày đăng: 29/3/2024

TỪ KHÓA

Đạo hàm
Không gian lưới
Hàm lưới
Tập điểm lân cận
Cấp chính xác

TÓM TẮT

Khi nghiên cứu giải quyết các bài toán thực tế trong các môi trường liên tục, thông qua phương pháp mô hình hóa thì đại đa số các bài toán đều đưa đến mô hình được mô tả bởi các phương trình vi phân đạo hàm riêng tức là các mô hình có chứa toán tử vi phân. Một lớp rất nhỏ các bài toán ứng với mô hình và điều kiện biên đơn giản, ta có thể thu được lời giải trực tiếp của bài toán thông qua các phương pháp giải tích, còn đại đa số các bài toán phức tạp đều thông qua các phương pháp rời rạc hóa các toán tử vi phân để chuyển về các hệ phương trình sai phân. Khi đó nghiệm xấp xỉ sẽ thu được thông qua việc giải các hệ phương trình sai phân dựa trên công cụ của máy tính điện tử. Với yêu cầu cần thu được lời giải với độ chính xác cao thì vấn đề nghiên cứu các phương pháp rời rạc hóa các toán tử vi phân với độ chính xác cao là một hướng nghiên cứu được các nhà toán học đặc biệt quan tâm. Trong bài báo này, chúng tôi đề xuất một thuật toán rời rạc hóa đạo hàm các cấp với độ chính xác bậc cao. Các kết quả lý thuyết và tính toán thực nghiệm đã khẳng định độ chính xác của thuật toán.

DOI: https://doi.org/10.34238/tnu-jst.9516

[*] Corresponding author. *Email: nddung@ictu.edu.vn*

1. Introduction

The vast majority of problems in continuous environments, through discretization of differential operators, result in models of systems of difference equations based on a suitable grid space. Then the use of formulas to discretize differential operations with what order of accuracy will determine the order of accuracy of the discrete solution obtained through algebraic methods or iterative methods of solving the system of differential equations. Therefore, studying methods to approximate the derivative value at a point of a function is an important research direction that mathematicians are especially interested in. In the literature on numerical methods [1], formulas for approximating the first and second derivatives of the function $u(x)$ with accuracy order $O(h^2)$ have been given, which uses the value of the function at 3 neighboring points of point x and h is denoted as the grid step on the grid space. Based on the interpolation polynomial approximation method, in the document [2], the m-order derivative approximation formulas with accuracy order $O(h^{5-m})$ are given, which uses value of the function at 5 neighboring points of point x. Recently, in the document [3], a set of formulas for approximating derivatives of all order with accuracy order $O(h^{n-m})$ has been introduced, which uses the value of the function at n+1 neighboring points of point x. Based on the published formulas, numerical solutions for differential problems with nonlinear differential equations of all order have been improved. In [4] – [8], algorithms for numerically solving 3rd and 4th order nonlinear differential equations with accuracy order $O(h^6)$ were published.

However, the published results of the above derivative approximation formulas are all results obtained by the direct method, results obtained by direct calculation of analytical expressions, not general. Thus, to improve the accuracy of derivative approximation, it is necessary to build a general algorithm to provide formulas for approximating derivatives of any degree with arbitrary order of accuracy based on the support of the computer.

The main content of the article will present theoretical research results based on interpolated polynomials to provide a derivative approximation method, thereby proposing a general algorithm for approximating mesh derivative vectors with orders of magnitude. Arbitrarily accurate based on the mesh function at $n+1$ neighboring points, proceed to build general derivative approximation formulas based on the coefficient matrix determined from the algorithm.

The structure of the article consists of four parts: In Section 1, we introduce some published results on the method of approximating higher order derivatives. Section 2, we present the theoretical basis of interpolated polynomials, based on which we propose an algorithm to approximate higher order derivatives in the case of regular and irregular grids. Section 3, we present some experimental calculation results to evaluate the accuracy of the proposed algorithm. Finally, there are conclusions and references.

2. Proposed method

Consider the function $u(x), x \in [a,b]$. Use a grid to divide segment $[a,b]$ into $N+1$ points by dividing points $a = x_0 < x_1 <\ldots< x_N = b$. Let $X = (x_0, x_1, \ldots, x_N)$ is the grid space, $U = (u(x_0), u(x_1)\ldots, u(x_N))$ is the grid function defined on the grid space, $D^{(m)}u(x_i)$ is the approximate value of the derivative $u^{(m)}(x_i)$ at point $x_i, (i = 0,1,2, \ldots, N)$; $D^{(m)}U = \left(D^{(m)}u(x_0), D^{(m)}u(x_1)\ldots, D^{(m)}u(x_N)\right)$ is the m-order derivative vector approximated on the grid space.

Let $Z = (z_0, z_1, \ldots z_n), n < N$ is the set of neighboring points of point $x_i, (i = 0,1,2, \ldots, N)$. We need to build an algorithm to determine the grid derivative vectors $D^{(m)}U$ when we know the grid function U with high-order accuracy using $(n+1)$ neighboring points Z.

2.1. Lagrangian interpolation polynomial [1]

Consider points x in the neighboring grid $Z = (z_0, z_1, \dots z_n), n < N$. Then suppose $n+1$ interpolation points are given

z_0	z_1	z_2	...	z_{n-1}	z_n
$u(z_0)$	$u(z_1)$	$u(z_2)$	...	$u(z_{n-1})$	$u(z_n)$

Consider the function approximation method using Lagrang interpolation polynomials in the Z neighborhood.

The symbol for Lagrange multipliers is n-th degree polynomials that satisfy the property

$$L_i(x) = \begin{cases} 1, x = z_i, \\ 0, x \neq z_i. \end{cases}$$

Lagrange multipliers are determined by the formula

$$L_i(x) = \frac{(x - z_0)(x - z_1) \dots (x - z_{i-1})(x - z_{i+1}) \dots (x - z_n)}{(z_i - z_0)(z_i - z_1) \dots (z_i - z_{i-1})(z_i - z_{i+1}) \dots (z_i - z_n)}, i = 1,2, \dots, n-1$$

Then we have the formula for approximating the function $u(x)$ in neighborhood Z using an n-th degree Lagrangian interpolation polynomial

$$u(x) = \sum_{i=0}^{n} L_i(x) u(z_i) + R_{n+1}(x), \quad (1)$$

where $R_{n+1}(x)$ is the error of the interpolation method.

Theorem 1 [1]. *Let $u(x) \in C^{n+1}[z_0, z_n]$,. Then the error of the interpolation method is determined according to the formula $R_{n+1}(x) = \frac{u^{(n+1)}(\xi))}{(n+1)!} \omega_{n+1}(x)$, where $z_0 < \xi < z_n$ is a point dependent on x, $\omega_{n+1}(x) = (x - z_0)(x - z_1) \dots (x - z_{n-1})(x - z_n)$.*

From the formula for approximating the function using the Lagrange interpolation polynomial $u(x)$, we obtain the formula for calculating the m order derivative of the function $u(x)$ in the neighborhood Z.

$$u^{(m)}(x) = \sum_{i=0}^{n} L_i^{(m)}(x) u(z_i) + R_{n+1}^{(m)}(x), \quad (2)$$

So we get the formula to approximate the m-order derivative

$$D^{(m)}(x) \approx \sum_{i=0}^{n} L_i^{(m)}(x) u(z_k) \quad (3)$$

Theorem 2 [2]. The error of the first derivative approximation is determined by the formula

$$R_{n+1}^{(1)}(x) = \frac{1}{(n+1)!} \prod_{k=0, k\neq i}^{n} (z_i - z_k) \sum_{j=0, j\neq i}^{n} \frac{u^{(n+1)}(\xi)(x - z_j)^{n-1}}{\prod_{k=0, k\neq i,j}^{n} (z_k - z_j)}. \quad (4)$$

Since formula (3), inorder to approximate the m -order derivative of the function $u(x)$ at every grid point, we need to develop a method to calculate the m- order derivatives of factors.

2.2. Method for calculating factor derivatives

Consider factor $L_i(x) = \frac{(x - z_0) .. (x - z_{i-1})(x - z_{i+1}) \dots (x - z_n)}{(z_i - z_0) .. (z_i - z_{i-1})(z_i - z_{i+1}) \dots (z_i - z_n)}$.

Let

$$T_i(x) = (x - z_0)(x - z_1) .. (x - z_{i-1})(x - z_{i+1}) \dots (x - z_n),$$
$$M_i = (z_i - z_0)(z - z_1) .. (z_i - z_{i-1})(z_i - z_{i+1}) \dots (z_i - z_n),$$
$$Z_i = \{(x - z_0), (x - z_1), .., (x - z_{i-1}), (x - z_{i+1}), \dots, (x - z_n)\}.$$

It is easy to see that $T_i(x)$ is the product of n elements in the set Z_i, in which all elements are present. We set $T_i(x)$ to correspond to a binary sequence H consisting of n bits that all receive the value 1. We have $T_i^{(1)}(x) = (x - z_1)(x - z_2)(x - z_3) \dots (x - z_n) + (x - z_0)(x - z_2)(x - z_3) \dots (x - z_n) + (x - z_0)(x - z_1)(x - z_3) \dots (x - z_n) + \dots + (x - z_0)(x - z_1)(x - z_2) \dots (x - z_{n-1})$.

$T_i^{(1)}(x)$ is the sum of the terms, each term is the product of $(n-1)$ elements taken from the set in the set Z_i in which the identical terms are repeated once. We let $T_i^{(1)}(x)$ correspond to a binary matrix H_1, each line is a binary vector consisting of n bits, of which 1 bit takes the value of 0. Number of rows in matrix H_1 equals C_n^1. For example

$$(x-z_1)(x-z_1)\dots(x-z_n) \leftrightarrow (0,1,1,1,\dots.1)$$
$$(x-z_0)(x-z_2)\dots(x-z_n) \leftrightarrow (1,0,1,1,\dots.1)$$

$T_i^{(2)}(x) = (x-z_2)(x-z_3)\dots(x-z_n) + (x-z_0)(x-z_3)\dots(x-z_n) + (x-z_0)(x-z_1)(x-z_4)\dots(x-z_n) + \dots + (x-z_0)(x-z_1)(x-z_2)\dots(x-z_{n-2})$.

Obviously $T_i^{(2)}(x)$ is the sum of the terms, each term is the product of $(n-2)$ elements taken in the set Z_i, in which the overlapping terms repeat 2! time. We let $T_i^{(2)}(x)$ correspond to a binary matrix H_2, each line is a binary vector consisting of n bits which 2 bits receive the value 0. Number of lines in matrix H_2 is equal to C_n^2. For example

$$(x-z_2)(x-z_3)\dots(x-z_n) \leftrightarrow (0,0,1,1,\dots.1)$$
$$(x-z_0)(x-z_3)\dots(x-z_n) \leftrightarrow (1,0,0,1,\dots.1)$$

In the general case

$T_i^{(m)}(x) = (x-z_m)(x-z_{m+1})\dots(x-z_n) + (x-z_0)(x-z_{m+1})\dots(x-z_n) + (x-z_0)(x-z_1)(x-z_{m+2})\dots(x-z_n) + \dots + (x-z_0)(x-z_1)(x-z_2)\dots(x-z_{n-m})$.

$T_i^{(m)}(x)$ be the sum of the terms, where each term is the product of $(n-m)$ elements taken in the set in the set Z_i where the duplicate terms repeat $m!$ time. We let $T_i^{(m)}(x)$ correspond to a binary matrix H_m, each line is a binary vector consisting of n bits which m bits take the value of 0. Number of rows in the matrix H_m equals C_n^m.

After determining the values $T_i^{(m)}(x)$ then $L_i^{(m)}(x) = m!\frac{T_i^{(m)}(x)}{M_i}$, We obtain approximately the m-order derivative of the mesh function determined by the formula

$$D^{(m)}U(x) = \sum_{i=0}^{n} L_i^{(m)}(x)V_i; m = 1,2,\dots \quad (5)$$

where $V = (u(z_0), u(z_1), \dots, u(z_n))$ is the value of the mesh function on neighborhood Z.

2.3. Proposed algorithm in case of irregular grid

According to the method analyzed, the matrix H_m is common to all $T_i^{(m)}(x), i = 0,.,\dots,n$. If the matrix H_m can be determined, then determining the derivatives $L_i^{(m)}(x)$ can be done using the algorithm.

```
function L_i^(m) =Factorial_Derivative(Z, x, n, m)
begin
    T_i^(m)(x) = 0; M_i = 1;
    for k =0: n
          if (i<>k) then M_i = M_i * (Z_i - Z_k);
    end
    for p =1: size(H_m)
        sp =1;
        for q =1: n
            if H_m(p, q) = 1 then sp = sp * Z_q;
        end
        T_i^(m)(x) = T_i^(m)(x) + sp;
    end
    L_i^(m) = m! * T_i^(m)(x) / M_i;
end
```

It is easy to see that the matrices H_m obtained from the backtracking algorithm generate binary sequences corresponding to m convolutional combinations of the set $\{0,1\}$.

```
function H =Geneeating_Algorithm(T, n, m, i, H)
    ok=ones(1, n);
    for j = 1: n
        if (i ==1)
            T (1)= j;ok(j)=0;
            if (m ==1)
                H =[ H; T];
            else H = Geneeating_algorithm (T, n, m, i +1, H);
            ok(j)= 1;
        end;
        if and((ok(j)== 1),( T (i − 1)<j))
            T (i)= j;ok(j)=0;
            if i == m
                H =[ H; T];
            else H = geneeating_algorithm (T, n, m, i + 1, H);
        end;
        ok(j)= 1;
    end;
end;
```

From the above results, we propose a general algorithm to approximate the mesh derivative $D^{(m)}U$.

Algorithm 1.

Input: U –grid function, N –Number of dividing points, (n+1) –Number of neighboring points, X - Irregular grid

Output: $D^{(m)}U$ – m -order derivative m on grid X

begin

Step 1: Determine the H_m according to the procedure Geneeating_Algorithm(T, n, m, i, H)

Step 2: For j =1,2,.., N +1

2.1 For each grid point x_j, determine (n+1) neighboring points Z_j and the corresponding grid function V.

2.2 Determine the coefficients $L_i^{(m)}(x)$, i =0,2,…, n according to the algorithm Factorial_Derivative(Z, x, n, m).

2.3. Determine the mesh derivative $D^{(m)}U(x_j)$ according to formula (5).

end

2.4. Proposed algorithm in case of regular grid

In case the grid is uniform with step h, consider points $x_k, k = 0,1, \dots, n$. We have some comments below:

Since $x_k = z_0 + (k-1)h, k = 0,1,\dots,n$, it follows that $x_k - z_j = (k-j)h, \forall k,j$. Given a set of neighboring points $Z = h^n\{k, (k-1), (k-2), .., (k-n)\}$, if we have $k = [\frac{n}{2}]$ then obviously Z is fixed. determined and the Z_i generated from Z depends only on i. Therefore the coefficients $L_i^{(m)}(x_k)$ will be fixed for all Z.

Let $Z0 = \{a, a+h, \dots, a+nh\}$ be the first neighbor set, we will calculate the coefficients $L_i^{(m)}(x_k), i = 0,1,\dots,n$ at points $k = 0,1, .., \left[\frac{n}{2}\right]$-1 are used to calculate $D^{(m)}U(x_k)$ at points corresponding to $k < \left[\frac{n}{2}\right]$, coefficients $L_i^{(m)}\left(x_{\left[\frac{n}{2}\right]}\right)$ is used to calculate $D^{(m)}U(x_k)$ at points corresponding to $\left[\frac{n}{2}\right] \le k \le N - \left[\frac{n}{2}\right]$. At the points corresponding to $k \ge N - \left[\frac{n}{2}\right]$ we can prove that $L_i^{(m)}(x_k) = -L_i^{(m)}(x_{N-k})$).

Since the coefficients $L_i^{(m)}(x_k)$ always contain a $\frac{1}{h^m}$, Let $B(k,i) = h^m L_i^{(m)}(x_k)$ then the derivatives at every grid point will be determined through the formula

$$D^{(m)}U(k) = \sum_{i=0}^{n} L_i^{(m)}(x_k)V(i) = \frac{1}{h^m}\sum_{i=0}^{n} B(k,i)V(i) \tag{6}$$

$$V = (U_0, U_1, \dots, U_n), k = 0,1,\dots,\left[\frac{n}{2}\right] - 1;$$

$$V = \left(U_{k-\left[\frac{n}{2}\right]}, U_{k-\left[\frac{n}{2}\right]+1}, \dots, U_{k+\left[\frac{n}{2}\right]}\right), \left[\frac{n}{2}\right] \le k \le N - \left[\frac{n}{2}\right];$$

$$V = (U_{N-n}, U_{N-n+1}, \dots, U_N), K \ge N - \left[\frac{n}{2}\right];$$

From the above comments, we build an algorithm to calculate mesh derivatives

function $D^{(m)}U$=Mesh_Derivative(B, U, N, n)

begin

$k = [\frac{n}{2}]$;

for i =0: N

if i < k

V = U (0: n); $D^{(m)}U$(0: k)= B(0: k,:)*V;

elseif i > N- k

V = U(N - n: N); $D^{(m)}U$(N- k: N)= B(N- k: N,:)*V;

else

V = U(i - k: i + k); $D^{(m)}U$(i)= B(k,:)*V;

end;

end

end

From the above results, we propose an algorithm on a regular grid

Algorithm 2.

Input: U –grid function, N –Number of dividing points, n+1 –Number of neighboring points, X - Irregular grid

Output: $D^{(m)}U$ – m -order derivative m on grid X

begin

Step 1: Determine the H_m according to the procedure Geneeating_Algorithm(T, n, m, i, H)

Step 2**:** For i=0:n

2.1 Determine vecto $V = (U_0, U_1, \dots, U_n)$, $Z = (X_0, X_1, \dots, X_n)$.

2.2 For k=0:n

+ Determine $L_i^{(m)}(x_k)$.

+ Determine matrix B:

$$B(k,i) = h^m \begin{cases} L_i^{(m)}(x_k), k < \left[\frac{n}{2}\right], \\ L_i^{(m)}\left(x_{\left[\frac{n}{2}\right]}\right), \left[\frac{n}{2}\right] \le k \le N - \left[\frac{n}{2}\right], \\ -L_i^{(m)}(x_{n-k}), k \ge N - \left[\frac{n}{2}\right]. \end{cases}$$

Step 3: Perform the procedure Mest_Derivative(B, U, N, n)

end

Theorem 3 [1]. The error of the m-order derivative approximation in the case of a regular grid is determined by the formula:

$$R_n^{(m)}(h) = \frac{(-1)^{n-m} m!\, h^{n-m+1}}{(n+1)!} \sum_{j=1, j\neq i}^{n+1} \frac{u^{(n+1)}(\xi)}{(-1)^j j!\,(n-j)!} (i-j)^{n+1} T_j^{(m)}(j)$$

From theorem 3, it can be evaluated that the accuracy of $D^{(m)}U$ is equivalent to $O(h^{n-m+1})$.

3. Experimental results and discussions

In this section, we verify the accuracy of the proposed algorithms. We give the function $u_d(x), x \in [a,b]$, thereby determining the derivative of arbitrary order $u_d^{(m)}(x)$. Divide the interval $[a,b]$ by $n+1$ points $a = x_0 < x_1 < \cdots < x_n = b$, determine the grid function value $U = (u_d(x_0), u_d(x_1), \dots, u_d(x_n))$ and the value of the derivatives $D^{(m)}U_d = \left(u_d^{(m)}(x_0), u_d^{(m)}(x_1), \dots, u_d^{(m)}(x_n)\right)$. Using the algorithm to calculate approximate derivative values on the grid $D^{(m)}U = \left(D^{(m)}(x_0), D^{(m)}(x_1), \dots, D^{(m)}(x_n)\right)$, thereby evaluating the error of the method.

Example 1: Given the function $u_d(x) = sin\frac{x}{2} + e^{-x}, x\epsilon[0,1]$, then

$$u_d^{(1)}(x) = \frac{1}{2}cos\frac{x}{2} - e^{-x},\ u_d^{(2)}(x) = -\frac{1}{4}sin\frac{x}{2} + e^{-x},$$
$$u_d^{(3)}(x) = -\frac{1}{8}cos\frac{x}{2} - e^{-x},\ u_d^{(4)}(x) = \frac{1}{8}sin\frac{x}{2} + e^{-x}.$$

Divide the interval [0,1] by the grid points

0<0.05<0.08<0.1<0.2<0.25<0.3<0.4<0.47<0.6<0.75<0.8<0.91<1.

Use algorithm 1 to calculate the derivatives $D^{(m)}U(x_i)$ then compare with the exact derivative value $D^{(m)}U_d(x_i), i = 0,1,2,\dots,13$. In table 1, we set $\varepsilon_m = \left|D^{(m)}U(x_i) - D^{(m)}U_d(x_i)\right|$.

Table 1. *Error results of the derivative on the irregular grid (Number of neighboring points n = 8)*

x	ε_1	ε_2	ε_3	ε_4
0	2.0 e-12	2.9e-10	2.0e-08	1.0 e-06
0.05	1.0 e-13	2.0e-11	2.0 e-09	8.0 e-08
0.08	2.0 e-14	1.0e-12	2.0 e-09	4.0 e-08
0.1	1.0 e-13	1.0e-11	2.0 e-09	2.0 e-08
0.2	2.0 e-13	2.0e-12	4.0 e-09	1.0 e-08
0.25	3.0 e-13	1.0e-12	9.0 e-09	1.0 e-08
0.3	1.0 e-12	4.0e-11	1.1 e-08	6.0 e-08
0.4	3.2 e-12	1.4e-12	4.0 e-08	5.0 e-08
0.47	6.1 e-12	1.4e-10	4.1 e-08	1.4 e-07
0.6	1.5 e-11	4.0e-11	8.6 e-08	3.0 e-08
0.75	9.8 e-12	2.5e-10	1.0 e-07	1.0 e-07
0.8	1.1 e-11	4.7e-10	5.2 e-08	5.1 e-07
0.9	6.4e-10	1.9e-09	1.0 e-07	2.4 e-06
1.0	4.3 e-09	2.4e-08	8.4 e-06	1.9 e-05

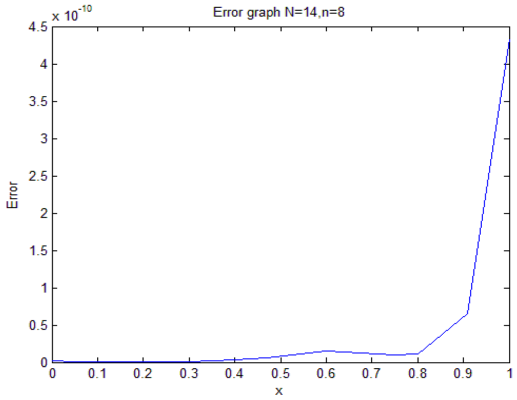

Figure 1. *Error graph with first derivative on irregular grid*

Example 2: Given the function $u_d(x) = sin\frac{x}{2} + e^{-x}, x\epsilon[0,1]$. The number of points divided on the regular grid is $(N+1)$ points. We calculate the derivatives $D^{(m)}U$ using algorithm 2. After calculation, we compare with the correct derivative value $D^{(m)}U_d$, calculation results error are in table 2, in which $E_m = \left\|D^{(m)}U - D^{(m)}U_d\right\|$, $\|.\|$ is determined as the max norm.

Table 2. *Error results of the derivative on the regular grid (Number of neighboring points n = 8)*

N	E_1	E_2	E_3	E_4
10	7.7568e-010	4.2324e-008	1.3767e-006	3.1636e-005
20	3.6441e-012	3.9606e-010	2.6038e-008	1.1798e-006
30	7.7716e-014	1.8229e-011	2.3907e-009	1.5935e-007
40	6.5059e-014	1.2019e-011	1.3482e-009	2.2535e-007

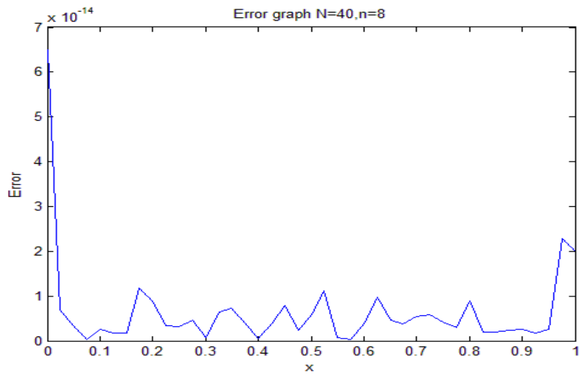

Figure 2. *Error graph with first derivative on regular grid*

The results in table 1 and table 2 show that algorithm 1 and algorithm 2 provide approximate derivative results on the grid with very high accuracy, equivalent to $O(h^{n-m+1})$, these results are consistent with the proposed theory, the largest error always occurs at the two ends. This is consistent with the properties of interpolated polynomials.

Example 3: In the case of a regular grid, we create a coefficient matrix B with 7 neighboring points, from which we get an approximate first derivative formula with $O(h^6)$ accuracy over the entire grid:

$$u^{(1)}(0) = \frac{1}{h}(-2.45u_0+6.00u_1-7.50u_2+6.6667u_3-3.75u_4+1.20u_5-0.1667u_6)$$

$$u^{(1)}(1) = \frac{1}{h}(-0.1667u_0-1.2833u_1+2.50u_2-1.6667u_3+0.8333u_4-0.25u_5+0.0333u_6)$$

$u^{(1)}(2) = \frac{1}{h}$(0.0333u_0-0.40u_1-0.5833u_2+1.3333u_3-0.50u_4+0.1333u_5-0.0167u_6)

$u^{(1)}(i) = \frac{1}{h}$(-0.0167u_{i-3}+0.15u_{i-2}-0.75u_{i-1}+0.75u_{i+1}-0.15u_{i+2}+0.0167u_{i+3})

$u^{(1)}(N-2) = \frac{1}{h}$(0.167$u_{N-6}-0.1333u_{N-5}-0.5u_{N-4}-1.3333u_{N-3}$+0.5833$u_{N-2}$+0.4$u_{N-1}-0.0333u_N$)

$u^{(1)}(N-1) = \frac{1}{h}$($-0.033u_{N-6}$+0.25$u_{N-5}-0.8333u_{N-4}$+1.6667$u_{N-3}-2.50u_{N-2}$+1.2833$u_{N-1}$+0.1667$u_N$)

$u^{(1)}(N) = \frac{1}{h}$(0.1667u_{N-6}-1.20u_{N-5}+3.75u_{N-4}-6.6667u_{N-3}+7.50u_{N-2}-6.00u_{N-1}+2.45u_N)

Similarly, we can build all the formulas that approximate the m-order derivative with (n +1) neighboring points. These formulas coincide with all known formulas. Derivative approximation formulas will be used to improve the accuracy of approximate solutions for applied problems that use numerical derivatives.

4. Conclusion

The main content of the article provides a method for determining the derivative of Lagrange factors by mapping with binary matrices. Based on that, we have proposed two higher order derivative approximation algorithms. on the grid space with high order accuracy in the cases of irregular and regular grids. We have performed experimental calculations on examples with exact solution functions. Through calculation results, we confirm that the proposed algorithm provides approximate m-th order derivative values with high order accuracy. These results will be used to improve the accuracy of solutions for the class of problems using derivative formulas on discrete data sets.

REFERENCES

[1] R. B. Srivastava and S. Shukla, *Numerical accuracies of Lagrange's and Newton polynomial interpolation: Numerical accuracies of Interpolation formulas*. LAP LAMBERT Academic Publishing, 2012.

[2] J. Li, "General explicit difference formulas for numerical differentiation," *Journal of Computational and Applied Mathematics*, vol. 183, no. 1, pp. 29-52, 2005.

[3] M. Kaur1, S. Kumar, and J. Bhatti, “Numerical Solution to Sixth Order Ordinary Differential Equation Using Three Stage Eighth Order Runge-Kutta Type Method,” *The Electrochemical Society*, vol. 107, no. 1, pp. 86-97, 2022.

[4] Q. A Dang and Q. L. Dang, "A unified approach to fully third order nonlinear boundary value problems," *J. Nonlinear Funct. Anal*, 2020, Art. no. 9, doi: 10.23952/jnfa.2020.9.

[5] Q. A Dang and Q. L. Dang, “Simple numerical methods of second and third-order convergence for solving a fully third-order nonlinear boundary value problem,” *Numer. Algor.*, vol. 87, pp. 1479–1499, 2021, doi: 10.1007/s11075-020-01016-2.

[6] Q. A Dang and T. H. Nguyen, “Solving the Dirichlet problem for fully fourth order nonlinear differential equation,” *Afr. Mat.*, vol. 30, pp. 623–641, 2019, doi: 10.1007/s13370-019-00671-6.

[7] J. Alzabut, S. R. Grace, and G. N. Chhatria, "New oscillation results for higher order nonlinear differential equations with a nonlinear neutral terms," *Journal of Mathematics and Computer Science*, vol. 28, no. 3, pp. 294-305, 2022.

[8] S. Baraket, S. Mahdaoui, and T. Ouni, "Limiting profile of the blow-up solutions for the fourth-order nonlinear Emden-Fowler equation with a singular source," *Discrete and Continuous Dynamical Systems - S*, vol. 16, no. 6, pp. 1181-1200, 2023.