LGURJCSIT
Volume No. 1, Issue No. 3 (July- September), pp. 1-14

AN ADAPTIVE COMPUTER ALGORITHM USING NEWLY DEVELOPED IMBEDDED INTERPOLATING POLYNOMIALS TO DETERMINE THE EQUATIONS OF POLYNOMIALS AND THEIR DERIVATIVES

Aftab Ahmad Malik, Mujtaba Asad and Waqar Azeem

Abstract: - The problem of generalization of certain principles and rules in Science, Engineering, Economics and all other disciplines dominates; particularly when these schemes are required to be expressed in mathematical functions or polynomial. The adaptive computer algorithm finds and displays the equations of polynomials of various degrees passing through given set of coordinate points: x [], y []. The automatic Adaptive Algorithm interpolates and uses a newly developed imbedded family of polynomial, having coefficients of higher degree computable using the coefficients of lowers degree polynomials. Apart from determining the equations of polynomials in classical algebraic form, the adaptive computer algorithm implemented in C++, computes the equations of function passing through these points as well as the equations of differentiated polynomials for various orders. The user need to enter the point through which the polynomial passes, rest will be the responsibility of the adaptive algorithm. The Adaptive Algorithm converts the Data into Polynomial in the classical-algebraic form and then proceeds to compute the desired derivate of first or higher order. The algorithm handles Equally Spaced Data as well as Unequally Spaced Data. The superiority of the proposed method is established as compared to other existing methods.

Keywords: ***-****Adaptive computer algorithm, Numerical Differentiation, Polynomial Interpolation*

Prof. Dr. Aftab Ahmad Malik, Ph.D. University of Kent, Canterbury, England
Department of Computer Science, Lahore Garrison University Lahore, Pakistan dr_aftab_malik@yahoo.com

Mujtaba Asad, Ph.D. Scholar, UET, Lahore, Pakistan
Lecturer (CS), Lahore Garrison University Lahore, Pakistan

Waqar Azeem, Ph.D. Scholar, Punjab University, Lahore, Pakistan
Lecturer (CS), Lahore Garrison University Lahore, Pakistan

1. Background and literature Review:

Most of the techniques uses in Engineering, Science, Technology, I.T and Economics require expressing the relevant rules and principles in the form of mathematical relations, functions, polynomials or their derivatives. According to [1], sometime certain methods give wrong results for practical problems, some methods are expensive; [1] discusses the applications of numerical differentiation in Hydraulic Engineering and also the drawbacks of spline interpolation. It is proposed by [2] that error analysis can be facilitated by using derivatives from scatted data. The work presented in this paper has been influenced by the philosophy of [3] and [4] in the sense that an embedded polynomial has been developed and used. Further, the adaptive algorithm of this is modified form of the Algorithm used by [11] and [12]. The modified form is more reliable and efficient. Numerical differentiation [15] is a technique to an estimate the derivative of a function values of the function at coordinates the simplest method is to use finite difference approximations. The detailed theory behind numerical differentiation is given in [9] and [14] along with computer programs and partially in [13].

It is sometimes burdensome to evaluate the derivatives of higher order by hand from experimental data and perform numerical differentiation on scattered experimental information Cubic-spline and discrete-quadratic polynomials which are described in [18] for determining up to 3rd order derivatives. First and higher order derivatives are required in stress analysis, loaded plated and beams. Software packages are available in MATLAB regarding Engineering and biomedical technologies applications discussed in [16], [17] and [19] particularly 7-point Lagrange interpolation.

The theory and experimentation in Science and all branches of engineering, most of the time, involve variable quantities and the rates how these quantities change with respect to other parameters. Such changes are normally expressed and dealt by Calculus and sometimes by techniques of Numerical Analysis. The topic of Differentiation in Mathematics has been very strong. Differential Equations, Differential Geometry and Calculus of Variations have been special features. But unfortunately, the area of Computational Numerical Differentiation did not develop much in the literature.

2. Introduction

For the purpose scientific computing, the derivatives of first order at various points lying in the given interval are calculated, approximately, by taking difference of two consecutive function values divided by the length of the interval. Similarly, the second and order derivatives are obtained by taking difference of two lower order derivatives divided by length of relevant interval.

The Algorithm generates the equations of all the Polynomials of degree $d \leq n$ for a set of (n+1) points. It interpolates the values of the function at any number of desired values without any additional computational cost. Moreover, it generates and displays all the Equations of the first and higher order derivative up to order n. The user may ask for any number of values of first and higher order derivatives at any prescribed point x. The scheme is highly efficient, reliable and accurate.

According to [11] the famous methods for computing the Equation of an interpolated Polynomial are Newton's Forward (NF) Difference, Backward (NB) Difference, Central and Divided Difference Methods. These methods involve computation of Difference Table and differences of various orders. Large amount of computational work is required for these methods. These methods use Taylor's series for the computation of first and higher order derivatives, therefore, require large amount of computational work to determine the values of differences of higher order and the Error term.

The exits another technique reported in [7] for Numerical Differentiation for a curve f(x) passing through the given n+1 points to Interpolate a Polynomial of degree n through these points and approximate the function at intermediate points as given by the following relation:

$$f(x)= P^{(n)}(x) + R^{(n)}(x) \qquad (1)$$

where, the second term on right hand side is the Error term. Then the first and higher order derivatives can be approximated [12] as

$$D^m f(x)= D^m P^{(m)}(x) + D^m R^{(m)}(x) : m=1,2,.., n \qquad (2)$$

where, D is the differential operator and D =d/dx. It is observed that the term $D^m R^{(n)}(x)$ requires almost equal amount of computational work what is required for the $D^m P^{(m)}(x)$. The term $R^{(n)}(x)$ can be minimized as described in [7]. It is observed that in case the length h_i of all

subintervals is reasonably small then one can estimate f(x) as:

$$f(x) \approx P^{(n)}(x) \quad (3)$$

Assumptions for Accuracy and Reliability

The term $R^{(n)}(x)$ can be reduced by the following considerations:

- h=step size=$|x_i - x_j|$ for $i \neq j$ must be smaller
- The number of Data Points should be larger.
- The function values $y_i=f(x_i)$ at any point $x=x_i$ must be accurate and not rounded off.

To compute the coefficients of interpolated polynomials $P^{(n)}(x)$, large number of mathematical operations (MO) is required, affecting the efficiency of these methods. Other difficulty is that [12] the Newton's Forward and Backward interpolation methods work with Equally Spaced Data and do not accept Unequally Spaced Data i.e. all the step sizes (subintervals) i.e. $(x_i - x_j)$ for $i \neq j$ must be of same length.

To find out coefficients $B_i^{(n)}$, the following equations [8] are formulated,

$$y_i = \sum_{i=0}^{i=n} B_i^{(n)} x_i^i \quad (4)$$

where the Vandermonde determinant $\det(x_i^i) \neq 0$. The famous methods in this case are Crammer's Rule (CR), Gauss Elimination method (GE), Gauss Jordan Elimination method (GJD), Crout's Method (CM), Gauss Jacobi Iterative Method (GJ), Gauss Seidel Iterative Method (GS), Newton's Forward (NF) and Newton's Backward Difference (NB) interpolation techniques. These are also used to find out the equation of a polynomial. They are expensive methods, discussed in [6], [7], [8], [9], [10] and [12].

It has been established, in view of the recursive relations, that the new Interpolation technique of [12], using Imbedded Polynomials for finding out the equations of $P^{(n-1)}(x)$ and $P^{(n)}(x)$ is most efficient and reliable as compared to other methods. While developing the equation of n^{th} degree Polynomial, the Adaptive Algorithm of [12] computes equations of all polynomials having degree less than n, free of computational cost. Therefore, in this paper, for determining Polynomial Equations, a Family M of [12] is applied to the problem of Numerical Differentiation where the Algorithm of this paper computes a family M of polynomials passing through given points having degree from 0 to n:

The coefficients of all members of the Family M such as $P^{(m)}(x)$ and $P^{(k)}(x)$ for $0 \leq m,k \leq n$; are inter-related and

- $P^{(n)}(x)$ polynomial of degree n,
- $P^{(n-1)}(x)$ polynomial of degree n-1,
- $P^{(n-2)}(x)$ polynomial of degree n-2
- .
- .
- $P^{(1)}(x)$ polynomial of degree 1,
- $P^{(0)}(x)$ polynomial of degree zero ..... (5)

interconnected which are developed in [12].

2.1 Remarks on Lagrange Interpolation Method:

The equation of Interpolated Polynomial can also be determined by the Lagrange Interpolation Method for lower degree n polynomials $P^{(n)}(x)$. But an inherent difficulty in using Lagrange Method [9] is that computing involves more mathematical operation when the number of given points increase. Hence no part of the previous calculations can be re-used to compute (interpolate) the new values y_p or Polynomial's Equations of higher degree. The Lagrange Method (LM) becomes unmanageable by hand for more than 4 or 5 points [12] and becomes inefficient for large number of points as compared to new technique using Family M.

2.2 Justification of using Imbedded Polynomial's Family

A Polynomial of degree n passing through N=n+1 point

$$P^{(n)}(x) = \sum_{i=0}^{i=n} B_i^{(n)} x^i$$

Where the imbedded coefficients are given by a recursiv

$$B_p^{(t)} = (-1)^{(t+p)} B_t^{(t)} * S_{(t-p)}^{(t-1)} + B_p^{(t-1)} : 0 \leq p < n \text{ an}$$

$$B_n^{(n)} = \sum_{i=0}^{i=n} A_i^{(n)}$$

where $$A_i^{(n)} = y_i \prod_{j=0}^{j=n} [1/(x_i - x_j)] \quad : \quad i \neq j$$

The Array $S_p^{(n)}$ is a lower triangular matrix and it is initiali as

$$S_0^{(t)} = x_i : 0,1,\ldots, n \qquad (11)$$

And all elements of 0*th* row as:

$$S_0^{(t)} = 0 : t=0,1,\ldots, n \qquad (12)$$

The upper most non-zero diagonal of $S_p^{(n)}$:

$$S_{t+1}^{(t)} = S_t^{(t-1)} * S_0^{(t)} \quad : \quad 2 \leq t \leq n. \qquad (13)$$

All other subscripted variables $S_p^{(t)}$ are given below:

$$S_p^{(t)} = S_p^{(t-1)} + S_{p-1}^{(t-1)} * S_0^{(t)} : \quad p \leq t \leq n \qquad (14)$$

The performance of the methods quoted above can be visualized by the amount of mathematical and arithmetical operations MO as computed by [12]:

In order to facilitate the computational task to carry out Numerical Differentiation as well as the design of Adaptive Algorithm of this paper, we proceed as follows:

The Table 1 obviously provides enough justification for selecting the Imbedded Polynomials formulas expressed by equations (6), (7) and (8) for the purpose of procedures of this paper

Table 1: Amount of Mathematical Operations (MO) for various methods for N=n+1 Points

	Method	N=3	N=4	N=5	N=6	N=7
1.	**Cramer's Rule**	**63**	**254**	**1535**	**10758**	**86023**
2.	Gauss Jacobi's Iterative (for 20 iterations)	333	638	850	1335	1939
3.	Gauss Seidel's Iterative (for 20 iterations)	333	638	850	1335	1939
4.	**Lagrange Method**	**88**	**228**	**546**	**1477**	**1638**
5.	Crout's Method	73	173	337	581	821
6.	Newton's Forward	21	48	91	203	505
7.	Newton's Backward	21	48	91	203	505
8.	Gauss Elimination	31	68	125	206	315
9.	Gauss Jorden Elimination	40	84	150	242	364
10.	**Malik and Ahmad[11] (Method of this paper)**	**14**	**20**	**26**	**32**	**38**

Table 1: Amount of Mathematical Operations (MO) for various methods for N=n+1 Points

	Method	N=3	N=4	N=5	N=6	N=7
1.	**Cramer's Rule**	**63**	**254**	**1535**	**10758**	**86023**
2.	Gauss Jacobi's Iterative (for 20 iterations)	333	638	850	1335	1939
3.	Gauss Seidel's Iterative (for 20 iterations)	333	638	850	1335	1939
4.	**Lagrange Method**	**88**	**228**	**546**	**1477**	**1638**
5.	Crout's Method	73	173	337	581	821
6.	Newton's Forward	21	48	91	203	505
7.	Newton's Backward	21	48	91	203	505
8.	Gauss Elimination	31	68	125	206	315
9.	Gauss Jorden Elimination	40	84	150	242	364
10.	**Malik and Ahmad[11] (Method of this paper)**	**14**	**20**	**26**	**32**	**38**

.

3.0 Development of New Technique for Numerical Differentiation

$D^m P^{(n)}(x) = \sum_{j=0}^{j=n} \{ \underline{X}^{\underline{\alpha}[m][j]} \Phi_j^{(m)} \}$ such that m=0,1,2,…..,(n-1) where $\underline{X}^{\underline{\alpha}[m][j]}$ is a two dimensional matrix and $\underline{\alpha}$ is also 2 dimensional matrix containing powers of x as shown below:

Example $\alpha[m][j]$:

0th Row : m=0: $[\alpha_0\ \alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6] = [0\ 1\ 2\ 3\ 4\ 5\ 6]$

1st Row : m=1: $[\alpha_0\ \alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6] = [0\ 0\ 1\ 2\ 3\ 4\ 5]$

last Row : m=n: $[\alpha_0\ \alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$$\alpha[m][j] = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Initially:
0! = 1! =1; zero=0;
Then Set all Q[m][j]=zero : m,j=0,1,2,…,n; Q[n][n]=n!;
{ Q[0][j]=1; Q[1][j]=j; j=0,1,2,…,n }
Loop1 begins : m=2,…,n;
    Loop2 begins : j=0,1,2,…,n;
    {if(m<=j) Q[m][j]=j ! /(j-m)!;
    else if(j<=m) Q[m][j]=zero;}
    Loop2 Ends
Loop1 Ends

Assumptions: The length of all closed subintervals $(x_i - x_j)$ must be reasonably small. All the INPUT Data points x_i, y_i must be arranged in ascending order with respect x_i before use.

The Pseudo codes

The Pseudo code:

Usefulness of the Scheme:

- It is reliable and returns the answer to high degree of accuracy
- The (n+1) input Data points x, y are entered by the user and it returns all

h=1; Number of Points =Nine

x_i	Method	0	1	2	3	4	5	6	7	8
$f(x_i)$	Actual value	1	2	5	10	17	26	37	50	65
$P(x_i)$	NM	1	2	5	10	17	26	37	50	65
$P'(x_i)$	NM	0	2	4	6	8	10	12	14	16
$P''(x_i)$	NM	-	2	2	2	2	2	2	2	2
$P'''(x_i)$	NM	-	0	0	0	0	0	0	0	0

where

$$\underline{X}^{\alpha[m][j]} = \begin{pmatrix} x^0 & x^1 & x^2 & x^3 & x^4 & x^5 & x^6 \\ 0 & x^0 & x^1 & x^2 & x^3 & x^4 & x^5 \\ 0 & 0 & x^0 & x^1 & x^2 & x^3 & x^4 \\ 0 & 0 & 0 & x^0 & x^1 & x^2 & x^3 \\ 0 & 0 & 0 & 0 & x^0 & x^1 & x^2 \\ 0 & 0 & 0 & 0 & 0 & x^0 & x^1 \\ 0 & 0 & 0 & 0 & 0 & 0 & x^0 \end{pmatrix}$$

$$\begin{pmatrix} D^0 \\ D^1 \\ D^2 \\ D^3 \\ D^4 \\ D^5 \\ D^6 \end{pmatrix} \quad P^{(n)}(x) = \begin{pmatrix} x^0 & x^1 & x^2 & x^3 & x^4 & x^5 & x^6 \\ 0 & x^0 & x^1 & x^2 & x^3 & x^4 & x^5 \\ 0 & 0 & x^0 & x^1 & x^2 & x^3 & x^4 \\ 0 & 0 & 0 & x^0 & x^1 & x^2 & x^3 \\ 0 & 0 & 0 & 0 & x^0 & x^1 & x^2 \\ 0 & 0 & 0 & 0 & 0 & x^0 & x^1 \\ 0 & 0 & 0 & 0 & 0 & 0 & x^0 \end{pmatrix} \begin{pmatrix} Q_0^{(m)} \\ Q_1^{(m)} \\ Q_2^{(m)} \\ Q_3^{(m)} \\ Q_4^{(m)} \\ Q_5^{(m)} \\ Q_6^{(m)} \end{pmatrix}$$

Define : $\Phi_i^{(m)} = Q_i^{(m)} * B_j^{(m)}$

the equations of polynomial of degree less or equal to n,

- It returns all the equations of differentiated polynomial of degree less or equal to n-1,
- It calculates the interpolated value of the function at any requested point
- It computes the values of the first and higher derivatives at any requested point

In order to establish superiority of New Method (NM) as compared to others, the following examples are tested and reported. The complicated functions also have been tested.

Test Functions:

Example 1: A curve passes through the following points (x_i , $f(x_i)$). Compute the derivative at all the given points and step size:

The Equation polynomial returned by NM: $y=f(x)=P(x)=1+x^2$

The Equation of first order derivative returned by NM: $y'=f'(x)=P'(x)=2x$

The Equation second order derivative returned by NM: $y''=f''(x)=P''(x)=2$

Example 2: A curve passes through the following seven points (x_i, $f(x_i)$). The Data and result using Divided Difference Method (**DDM)** to compute the first order derivative at x=3.3 is taken from [10]. Compare the results with adapt AlgoDerivative of the New Method NM.

x_i	Method	1.30	1.90	2.50	3.10	3.70	4.30	4.90
$f(x_i)$	Actual value	3.669	6.686	12.182	22.198	40.447	73.7	134.290

Equation of the Polynomial developed by NM:

$P(x)=5.817763 - 11.745625\ x + 14.213159\ x^2 - 7.577301\ x^3 + 2.489935\ x^4 - 0.410594\ x^5 + 0.033281\ x^6$

Equation of first order Derivative developed by NM:

P '(x)= -11.745625 + 28.426318 x

The Polynomial of degree 8 returned by adaptAlgoDerivative:

$y=f(x) =P(x)=x^8 -7\ x^7 + 5\ x^4 -20$

Results of Example 3:

x_i	-4	-3	-2	-1	0	1	2	3	4
$f(x_i)$	182786	22817	1394	35	2	-19	-718	-8881	-49150
$f'(x_i) = P'(x_i)$	-334016	-54297	-4560	-137	0	-81	-2192	-18225	-69312
$f''(x_i) = P''(x_i)$	531872	113166	13472	530	0	-298	-5824	-30438	-71200
$f'''(x_i) = P'''(x_i)$	-720984	-201198	-34632	-2046	-120	-1134	-12648	-37182	-31896
$f^{iv}(x_i) = P^{iv}(x_i)$	806520	294960	74040	7680	120	-4080	-20040	-22560	-53880
$f^{viii}(x_i) = P^{viii}(x_i)$	40320	40320	40320	40320	40320	40320	40320	40320	40320

$-22.731903\ x^2 + 9.959741\ x^3 - 2.05297\ x^4 + 1.99686\ x^5$

f '(3.3) computed by **DDM**= 26.875

f '(3.3) computed by N**M**= 27.116

Exact Value given by [12] = 27.113

Note: The coefficients of P(x) and P'(x) are rounded.

Example 3: A curve passes through the points $(x_i, f(x_i))$. Compute first, second, third, fourth and eighth order derivatives. Compute the derivative at all the given points. The points are obtained from the following equation of function:

$y=f(x) =x^8 -7\ x^7 + 5\ x^4 -20\ x^3 + 2$

$x^3 + 2$

Number of Points =Nine; Step Size=1:

Results of NM with Accuracy & Reliability: 100%

Example 4: A curve passes through the following six points (x_i , $f(x_i)$). The Data and result using Newton's Forward Difference Formula (**NFD)** to compute the first order derivative at x=1.5 is taken from [10]. Compare the results with adaptAlgoDerivative of the New Method NM for the function:

Step size= h=0.1; Number of points=6; Results of NM with Accuracy & Reliability: 100%

Results of Example 4:

$x_i \rightarrow$	Method ↓	1.0	1.1	1.2	1.3	1.4	1.5
$y_i=f(x)$		16.40	19.01	21.96	25.29	29.03	33.21
$f'(x)=P'(x)$	NM	24.57933333	27.72	31.345	35.30333333	39.545	44.12
$f'(x)$	NFD	-	-	-	-	-	44.12
$f''(x)=P''(x)$	NM	28.25	34.1666667	38.0833333	41	43.91666667	47.833333
$f''(x)$	NFD	-	-	-	-	-	47.83
$f'''(x)=P'''(x)$	NM	72.5	47.5	32.5	27.5	32.5	47.5
$f^{iv}(x)=P^{iv}(x)$	NM	-300	-200	-100	0	100	200
$F^{v}(x)=P^{v}(x)$	NM	1000	1000	1000	1000	1000	1000

REFERENCES

[1] John D. Fenton, "Interpolation and numerical differentiation in civil engineering problems", from Civil Engineering Transactions, I.E. Aust., CE36, 331-337 (1994), Department of Mechanical Engineering, Monash University, Clayton, Victoria, Australia 3168

[2] J. Cheng , X. Z. Jia & Y. B. Wang, "Numerical differentiation and its applications", Pages 339-357 Accepted 24 Apr 2006, Published online: 08 May 2007 Download citation; http://dx.doi.org/10.1080/17415970600839093

[3] A C Genz and A A Malik, Remarks on Algo :006: An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region", Journal of Computational Applied Mathematics Volume 6 pp 295–302, Belgium [1980].

[4] A.C Genz and A.A Malik, An imbedded family of fully symmetric numerical integration rules, . SIAM Journal of Numerical Analysis Volume 20 3, Philadelphia USA (June), PP 580-588 [1983].

[5] T.H.Cormen, C.E. Leiserson, R.L.Rivest and Clifford Stein, Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill,. ISBN 0-262-03293-7, Problem 2-3 PP. 39 and page 823 of section 30.1: Representation of Polynomials, 2001.

[6] D.V Griffiths., I.M. Smith, Numerical Methods For Engineers, Chapman & Hall, [1990].

[7] B Carnahan., H.A. Luther, Applied Numerical Analysis, Krieger Publishing Company, Malabar Florida ISSBN: 089464-486-6 [1997].

[8] R.W Hamming, Numerical Methods for Scientists and Engineers, II Edition; Dover Publishing Inc. New York [1975] PP:227-33.

[9] P. Madhumangle, "Numerical Analysis for Scientists and Engineers"(Theory and Programs)" Naroosa Publishing House New Delhi, [2007] PP: 403-432

[10] C.F. Gerald and Wheatley P.O , Applied Numerical Analysis, 6th Edition; Addison Wesely Publishing Company, PP 358-362 [1997]

[11] A A Malik and N Ahmad, New Dimensions in Computer Methods for Binary Based Polynomials, Construction of new Polynomials, Research Journal, B.Z. University Multan, Pakistan; Vol 8, , ISSN:1021-1012, [1996-97] pp:25-49

[12] A A Malik & Muzaffar Baig,"A Technique for Numerical Integration and Differentiation using coefficients of imbedded interpolation Polynomials", Journal of Research(Science) ; ISSN:1021- 1012 Volume 7, No 1 and 2 June & December,1995, pp:17-27.

[13] S.D. Conte and Carl de Boor, Elementary Numerical Analysis: An Algorithmic Approach, McGraw-Hill, 1972. William H. Press, [14] Brian P. Flannery , Saul A. Teukolsky , William T. Vetterling, "Numerical Recipes in Fortran 77": The Art of Scientific Computing [Hardcover] ; Cambridge University Press, USA

[15]Numerical Differentiation Formulas http://www.tutorvista.com/bow/numerical-differentiation-formulas

[16] Wu, J.K.; Long, J.; He, F.; He, Q.L., Numerical

differentiation based algorithm for power measurement ; Power Electronics and Drive Systems, 2003. PEDS 2003. The Fifth International Conference on Digital Object Identifier: 10.1109/PEDS.2003.1282805 Publication Year: 2003 , Volume: 1 Page(s): 302 - 307 Vol.1

[17]Steven Chapra, Raymond Canale, Numerical Methods for Engineers: With Software and Programming Applications, McGraw Hill, Austalia and New Zealand ,ISBN13: 9780071121804

[18] R. E. Rowlands, T. Liber, I. M. Daniel and P. G. Rose, Experimental Mechanics Volume 13, Number 3, 105-112, DOI: 10.1007/BF02323967 http://www.springerlink.com/content/f85h3253h25x4913/

[19] Alkis Constantinides / Navid Mostoufi, Numerical Methods for Chemical Engineers with MATLAB Applications, : Prentice Hall , Verlag: CD ISBN13: 9780130138514 ISBN10: 0-13-013851-7