A NEW EXPLICIT FORM FOR HIGHER ORDER APPROXIMATIONS OF DERIVATIVES AND ITS IMPLEMENTATION

J. A. M. P. Weerasinghe*, W. A. Gunarathna and M. A. M. Mohommad

Department of Physical Sciences, Faculty of Applied Sciences,Rajarata University of Sri Lanka, Mihinthale,Sri Lanka
*malkipiyumali@gmail.com

Numerous applications of derivatives abound in many areas such as science, engineering, economics, mathematics, medicine, optimization, etc. There is great demand in those applications for higher-order accurate computational techniques. In many situations, due to the limitations of the use of analytical methods, computational techniques have been preferred. Finite difference method is a simple computational technique, in compaction with other methods such as finite element and finite volume techniques, is used to discretise derivatives. The Taylor series is often devised to obtain finite difference forms for derivatives. In this method, a linear combination of the Taylor series of a function at various grid points are used to derive finite difference forms for derivatives. For higher-order approximations, this process includes heavy hand computations and solving large linear systems, making computational procedures cumbersome. In this study we consider higher order approximations for the first and second derivatives. For sufficiently smooth function $f(x)$ and grid size h, we define the weighted average operators: $A^1_{h,s_1,s_2,\dots,s_p}f(x) = \sum_{i=1}^{p} \lambda_i Q^1_{h,s_i}f(x)$ and $A^2_{h,r_1,r_2,\dots,r_p}f(x) = \sum_{i=1}^{p} \mu_i Q^2_{h,r_i}f(x)$ for the first and second derivatives, respectively, where $Q^2_{h,r_i}f(x) = (f(x+r_ih) - 2f(x) + f(x-r_ih)(r_ih^{-2})$,
$Q^1_{h,s_i}f(x) = (f(x+s_ih) - f(x-s_ih)(s_ih)^{-2}$,$r_i$, and s_i, $i = 1, 2, \dots, p$, are some real numbers. Then it is shown that the foregoing derivatives approximate the first and second derivatives with an accuracy of order $2p$. Furthermore, an explicit formula is constructed in the numerator-denominator forms to find the weights, λ_i and μ_i of these operators. Using the symbolic math tool box, the MATLAB codes are developed to implement the explicit formula. To attain efficient computations, separate MATLAB implementations are presented for the numerator and denominator parts of the explicit formula. Then, using the MATLAB codes, generic weight coefficients for accuracy order 4, 6, 8, and 10 are also obtained in symbolic form. Numerical tests are also presented to show the effectiveness of the proposed difference approximations.

Keywords: *First and second derivatives; Finite difference approximations; Taylor series*

A NEW EXPLICIT FORM FOR HIGHER ORDER APPROXIMATIONS OF DERIVATIVES AND ITS IMPLEMENTATION

J. A. M. P. Weerasinghe*, W. A. Gunarathna and M. A. M. Mohommad

Department of Physical Sciences, Faculty of Applied Sciences,Rajarata University of Sri Lanka, Mihinthale,Sri Lanka

*malkipiyumali@gmail.com

INTRODUCTION

Numerous applications of derivatives abound in many areas, including science, engineering, economics, mathematics, medicine, optimization, etc. (Boyer, 1959; Braun & Golubitsky, 1983). These applications have demanded various techniques to solve derivative-based problems such as initial and boundary problems. In many cases, the existing analytical methods fail to give closed form solution forms for differential equations (DE's), computational techniques have received great attention in the literature. For example, finite element and volume computations methods are presented to compute derivatives in (Christie et al., 1976; Eymard, 2000). In contrast to those methods, the finite difference technique is widely employed to solve the DEs (Khan & Ohba, 1999a; Khan & Ohba, 2003b). Taylor series is often devised to derive finite difference weights (Fornberg, 1988). For higher order approximation, this process includes heavy hand computations and solving large linear systems and thus the computational procedure is cumbersome. This study proposes two higher-order weighted average finite difference approximations for the first and second derivatives. An explicit formula is derived to compute the difference weights. A MATLAB code is also designed to implement the explicit form using the Symbolic Math Toolbox.

METHODOLOGY

This section is dedicated to presenting the derivations of new weighted average finite difference approximation for the first and second derivatives.

Define the following central difference operators:

Definition 1: Let $f \in C^2(\mathbb{R})$. Then we define the following two central difference operators:

$$Q_{h,r}^2 f(x) = \frac{f(x+rh) - 2f(x) + f(x-rh)}{(rh)^2}$$

and

$$Q_{h,s}^1 f(x) = \frac{f(x+sh) - f(x-sh)}{(2sh)},$$

where $r, s \neq 0$ and r, s are some real numbers and h is the grid size. r, and s are known as shifts.

Using Taylor series, it is not hard to see, a fixed r, that;

$$Q_{h,r}^2 f(x) = f^{(2)}(x) + 2\sum_{j=1}^{\infty} \frac{(rh)^{2j}}{(2j+2)!} f^{(2j+2)}(x) = f^{(2)}(x) + O((rh)^2) = f^{(2)}(x) + O(h^2), \quad (1)$$

as $h \to 0$.

That is, the operator $Q_{h,r}^2 f(x)$ approximates the second derivative with an accuracy of order 2. With analogous reasoning, it can be seen that $Q_{h,s}^1 f(x)$ approximates the first derivative with an accuracy of order 2. Now, to attain higher accuracy order greater than 2 for the second and first derivatives, we define the following weighted average operators:

Definition 2: For a sufficiently smooth function f, we define the following weighted average operators:

$$A_{h,r_1,r_2,\cdots r_p}^2 f(x) = \sum_{i=1}^{p} \lambda_i Q_{h,r_i}^2 f(x) \quad (2)$$

$$A_{h,s_1,s_2,\cdots s_p}^1 f(x) = \sum_{i=1}^{p} \mu_i Q_{h,s_i}^1 f(x), \quad (3)$$

where λ_i and μ_i are some real weights to be determined.

Theorem 1: Given a sufficiently smooth function $f(x)$, the operator $A_{h,r_1,r_2,\cdots r_p}^2 f(x)$ approximates the second derivative, $f^{(2)}(x)$, with an accuracy of order $2p$ if

$$\lambda_j = (-1)^{2j-p+1} \prod_{\substack{i=1 \\ i \neq j}}^{p} \left[\frac{r_i^2}{r_j^2 - r_i^2} \right] \quad (4)$$

for $j = 1,2,\cdots,p$, provided $p > 1$.

Proof: In view of (1), we have:

$$A_{h,r_1,r_2,\cdots r_p}^2 f(x) = \sum_{i=1}^{p} \lambda_i Q_{h,r_i}^2 f(x) = \sum_{k=1}^{p} \lambda_k \left(f^{(2)}(x) + 2\sum_{j=1}^{\infty} \frac{(r_k h)^{2j}}{(2j+2)!} f^{(2j+2)}(x) \right)$$

$$= \sum_{k=1}^{p} \lambda_k f^{(2)}(x) + 2\sum_{k=1}^{p} \lambda_k \sum_{j=1}^{\infty} \frac{(r_k h)^{2j}}{(2j+2)!} f^{(2j+2)}(x)$$

$$= \sum_{k=1}^{p} \lambda_k f^{(2)}(x) + 2\sum_{j=1}^{\infty} \left(\sum_{k=1}^{p} \lambda_k r_k^{2j} \right) \frac{(h)^{2j}}{(2j+2)!} f^{(2j+2)}(x)$$

$$= \sum_{k=1}^{p} \lambda_k f^{(2)}(x) + 2\sum_{j=1}^{p=1} \left(\sum_{k=1}^{p} \lambda_k r_k^{2j} \right) \frac{(h)^{2j}}{(2j+2)!} f^{(2j+2)}(x)$$

$$+2\sum_{j=p}^{\infty} \left(\sum_{k=1}^{p} \lambda_k r_k^{2j} \right) \frac{(h)^{2j}}{(2j+2)!} f^{(2j+2)}(x) \quad (5)$$

$$= \quad 1 + O(h^{2p}). \tag{6}$$

It is worth noting that the operator $A^2_{h,r_1,r_2,\cdots r_p}f(x)$ yields an approximation with an accuracy of order $2p$ for the second derivative if and only if each coefficient of $h^2, h^4, \cdots, h^{2(p-1)}$, in Equation (5), vanishes and the coefficient of $f^{(2)}(x)$, $\sum_{k=1}^{p}\lambda_k = 1$ and consequently Equation (6) comes. Now, doing so, that is, equating the coefficients of h^{2j}, for $j = 1,2,\cdots,p-1$, of (5) with those of (6) yields the linear system: $\sum_{k=1}^{p}\lambda_k r_k^{2j} = \delta_j$, where $\delta_j = 1$ if $j = 0$ and $\delta_j = 0$ if $j = 1,2,\cdots,p-1$. The foregoing system has p number of equations giving the Vandermonde matrix equation form:

$$\mathbf{V}(R_1,R_2,R_3,\cdots,R_p)\mathbf{b} = \mathbf{d}, \tag{7}$$

where $\mathbf{V} = \mathbf{V}(R_1,R_2,R_3,\cdots,R_p)$ is the Vandermonde matrix with $R_i = r_i^2$, $i = 1,2,\cdots,p$ given by (8), and $\mathbf{b}$ and $\mathbf{d}$ are column vectors given by respectively $\mathbf{b} = [\lambda_1, \lambda_2,\cdots,\lambda_p]^T$ and $\mathbf{d} = [1,0,0,\cdots,0]^T$ with superscript T denoting matrix transpose.

$$V = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ R_1 & R_2 & R_3 & \cdots & R_p \\ R_1^2 & R_2^2 & R_3^2 & \cdots & R_p^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ R_1^{p-1} & R_2^{p-1} & R_3^{p-1} & \cdots & R_p^{p-1} \end{pmatrix} \tag{8}$$

Now, employing the standard Cramer's rule, we solve the system (7) for λ_j :

$$\lambda_j = \frac{|V_j(\mathbf{d})|}{|V(R_1,R_2,R_3,\cdots,R_p)|}, \qquad j = 1,2,3,\ldots, \tag{9}$$

where $V_j(\mathbf{d})$ is a matrix obtained from $V(R_1,R_2,R_3,\cdots,R_p)$ by replacing the j-th column,R_j, of V by $\mathbf{d}$ for each $j = 1,2,3,\cdots,p$.

For example, if $p = 2$, $|V_1(\mathbf{d})| = \begin{vmatrix} 1 & 1 \\ 0 & R_2 \end{vmatrix} = R_2$, $|V_2(\mathbf{d})| = \begin{vmatrix} 1 & 1 \\ R_1 & 0 \end{vmatrix} = -R_1$.

Hence, $\lambda_1 = \frac{R_2}{R_2 - R_1}$ and $\lambda_2 = \frac{-R_1}{R_2 - R_1}$.

Now choosing the j-th column of $V_j(\mathbf{d})$ as the pivot column and using the determinant rule gives:

$$|V_j(\mathbf{d})| = (-1)^{i+j}\begin{vmatrix} R_1 & R_2 & R_3 & \cdots & R_p \\ R_1^2 & R_2^2 & R_3^2 & \cdots & R_p^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ R_1^{(p-1)} & R_2^{(p-1)} & R_3^{(p-1)} & \cdots & R_p^{(p-1)} \end{vmatrix}$$

$$= (-1)^{i+j}(R_1 \times R_2 \times \cdots \times R_p)\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ R_1 & R_2 & R_3 & \cdots & R_p \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ R_1^{(p-2)} & R_2^{(p-2)} & R_3^{(p-2)} & \cdots & R_p^{(p-2)} \end{vmatrix}$$

$$= (-1)^{i+j}\left[\prod_{\substack{i=1\\ i\neq j}}^{p} R_i\right]\left[\prod_{\substack{(1\leq i<k\leq p)\\ (i,k\neq j)}} (R_k - R_i)\right] \tag{10}$$

Using the standard formulation of the determinant of a Vandermonde matrix, we have the determinant of the Vandermonde matrix $V(R_1, R_2, R_3, \cdots, R_p)$ as follows:

$$|V(R_1, R_2, R_3, \cdots, R_p)| = \prod_{1\leq i<k\leq p} (R_k - R_i)$$

$$= \left[\prod_{k=j+1}^{p} (R_k - R_j)\right]\left[\prod_{i=1}^{j-1} (R_j - R_i)\right]\left[\prod_{\substack{1\leq i<k<p\\ (i,k\neq j)}} (R_k - R_i)\right]$$

$$= (-1)^{p-j}\left[\prod_{\substack{i=1\\ i\neq j}}^{p} (R_j - R_i)\right]\left[\prod_{\substack{1\leq i<k<p\\ (i,k\neq j)}} (R_k - R_i)\right] \tag{11}$$

Now, on substituting (10), (11) and $R_i = r_i^2$ into (9) gives:

$$\boldsymbol{\lambda_j = (-1)^{2j-p+1}\prod_{\substack{i=1\\ i\neq j}}^{p}\left[\frac{r_i^2}{r_j^2 - r_i^2}\right]}. \tag{12}$$

Hence the proof is complete.

Theorem 2: Given a sufficiently smooth function, the operator $A^1_{h,s_1,s_2,\cdots s_p} f(x)$ approximates the first derivative $f'(x)$ with an accuracy of order $2p$ if

$$\boldsymbol{\mu_j = (-1)^{2j-p+1}\prod_{\substack{i=1\\ i\neq j}}^{p}\left[\frac{s_i^2}{s_j^2 - s_i^2}\right]}$$

for $j = 1,2,3, \cdots, p$, provided $p > 1$.

Proof: The proof follows the analogous reasoning of the proof of Theorem 1.
It should be noted that Theorem 1 and Theorem 2 yield the same formulas for the weight's coefficients λ_j and μ_j. Therefore, taking $\mu_j = \lambda_j$ and $s_i = r_i$, we combine Theorem 1 and Theorem 2 to set off Theorem 3 as follows:

Theorem 3: Given a sufficiently smooth function $f(x)$, the operators $A^1_{h,r_1,r_2,\cdots r_p}f(x)$ and $A^2_{h,r_1,r_2,\cdots r_p}f(x)$ approximate the first and the second derivatives with an accuracy of order $2p$ if

$$\lambda_j = (-1)^{2j-p+1}\prod_{\substack{i=1\\ i\neq j}}^{p}\left[\frac{r_i^2}{r_j^2 - r_i^2}\right] = \frac{(-1)^{2j-p+1}\prod_{\substack{i=1\\ i\neq j}}^{p} r_i^2}{\prod_{\substack{i=1\\ i\neq j}}^{p}(r_j^2 - r_i^2)}. \quad (13)$$

IMPLEMENTATION

The coefficient λ_j in (13) are implemented in two functions, namely, the numerator (Num) and the denominator (Den). The numerator part is implemented in recursive form for efficient computation so that

$$Num_1 = (-1)^{(3-p)}\prod_{i=2}^{p} {r_i}^2;\ Num_{j+1} = \frac{r_j^2}{r_{j+1}^2}Num_j, \text{for } j = 1, 2, 3, \cdots, p-1.$$

To attain an accuracy of order $2p$, there require p shifts. Let Sh denote the set of shifts so that $\text{Sh} = \{r_1,\ r_2\ ,\cdots, r_p\ \}$. MATLAB implementation of the numerator and denominator parts are given by Listings 1 and 2. We use the MATLAB symbolic tool box. In each case, a desired even order of accuracy, that is, a value for p (even) must be given as an input argument. Then, the number of shifts required is $p/2$ and the second input argument, the shift set Sh, must be given in symbolic form. Listings 1 and 2 display four shift values corresponding to $p = 8$ as an example, but one can run them with any even integer p. Further, by combining Listings 1and 2 according to Equation 13, we can obtain λ_j.

Listing 1: MATLAB code for numerator in (13)

```
function Num = ExplicitNum(p)
clc;
clear all;
close all;
syms r1 r2  r3 r4 real % for p=8
Sh=[r1, r2,r3,r4];
L=length(Sh);
Num=sym(zeros(1,L)); % Numerator
Num1=1;
for i=2:L
    Num1=((-1)^(3-L))*((Sh(i))^2)*Num1;
end
Num(1)=Num1;
```

Listing 2: MATLAB Code for Denominator in (13)

```
function Den = ExplicitDen(p)
clc;
clear all;
close all;
syms r1 r2  r3 r4 real % for p=8
Sh=[r1, r2,r3, r4];
L=length(Sh);
Den=sym(zeros(1, L)); % Denominator
for k=1:L
    Den(k)=1;
    for i=1:L
         if (i~=k)
             Den(k)=((Sh(k))^2-
(Sh(i))^2)*Den(k);
```

```
for j=1:(L-1)

Num(j+1)=((Sh(j)/Sh(j+1)).^2)*Num(j);
end
end
                                        end
                                    end
                              end
                              end
```

RESULTS AND DISCUSSION

This section gives some numerical results obtained for some application examples using some of our higher-order approximations. The required finite difference weights are computed using Theorem 3. Table 1 depicts weight coefficient obtained in symbolic form for $p = 4, 6,\ 8$. Table 2 displays the absolute and convergence orders of the results obtained for the first(DER1) and second(DER2)derivatives of $f(x) = \sin x$ at x=π/8 using the new fourth order approximation with $r_1 =, r_2 = 2.$ Table 3 depicts absolute errors and convergence orders of the results obtained for DER1 and DER2 of f$(x) = e^x - 2x$ at $x = 0.1$ using the new sixth order approximation with $r_1 = 1, r_2 = 2, r_3 = 3.$

Table 1: Weight coefficients λ_j for accuracy order $p = 4, 6,$ and 8

Order p	Number of shifts	Weights
4	2	$\lambda_2 = \frac{r_2^2}{r_2^2 - r_1^2}$,$\lambda_2 = \frac{-r_1^2}{r_2^2 - r_1^2}$
6	3	$\lambda_1 = \frac{r_2^2 r_3^2}{(r_1^2 - r_3^2)(r_1^2 - r_2^2)}$, $\lambda_2 = \frac{r_1^2 r_3^2}{(r_2^2 - r_3^2)(r_2^2 - r_1^2)}$, $\lambda_3 = \frac{r_1^2 r_2^2}{(r_3^2 - r_2^2)(r_3^2 - r_1^2)}$
8	4	$\lambda_1 = \frac{-r_2^2 r_3^2 r_4^2}{(r_1^2 - r_4^2)(r_1^2 - r_3^2)(r_1^2 - r_2^2)}$, $\lambda_2 = \frac{-r_1^2 r_3^2 r_4^2}{(r_2^2 - r_4^2)(r_2^2 - r_3^2)(r_2^2 - r_1^2)}$ $\lambda_3 = \frac{-r_1^2 r_2^2 r_4^2}{(r_3^2 - r_4^2)(r_3^2 - r_2^2)(r_3^2 - r_1^2)}$, $\lambda_4 = \frac{-r_1^2 r_2^2 r_3^2}{(r_4^2 - r_3^2)(r_4^2 - r_2^2)(r_4^2 - r_1^2)}$

Table 1: Absolute errors and convergence orders of $f(x) = \sin x$ at $x = \pi/8$

h	Order 4 approximation				Order 2 approximation			
	Absolute errors		Convergence order		Absolute errors		Convergence order	
	DER1	DER2	DER1	DER2	DER1	DER2	DER1	DER2
1	0.0273	0.0039	-	-	0.1465	0.0309	-	-
0.5	0.0019	2.6E-04	3.8710	3.9037	0.0380	0.0079	1.9458	1.9639
0.25	1.2E-04	1.7E-05	3.9678	3.9759	0.0096	0.0019	1.9865	1.9909
0.125	7.5E-06	1.0E-06	3.9939	3.9939	0.00240	4.9E-04	1.9967	1.9978
0.0625	4.7E-07	6.5E-08	3.9985	3.9985	6.0E-04	1.3E-04	1.9992	1.9994
0.03125	2.9E-08	4.1E-09	3.9996	3.9996	1.5E-04	3.1E-05	1.9998	1.9999

Table 2: Absolute errors and convergence orders of $f(x) = e^x - 2x$ at $x = 0.1$

h	Order 6 approximation				Order 2 approximation			
	Absolute errors		Convergence order		Absolute errors		Convergence order	
	DER_1	DER_2	DER_1	DER_2	DER_1	DER_2	DER_1	DER_2
1	0.0095	0.0023	-	-	1.9E-01	9.8E-02	-	-
0.5	1.3E-04	3.2E-05	6.2099	6.1687	0.0466	0.0232	2.0540	2.0361
0.25	1.9E-06	4.9E-07	6.0526	6.0421	0.0116	0.0058	2.0135	2.0090
0.125	3.0E-08	7.5E-09	6.0132	6.0105	0.0029	0.0014	2.0034	2.0023
0.0625	4.7E-10	1.2E-10	6.0033	6.0021	7.2E-04	3.6E-04	2.0009	2.0006
0.03125	7.4E-12	1.6E-12	6.0016	6.2289	1.8E-04	8.9E-05	2.0002	2.0001

CONCLUSIONS/RECOMMENDATIONS

Higher order approximations for first and second derivatives were obtained using two weighted average operators. The operators with order p central differences approximate the derivative with an accuracy of order $2p$. Further, two explicit formulas were derived to find the weights of the proposed operators. Some numerical tests were presented to demonstrate the effects of the approximations. This explicit formulation is desirable to automate the approximations for the derivatives. We derived the explicit form for the first and second derivatives, but may be extended to other higher derivatives.

References

Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of computation, 51*(184), 699-706.

Boyer, C. B. (1959). *The history of the calculus and its conceptual development:(The concepts of the calculus).* Courier Corporation.

Braun, M., & Golubitsky, M. (1983). *Differential equations and their applications* (Vol. 2). New York: Springer-Verlag.

Christie, I., Griffiths, D. F., Mitchell, A. R., & Zienkiewicz, O. C. (1976). Finite element methods for second order differential equations with significant first derivatives. *International Journal for Numerical Methods in Engineering, 10*(6), 1389-1396.

Eymard, R. (2000). Finite volume methods, handbook of numerical analysis. *<em>Handb. Numer. Anal.*, 7, 713.

Khan, I. R., & Ohba, R. (1999a). Closed-form expressions for the finite difference approximations of first and higher derivatives based on Taylor series. *Journal of Computational and Applied Mathematics, 107*(2), 179-193.

Khan, R. K., & Ohba, R. (2003b). Taylor series based finite difference approximations of higher-degree derivatives. *Journal of Computational and Applied Mathematics, 154*(1), 115-124.