A New Newton-Wavelet Algorithm to Solve Non-Linear Equations

Kais I. Ibraheem **Riyad M. Abdullah**

College of Education
University of Mosul, Iraq

Received on: 16/02/2005 **Accepted on: 09/05/2005**

ABSTRACT

In this research, a new algorithm to solve the non-linear equations (f(x)=0) was developed. The new method was called Newton –Wavelet which can be defined as a mix between two methods, Newton and wavelet. By applying this algorithm on seven examples and compared the result with the Broyden method it has shown a good efficiency. The new method shows that it can decrease the number of iterations and then decrease the time needed to solve the used equations. This algorithms considered a new technique of mixes between two subjects the first one is a numerical analysis (by using Newton method) and the second is an image processing and data compression (by using Wavelet analysis), The basic aim of this research is decrease the time of solution.

Keywords: Newton method, Broyden method, Wavelet method

خوارزمية Newton–Wavelet الجديدة لحل المعادلات غير الخطية

قيس إبراهيم **رياض عبد الله**

كلية التربية، جامعة الموصل

تاريخ الاستلام: 2005/02/16 **تاريخ القبول: 2005/05/09**

الملخص

فـي هـذا البحـث تـم تطـوير خوارزميـة جديـدة لحـل المعـادلات غيـر الخطيـة (f(x)=0) الخوارزمية الجديدة سميت (Newton –wavelet) والتي هي عبارة عن دمج بين طريقة نيوتن و طريقة الموجـة القصـيرة. تـم تطبيـق هـذه الخوارزميـة علـى سبعة أمثلـة وقورنـت نتائجهـا مـع طريقـة Broyden وقد أثبتت كفاءة جيدة فـي الحل. من أهم مميزات خوارزميـة (Newton –wavelet) أنهـا تعمـل علـى تقليـل عـدد التكـرارات (iterations) ممـا يـودي إلـى اختـزال الوقـت الـلازم لحـل المعـادلات المستخدمة. تعتبر هذه الخوارزمية تقنيـة جديدة حيث تـم الدمج بين موضـوعي التحليل العددي (استخدام طريقة نيوتن) و معالجه الصور وكبس البيانات (استخدام الموجـة القصـيرة) والغايـة الأساسية هي السرعة في أيجاد الحل.

الكلمات المفتاحية: طريقة نيوتن، طريقة Broyden، طريقة الموجة القصيرة.

1. Introduction

In this paper, we introduced three methods for solving non-linear equations, the selected methods are chosen from many wavelet methods. The problem of solving non-linear equations arises frequently and naturally from the study of a wide range of practical problems. The problem may involve a system of non-linear equations in many variables or one equation in one unknown. We shall initially confine ourselves to considering the solution of one equation in one unknown. The general form of the problem may be simply stated as finding a value of the variable *x* such that $f(x) = 0$, where *f* is any non-linear function of *x*. The value of *x* is then called a solution or root of this equation and may be just one of many solutions [4]. We have a solving system of linear equation in paper [5]. Then, we continue the work with the system of non-linear equation by using Wavelet also.

2. Newton Algorithm [2]

Compute $f(x_0), f'(x_0)$

Set $x_1 = x_0$

IF $(f(x_0) \neq 0)$ AND $(f'(x_0) \neq 0)$

Repeat

Set $x_0 = x_1$

Set $x_1 = x_0 - f(x_0)/f'(x_0)$

Until $(|x_0 - x_1| < \text{tolerance value1})$ OR

$(|f(x_1)| < \text{tolerance value2})$

3. Broyden's Method

The method of Newton does not provide a practical procedure for solving any but the smallest systems of non-linear equations. As we have seen, the method requires the user to provide not only the function definitions but also the definitions of the n^2 partial derivatives of the functions. Thus, for a system of 10 equations in 10 unknown, the user must provide 110 function definitions.

To deal with this problem, a number of techniques has been proposed but the group of methods which appears most successful is the class of methods known as the quasi-Newton methods. The quasi-Newton methods avoid the calculation of the partial derivatives by obtaining approximation to them involving only the function values. The set of derivatives of the functions evaluated at any point x^r may be written in the form of the Jacobian matrix.

$J_r = \left[\partial f_i(x^r)/\partial x_j\right]$ for $i = 1,2,3,\ldots,n$ and $j = 1,2,3,\ldots,n$ (9)

The quasi-Newton methods provide an updating formula, which give successive approximations to the Jacobian for each iteration. Broyden and others have shown that under specified circumstances these updating formula provides satisfactory approximations to the inverse Jacobian. The structure of the algorithm suggested by Broyden is:

1. Input an initial approximation to the solution. Set the counter *r* to zero.
2. Calculate or assume an initial approximation to the inverse Jacobian B^r.
3. Calculate $p^r = -B^r f^r$ where $f^r = f(x^r)$.
4. Determine the scalar parameter *t* such that $\left\| f(x^r + t_r p^r \right\| < \left\| f^r \right\|$.
5. Calculate $x^{r+1} = x^r + t_r p^r$.
6. Calculate $f^{r+1} = f(x^{r+1})$. If $\left\| f^{r+1} \right\| < \varepsilon$, where ε is a small preset positive quantity), then exit. If not continue with step (7)
7. The use the updating formula to obtain the required approximation to the Jacobian

$$B^{r+1} = B^r - (B^r y^r - p^r)(p^r)^T B^r / \{(p^r)^T B^r y^r\} \text{ where } y^r = f^{r+1} - f^r$$

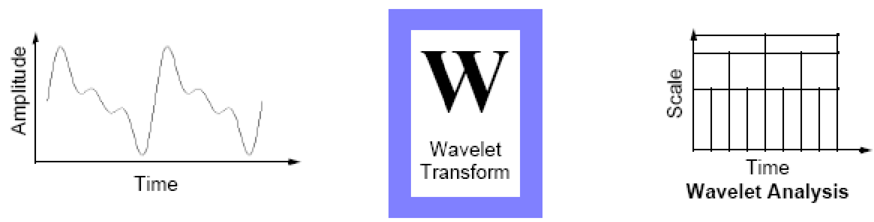
8. Set $i = i + 1$ and return to step (3)

The initial approximation to the inverse Jacobian B is usually taken as a scalar multiple of the unit matrix. The success of this algorithm depends on the nature of the functions to be solved and on the closeness of the initial approximation to the solution. In particular, step (4) may present major problems. It may be very expensive in computer time and to avoid this *t* is sometimes set as a constant, usually one or smaller. This may reduce the stability of the algorithm but speed it up.

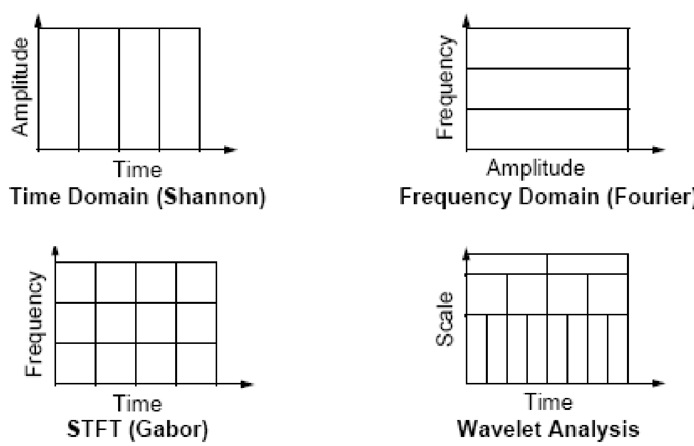
It should be noted that other updating formula has been suggested and it is fairly easy to replace the Broyden formula by others in the above algorithm. In general, the problem of solving a system of non-linear equation is a very difficult one. There is no algorithm that is guaranteed to work for all systems of equations. For large systems of equations the variable algorithms tend to require large amounts of computer time to obtain accurate solutions [2].

4. Wavelet Analysis

Wavelet analysis represents the next logical step: a windowing technique with variable-sized regions. Wavelet analysis allows the use of long time intervals where we want more precise low-frequency information, and shorter regions where we want high-frequency information.



Here's what this looks like in contrast with the time-based, frequency-based, and STFT views of a signal:



You may have noticed that Wavelet analysis does not use a time-frequency region, but rather a time-*scale* region [3].

A Wavelet is new family of basis function that can be used to approximate general functions [6]. Wavelet is a waveform of effectively limited duration that has an average value of zero. Compare Wavelets with sine waves, which are the basis of Fourier analysis. Sinusoidal do not have limited duration, they extend from minus to plus infinity. And where sinusoids are smooth and predictable, Wavelets tend to be irregular and asymmetric.

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, Wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original Wavelet or mother Wavelet [3].

Both Fourier and Wavelet transform are extensively utilized in studying signals-- waves which could be continuous of time or waves which are available only at discrete instances of time. A matrix could conveniently be considered as a row-wise or column-wise arrangement of discrete signals, as such it is amendable to transform analysis. If such an operation is performed on a matrix equation $Ax = b$, a transformed equation $WAx = Wb$ is obtained (W is a Wavelet). From this, one could write $(WAW^{-1})(Wx) = Wb$. Choosing, for instance, orthogonal transform W, a relation $(WAW^{T})Wx = Wb$, - similar to block triangularization operation - which avoids costly inversion operation is now on hand to proceed with the computation of the desired numerical solution. An interesting common property of this method is that a Wavelet transform of a dense matrix gives rise to a sparse matrix [1]. Hence an $O(N^3)$ cost of computing could be reduced into much cheaper operation. [5] gives a brief presentation on Wavelet and Wavelet transforms.

Newton-Wavelet Algorithm

Start with an initial estimate x_0;

For $k = 0,1,2,...$:

Compute $F(x_k)$, $J(x_k)$ Jacobian of $F(x)$ at x_k;

Solve the linear equation for S_k: $J(x_k)\, S_k = -F(x_k)$ using wavelet Haar

i. $[J(x_k)]\, S = - F(x_k)$, $S = [x_{k+1} - x_k]$
ii. Calculate matrices [w] = wavelet ;
iii. Calculate aw = w$J(x_k)$w';
iv. Calculate bw = w$F(x_k)$;
v. Calculate xw = inv(aw)bw;
vi. Calculate S = w'xw;

If $\left|x_{k+1} - x_k\right| \leq \in x_k$, exit;

otherwise set $k = k+1$, and go to (i).

4.1 Haar Wavelet

Haar wavelet is defined as

$$\psi_{\mathrm{H}}(x) = \begin{cases} 1 & \text{for} \quad 0 \le x < \frac{1}{2} \\ -1 & \text{for} \quad \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Following Fourier, any wavelet $\psi(x)$ could be used as a basic block to build any wave $f(x)$,

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{j,k} \psi_{j,k}(x)$$

with

$\psi_{j.k}(x) = 2^{j/2}\psi(2^j x - k)$, for all $j,k \in Z$.

The coefficients $c_{j,k}$ are computable from:

$$c_{j,k} = \left\langle f, \psi_{j,k} \right\rangle .$$

Also following the idea of Fourier transform, Wavelet transform W_ψ of any wave *f(x)* can now be defined as follows:

$$(W_\psi f)(b,a) = |a|^{\frac{-1}{2}} \int_{-\infty}^{\infty} f(x)\overline{\psi(\frac{x-b}{a})}dx .$$

The coefficients $c_{j,k}$ are now computable from the following relation:

$$c_{j,k} = (W_\psi f)\left(\frac{k}{2^j}, \frac{1}{2^j}\right) \text{ [1].}$$

5. Numerical Results

We end our discussion of the solution of non-linear systems of equations by comparing the performance of the Newton, Broyden and Newton-Wavelet functions developed in section 2,3 and 4. The following script calls function provides the number of iterations. Then running a similar experiment on six examples, the following results are obtained:

Example 1:

The following system of two equations in two variables is described below:

$$xe^{xy+0.8} + e^{y^2} = 3$$
$$x^2 - y^2 - 0.5e^{xy} = 0$$

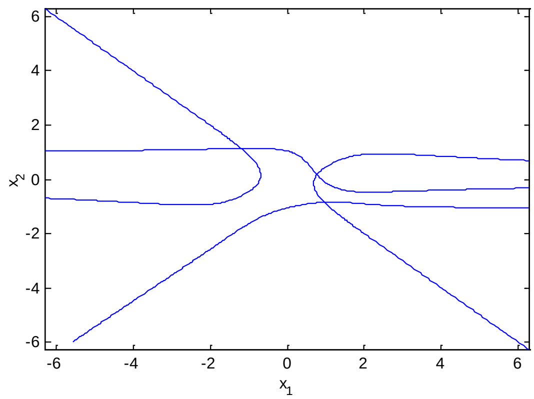


Fig (1): System of the two equations above.

The results of initial approximation for the root shown in Fig. (1):

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
1 1	0.7750 0.1716	5	-	-	0.7749 0.1725	4
-1 -1	0.7750 0.1716	5	0.7750 0.1716	29	0.7750 0.1716	5
0.5 0.5	0.7750 0.1716	3	-	-	0.7750 0.1716	2

Example 2:

The following system of two equations in two variables is described below:

$$xy^3 - 2\sin(1+x) = 1$$
$$e^{1-y^2} + x^2 y = 2$$

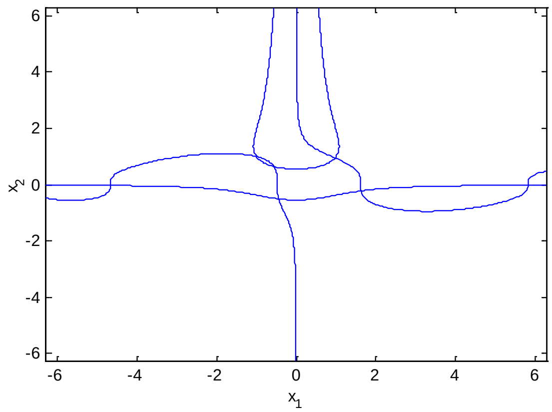


Fig (2). System of the two equations above.

The results of initial approximation for the root are shown in Table (2):

Table (2) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
1 1	0.9734 0.9521	2	0.9733 0.9521	7	0.9734 0.9521	2
-1 -1	-0.4425 -0.5082	4	0.9733 0.9521	40	-0.4426 -0.5083	3
0.5 0.5	0.9733 0.9521	9	-0.4425 -0.5082	11	0.9755 0.9522	8

Example 3:

The following system of three equations in three variables is described below:

$$3.1x_2 - \cos(x_1 x_2) = 0.6$$

$$1.03e^{-x_1 x_2} + 1.95x_3 = 11$$

$$x_2^2 - 83(x_1 + 0.11)^2 + \sin x_3 = 0.97$$

The results of initial approximation for the root are shown in Table (3):

Table (3) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
-1 -1 -1	0.0179 0.5161 -6.1644	6	-	-	0.0179 0.5161 -6.1644	5
2 7 3	-0.2342 0.5138 -6.2368	8	-	-	-0.2345 0.5138 -6.2369	7
2 4 9	0.0178 0.5161 -6.1644	7	-	-	0.0179 0.5161 -6.1644	6

Example 4:

The following system of four equations in four variables is described below:

$$x_1 = 1$$

$$x_1 x_2 = 1$$

$$x_1 x_2 x_3 = 1$$

$$x_1 x_2 x_3 x_4 = 1$$

The results of initial approximation for the root are shown in Table (4):

Table (4) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
-1 -1 -1 -1	1 1 1 1	3	1 1 1 1	15	1 1 1 1	3
2 2 2 2	1 1 1 1	4	1 1 1 1	14	1 1 1 1	4
5 5 5 5	1 1 1 1	4	-	-	1 1 1 1	4

Example 5:

The following system of eight equations in eight variables is described below:

$$8(x_1 - x_2^2) = 0$$

$$16x_2(x_2^2 - x_1) - 2(1 - x_2) + 8(x_2 - x_3^2) = 0$$

$$16x_3(x_3^2 - x_2) - 2(1 - x_3) + 8(x_3 - x_4^2) = 0$$
$$16x_4(x_4^2 - x_3) - 2(1 - x_4) + 8(x_4 - x_5^2) = 0$$
$$16x_5(x_5^2 - x_4) - 2(1 - x_5) + 8(x_5 - x_6^2) = 0$$
$$16x_6(x_6^2 - x_5) - 2(1 - x_6) + 8(x_6 - x_7^2) = 0$$
$$16x_7(x_7^2 - x_6) - 2(1 - x_7) + 8(x_7 - x_8^2) = 0$$
$$16x_8(x_8^2 - x_7) - 2(1 - x_8) = 0$$

The results of initial approximation for the root are shown in Fig. (5):

Table (5) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
2	1	7	-	-	1	6
2	1				1	
2	1				1	
2	1				1	
1	1				1	
1	1				1	
1	1				1	
1	1				1	
0.9	1	6	-	-	1	5
0.9	1				1	
0.9	1				1	
0.9	1				1	
0.9	1				1	
0.9	1				1	
0.9	1				1	
0.9	1				1	
1.1	1	6	-	-	1	5
1.1	1				1	
1.1	1				1	
1.1	1				1	
1.1	1				1	
1.1	1				1	
1.1	1				1	
1.1	1				1	

Example 6:

The following system of sixteen equations in sixteen variables is described below:

$$x_1 + x_7 x_{16} = 3$$
$$x_1 + 2x_2 + x_3 + x_{10} + x_{13} x_{16} = 10$$
$$x_3 = 1$$
$$x_1 + x_3 + 2x_4 + x_5 + x_6 + x_{12} x_{13} = 11$$
$$2x_1 + x_5 x_7 = 3$$
$$x_1 + x_6 + x_8 x_{14} = 7$$
$$x_7 x_8 = 2$$
$$x_7^2 + x_8 = 3$$
$$x_1 + 4x_9 + x_{11} + x_{12} x_{16} = 10$$
$$x_1^2 + 3x_5 + x_{10} x_{13} = 6$$
$$x_9 + x_{10} x_{11} = 3$$
$$x_6 + 2x_9 + x_{11} + x_{12} = 6$$
$$x_{10} x_{13} = 2$$
$$x_3 + x_8 x_{14} = 5$$
$$x_3 + 2x_4 + x_8 + 3x_{10} + x_{14} x_{15} = 15$$
$$x_2 + 2x_5 + x_{10} x_{16} = 8$$

The results of initial approximation for the root are shown in Fig. (6):

Table (6) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
2	0.9969	9			0.9969	7
2	2.0042				2.0042	
2	1.0000				1.0000	
2	5.4752				5.4752	
2	1.0021				1.0021	
2	2.0031				2.0031	
2	1.0042				1.0042	
2	1.9916				1.9916	
2	4.9676				4.9676	
2	2.0010				2.0010	
2	-0.9833				-0.9833	
2	-4.9550				-4.9550	
2	0.9995				0.9995	
2	2.0084				2.0084	
2	-2.4621				-2.4621	
2	1.9948				1.9948	
15	0.9969	10	-	-	0.9969	10
15	2.0042				2.0042	
15	1.0000				1.0000	
15	5.4752				5.4752	
15	1.0021				1.0021	
15	2.0031				2.0031	
15	1.0042				1.0042	
15	1.9916				1.9916	
15	4.9676				4.9676	
15	2.0010				2.0010	
15	-0.9833				-0.9833	
15	-4.9550				-4.9550	
15	0.9995				0.9995	
15	2.0084				2.0084	
15	-2.4621				-2.4621	
15	1.9948				1.9948	
20	0.9994	11	-	-	0.9954	8
20	2.0008				2.0062	
20	1.0000				1.0000	
20	5.4954				5.4638	
20	1.0004				1.0031	
20	2.0006				2.0046	
20	1.0008				1.0062	
20	1.9985				1.9877	
20	4.9940				4.9528	
20	2.0002				2.0015	
20	-0.9969				-0.9756	
20	-4.9917				-4.9345	
20	0.9999				0.9992	
20	2.0015				2.0123	
20	-2.4930				-2.4447	
20	1.9990				1.9923	

Example 7:

The following system of thirty-two equations in thirty-two variables is described below:

$$8(x_1 - x_2^2) = 0$$

$$16x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 8(x_j - x_{j+1}^2) = 0, \text{ for } j = 2, \ldots, n-1$$

$$16x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) = 0, \text{ for } n = 32$$

The results of initial approximation for the root are shown in Fig. (7):

Table (7) Results of initial approximation for the root.

Initial value	Newton		Broyden		Newton-Wavelet	
	Roots	Iteration	Roots	Iteration	Roots	Iteration
1.1	1.000	7	-	-	1.000	6
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	
1.1	1.000				1.000	

2	**1.000**	**7**			**1.000**	**7**
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
2	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	
1	**1.000**				**1.000**	

6. Conclusions

In this work, we use a new algorithm to solve system of non-linear equation, which is called by Newton-Wavelet method. Finally, we conclude that this algorithm gives a more satisfactory result than Broyden Method. Performance of this algorithm sometimes is better than Newton Method for some examples, in term number of iteration

REFERENCES

[1] Bond, D. M. and S. A. Vavasis, 1994, “Fast Wavelet Transforms for Matrices Arising from Boundary Element Methods”, Center for Applied Mathematics, Engineering and Theory Center, Cornell University, Ithaca, March 24.

[2] Gerald, C. F. and Wheatley, P. O., 1994, “Applied Numerical Analysis”, Fifth Edition, United States of America.

[3] Misiti, M., Misiti, Y., Oppenheim, G. and Poggi, J. M., 1997,” Wavelet Toolbox For Use With MATLAB”, User’s Guide, Version 1, MathWorks, Inc.

[4] Penny, J. and Lindfield, G., 1995,” Numerical Methods Using MATLAB”, Department of Mechanical Engineering, Aston University, Ellis Horwood Limited.

[5] Soesianto, F., and Abdullah, R. M., 2000, "Two Competing Methods to Solve Sparse Linear Equations", a paper presented to the International Seminar on Numerical Analysis in Engineering, University of North Sumatra Medan, March, 2000.

[6] Sweldens, W., 1994,” Construction and application of Wavelet in Numerical Analysis”, Ph.D. Thesis, Columbia.