



## The Performance of Two Mothers Wavelets in Function Approximation

Mohd Fazril Izhar Mohd Idris

Department of Mathematical Sciences and Statistic

Faculty of Computer Science and Mathematics

Universiti Teknologi MARA Perlis

02600 Arau, Perlis, Malaysia

Tel: 60-4987-4387 E-mail: [fazrilizhar@perlis.uitm.edu.my](mailto:fazrilizhar@perlis.uitm.edu.my)

Zaki Ahmad Dahlan

Department of Surveying and Geomatics

Faculty of Architecture Planning and Surveying

Universiti Teknologi MARA Perlis

02600 Arau, Perlis, Malaysia

Tel: 60-4986-4360 E-mail: [zaki@perlis.uitm.edu.my](mailto:zaki@perlis.uitm.edu.my)

Hj. Kamaruzaman Jusoff (Corresponding author)

Faculty of Forestry, Universiti Putra Malaysia

43400 Serdang, Selangor, Malaysia

Tel: 60-3987-4252 E-mail: [kjusoff@yahoo.com](mailto:kjusoff@yahoo.com)

### Abstract

Research into Wavelet Neural Networks was conducted on numerous occasions in the past. Based on previous research, it was noted that the Wavelet Neural Network could reliably be used for function approximation. The research conducted included comparisons between the mother functions of the Wavelet Neural Network namely the Mexican Hat, Gaussian Wavelet and Morlet Functions. The performances of these functions were estimated using the Normalised Square Root Mean Squared Error (NSRMSE) performance index. However, in this paper, the Root Mean Squared Error (RMSE) was used as the performance index. In previous research, two of the best mother wavelets for function approximations were determined to be the Gaussian Wavelet and Morlet functions. An in-depth investigation into the two functions was conducted in order to determine which of these two functions performed better under certain conditions. Simulations involving one-dimension and two-dimension were done using both functions. In this paper, we can make a specifically interpretation that Gaussian Wavelet can be used for approximating function for the function domain  $[-1, 1]$ . While Morlet function can be used for big domain. All simulations were done using Matlab V6.5.

**Keywords:** Neural network, Wavelet neural network, Function approximation, Morlet function, Gaussian wavelet function

### 1. Introduction

An Artificial Neural Network (ANN) is an information-processing paradigm that was inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN was configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between neurons and a similar process prevails in ANNs

Neural network simulations appear to be a recent development although this field was established before the advent of computers. Warren & Walter (1943) produce the first artificial neuron. However, the technology available at that time did not allow them to do too much. Many important advances have boosted by the use of inexpensive computer simulations. Despite undergoing a period of insufficient funding and support, important advances were made by a few researchers who developed technologies that overcome the limitations identified by Minsky & Papert (1969), who in their seminar work summed up a general feeling of frustration regarding the deficiencies of neural networks. Previous research, found that the Gaussian Wavelet, one of the transfer functions of the Wavelet Network, is the most suitable for other transfer function (Ramli, 2004). Other researchers support this conclusion. However, on their research, they employ the Normalised Square Root Mean Squared Error (NSRMSE) for calculating performance. The main purpose of this paper is to use Wavelet Network in function approximation and demonstrate that the performance depends on how good the chosen mother / activation / transfer function “fits” the function itself. This paper utilizes the Root Mean Squared Error (RMSE) as its performance index. We will find some different result and conclusion compared to the previous research. In order to evaluate the performance of transfer function, we used four different functions with two types of dimensions i.e., one dimension and two dimension.

## 2. Method

Wavelet Neural Networks (WNNs) are the outcomes derived from the implementation of Wavelet Decomposition, a technique that has recently emerged as a powerful tool for many applications in the field of signal processing, such as data compression and function approximation. The wavelet network is an approach for system identification in which nonlinear functions are approximate as the superposition of dilated and translated versions of a single function. A Wavelet network uses a wavelet like an activation function. Families of wavelet functions, especially wavelet frames, are universal approximators for the identification of nonlinear systems. The parameters of wavelet networks are dilation ( $t$ ), translation ( $\epsilon$ ), bias ( $\Theta$ ) and weight ( $\omega$ ). These parameters are optimizing during the learning phase. The basic idea of Wavelet decomposition is to expand a generic signal  $f(x) \in L^2(\mathfrak{R}^N)$  into a series of functions obtained by dilating and translating a single function  $\Phi(x)$ , the so-called “mother wavelet”. The term “mother wavelet” gets its name from two important properties of the wavelet analysis. The term wavelet means a small wave. The term mother implies that functions with different regions of support used in the transformation process. They derived from one main function, the mother wavelet. In other words, the mother wavelet is a prototype for generating the other window functions. The mother wavelet function gives an efficient and useful description of the signal of interest. This function has a universal property. In the following, we shall consider only radial wavelet in function space  $L^2(\mathfrak{R}^N)$ , for which  $\Phi(x) = g(\|x\|)$  where  $g: \mathfrak{R} \rightarrow \mathfrak{R}$ . Radial functions are characterized by a radial Fourier transform; a function is admissible as a wavelet if  $C_\Phi = (2\pi)^N \int_0^\infty \frac{|\hat{\Phi}(hw)|^2}{h} dh < \infty$  and  $C_\Phi$  is independent of  $\omega$ .

In other words for the Discrete Wavelet Transform, the parameters which determine the dilation and translation of the mother wavelet are discretised, namely a countable set is extracted, such that of the corresponding wavelet family. For instance,

$$\Phi_k = \left\{ \det \left| D_k^{\frac{1}{2}} \right| \Phi [D_k(x - t_k)] : t_k \in \mathfrak{R}^N, D_k = \text{diag}(d_k), d_k \in \mathfrak{R}_+^N, k \in \mathbb{Z} \right\} \quad (1)$$

is a basis for the function in  $L^2(\mathfrak{R}^N)$ . To attain this aim, an additional condition is required both on and on the parameters discretisation. The obtained basis is not necessarily orthonormal and can be somehow redundant. In this latter case, family (1) is more correctly referred to as a frame. Frames of wavelet have used extensively to approximate functions of one or two variables, but as the number of variables increases, the required number of basis functions grows exponentially. In practice, a signal  $g$  is approximated by the weighted sum of a finite number of functions in (1) plus a bias that help the approximation of functions with a non-zero mean value:

$$g(x) = \sum_{k=1}^K a_k \Phi [D_k(x - t_k)] + \Theta \quad (2)$$

This is analogous to the output of a two-layer network provided that the activation function of the hidden neurons is wavelets. Such a network named a Wavelet Network (WN). WN with radial wavelets presents the main advantage of an efficient initialization procedure derived from wavelet decomposition. Furthermore, a fast procedure based on the Orthogonal Least Squares (OLS) algorithm, a method already applied to radial basis function (RBF) networks, is provided for choosing among all the basis functions those that give the greatest contribution to the approximation. Depending on the form of the function to be approximate, the expansion of a signal into a wavelet series can be more efficient than other solutions, in the sense that fewer basis functions are needed for achieving a fixed approximation error. This is due to the time-frequency properties of most wavelets, which make them particularly suitable to represent short-time high-frequency signal features. Hence, fewer basis functions and more efficient initialization lead to smaller networks and fast training. On the other hand some signal features are better represented by the linear combination of different functions, thus WN

are not suitable to fit any curve. The success of RBF neural networks for function approximation was a good indicator of this yet another field of application on wavelets. A wavelet frame replaces the radial basis functions in a RBF network, the shifts and scales of the wavelets replace the center and covariance matrix (spread). The architecture of a wavelet network consists of three different layers: an input layer that is made of source nodes, a hidden layer in which each neuron computes its output using a wavelet basis function and an output layer that builds a linear weighted sum of the hidden layer. The transformation from input layer to the hidden layer is nonlinear while that from the hidden layer to the output layer is linear. The methodology for the implementation of Neural Networks using the Neural Network Toolbox (version 6.5) comprises the following steps. (1) Define the function to be approximated. (2) Give the domain of the function to be approximated. (3) State the sum squared error goal to Network Radial Basis (NEWRB) to get the best result. (4) Decide training and testing data. (5) Manipulate the data and target its generation. (6) Create neural networks using NEWRB and initialize. (7) Train and test the network. (8) Evaluate performance. These steps are used to design radial basis function networks in function approximation.

### 3. Results and discussions

#### 3.1 Program simulation

In this section, four functions are used to simulate the program. They comprise one-dimensional and two-dimensional functions in order to avoid biasness.

The functions used in the simulation for function approximation were from Jaya (2005). The Gaussian Wavelet Neural Network and the Morlet Neural Network, the two transfer functions of NEWRB that modified, were used.

#### 3.2 Modeling One Dimensional Nonlinear Functions

*Simulation One:* From (Jaya, 2005), the first example used was:

$$y = x^3 + 0.3x^2 - 0.4x; x \in [-1, 1] \quad (3)$$

The training set composes of 100 points, chosen uniformly by partitioning the domain [-1, 1] based on a grid measurement of 0.02. While the test set comprises 100 points randomly sampled from the same domain. The graph obtained illustrated in Figure 1.

An error goal of 0.01 for Gaussian Wavelet was used with the spread constant value remaining uninfluenced by any value. In this way, the error goal value and the optimal result were obtained by trial and error. After running the program by changing certain error goal value, the optimal results obtained are shown in Table 3.1 for simulation one. The graph showing the variations between the original and Gaussian Wavelet function is shown in Figure 2.

An error goal of 0.01 for the Morlet function with a spread value of 0.83 was used in simulation one. The value of the error goal and the spread constant value were manipulated until the best results were obtained using the trial and error technique. The results are shown in Table 3.1 while Figure 3 depicts the variations between the original and the Morlet function.

*Simulation Two:* According to (Jaya, 2005), the example used was:

$$y = \tan(x) + \cos(x); x \in [0, 10] \quad (4)$$

Figure 4 depicts the original function.

The training set consisted of 100 points, chosen uniformly by partitioning the domain [0, 10] based on a grid measurement of 0.1. While the test set comprises 100 points randomly sampled from the same domain. The error goal for Gaussian Wavelet was set at 0.01 and the results are shown in Table 3.1. Figure 5 illustrates the differences between the Gaussian and the original function.

An error goal of 0.02 was used for the Morlet function with a spread value of 2.273. By changing the error goal value via trial and error, the best result was obtained as shown in Table 3.1 for simulation two. The difference between the original and the Morlet function is depicted in Figure 6.

#### 3.3 Modeling Two Dimensional Nonlinear Functions

*Simulation Three:* Concern a two-dimensional example by (Jaya, 2005), a function described by

$$z = \sin(10x) \sin(10y); x, y \in [0, 1] \quad (5)$$

The training set composed of 101 points, which uniformly chosen by partitioning the domain [0, 1] based on a grid measurement of 0.01. While the test set comprises 101 points randomly sampled from the same domain. The surface obtained is depicted in Figure 7. An error goal value of 0.2 for the Gaussian Wavelet was selected after the trial and error.

After running the program, the results shown in Table 3.2 in simulation three was obtained. The surface for the Gaussian Wavelet function is shown in Figure 8. An error goal of 0.02 for the Morlet function with a spread value of 0.415 was used. By using the trial and error technique, the best results shown in Table 3.2 for simulation three were obtained. The surface for the Morlet function is shown in Figure 9.

*Simulation Four:* The next example is a function described by

$$z = \left(\frac{5}{2\pi}\right) e^{\left(-\frac{x^2+y^2}{2}\right)}; \quad x, y \in [-4, 4] \quad (6)$$

The training set consisted of 100 points, chosen uniformly by partitioning the domain  $[-4, 4]$  based on a grid size of 0.08. The test set comprises 100 points randomly sampled from the same domain. The surface determined by this function is shown in Figure 10.

We used an error goal value of 0.05 for the Gaussian Wavelet after the trial and error process. After running the program, we obtained the results as shown in Table 3.2 for simulation four. Figure 11 depicts the Gaussian Wavelet function's surface. An error goal of 0.02 for was used for the Morlet function with a spread value of 3.33. The value of error goal and spread constant values are manipulated using the trial and error technique to obtain the results as shown in Table 3.2 for simulation four. The surface for the Morlet function is shown in Figure 12.

Network radial basis (NEWRB) was implemented to approximate all the functions. Next, the Root Mean Squared Error (RMSE) was used to determine the accuracy of the function approximate. After running the program, a comparison was made between the two-transfer function / mother of wavelet namely, the Gaussian Wavelet and the Morlet function. From the tables, only the first simulation showed that the Gaussian Wavelet performed better than the Morlet function. While the other revealed that, the Morlet function performed better.

In these simulation problems, several domain functions were used in order to obtain feedback on the efficacy of the two transfer functions. Next, an analysis of the performance of each function in the respective simulation was done. In simulation one, the two transfer functions appeared to generate the exact design. However, this was not the case because there were still some obvious errors when we estimated the performance of these networks using RMSE. As for the remaining four simulations, the Morlet function generated superior results, as the designs produced were almost similar to that generated by the original function.

Mallat (1998), notes that this superior performance is due to the Morlet function possessing compact support, which means that it vanishes outside a finite interval. This compact support property is absent in the Gaussian Wavelet function. Functions with compact support in  $x$  are those with support that is a compact subset of  $x$ . For example, if  $x$  is the real line, they are function of bounded support and therefore vanish at infinity (and negative infinity). This will cause the function to be approximate with far or big function domains. In other words, functions with compact support are suitable to be used as approximate functions in big function domains (Mallat, 1998).

Therefore, from this property it gives us the reason to the result we get. On the other hand, we can make a specific interpretation that Gaussian Wavelet can be used for approximating function for the function domain  $[-1, 1]$  while Morlet function can be used for a big domain. This interpretation is derived from the result we get.

#### 4. Conclusion

The two mother wavelet functions selected for function approximation were the Gaussian Wavelet and Morlet functions. The RMSE was used as the performance of index for function approximation in this research in contrast to the use of the NSRMSE as the performance index by other researchers. The main objective of this study was to investigate the capabilities of the two mother wavelets and its superiority. After simulating several functions onto these transfer functions, several variations from previous research observed especially when domain functions for function approximation were used. One reason for the variations, was that in order to obtain the best approximation, the spread and error goal values for the Morlet function were obtained after a trial and error procedure while this was not the case for the Gaussian wavelet which only involved changing the values of the error goal.

In contrast, the superior performance of the Morlet function is mainly attributable to the fact that two values can be altered. The other reason concerns the compact support property. Morlet function is a compact support that vanished outside of finite interval while the Gaussian Wavelet does not. This is a concrete reason why the function with far function domain is suitably approximated by the function with this compact support property.

Then from the property, we can make a specific interpretation that Gaussian Wavelet can be used for approximating function for the function domain  $[-1, 1]$  while Morlet function can be used for the big domain. This interpretation is in accordance to the result we get in simulation problem part.

#### References

Bosman, S. (1996). *Locally weighted approximations : Yet Another Type of Neural Network*. Intelligent Autonomous

Systems Group. University of Amsterdam.

Demuth, H. & Beale, M. (2000). *Neural Network Toolbox User's Guide*. (6th ed.). New York: Mathworks Inc.

Echaz, J. (1998). Strategies for Fast Training of Wavelet Neural Networks. *2nd International Symposium on Soft Computing for Industry, 3rd World Automation Congress (WAC'98)*, 1–6, Anchorage, Alaska.

Jaya K. Y. (2005). *Neuro-Fuzzy System*. Master Thesis, USM.

Katic, D., Vukobratovic, M. (1997). Wavelet Neural Networks Approach for Control of Non-Contact and Contact Robotic Tasks. *Proceedings IEEE Symposium of Intelligent Control*. 245–250. Istanbul.

Mallat, S. G. (Stephane G.) (1998), *A Wavelet Tour of Signal Processing*, San Diego, Academic Press. Matlab 6.5 Release 14, *Neural Network Toolbox*, The MathWork.

McCulloch, W. S. & W. Pitts. (1943). *A logical calculus of the ideas immanent in nervous activity*. Bull. math. Biophys. 5, 115-133.

Minsky, M. & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Morozov, V.A. (1993). *Regularization Methods for Ill-Posed Problems*. Boca Raton, Florida: CRC Press.

Ramli R. (2004). *Function Approximation Using Wavelet and Radial Basis Function Networks*. Master Thesis, USM.

Robert J. Schalkoff (1997). *Artificial Neural Network*. The McGraw-Hill Companies, Inc.

Rong Gao & Tsoukalas L.H. (2001). Neural-Wavelet Methodology for Load Forecasting. *Journal of Intelligent and Robotic Systems*. 31, 149–157.

Sgarbi, M., Colla, V., Reyneri, L. M. (1998). A Comparison Between The Weighted Radial Basis Functions Networks and Wavelet Networks. *Proceedings ESANN'98*. 13–19. Bruges, Belgium.

The MathWork, Matlab 6.5 Release 14, *Neural Network Toolbox*.

Thomas, J.H., Dubuisson, B., Dillies-Pelter, M.A. (1996). Engine knock detection from vibration signals Using Pattern Recognition. *Proceeding of Neural Network*. 32, 431–439. Meccanica.

Zarita Zainuddin, & Ong Pauline (2008). Function Approximation Using Artificial Neural Networks, *WSEAS Transactions on Mathematics*, 7 (6), 333–338.

Zarita Zainuddin, & Ong Pauline (2007). Function Approximation Using Artificial Neural Networks, *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*, 140–145, Cairo, Egypt

Zhang, Q. (1997). Using Wavelet Networks in Non-Parametric Estimation. *IEEE Transaction on Neural Networks*. 8 (2), 227–236.

Table 1. The results for the One Dimensional Model

SIMULATION	GAUSSIAN WAVELET	MORLET
1 Training	1.0975e-022	0.00000745
Testing	0.00000013	0.00010466
2 Training	0.00942726	1.7631e-010
Testing	0.06337701	0.00000018

Error: RMSE

Table 2. The results for the Two Dimensional Model

SIMULATION	GAUSSIAN WAVELET	MORLET
3 Training	0.00299518	0.00006062
Testing	0.00166752	0.00032344
4 Training	0.09590531	0.00000142
Testing	0.18258935	0.00007684

Error: RMSE

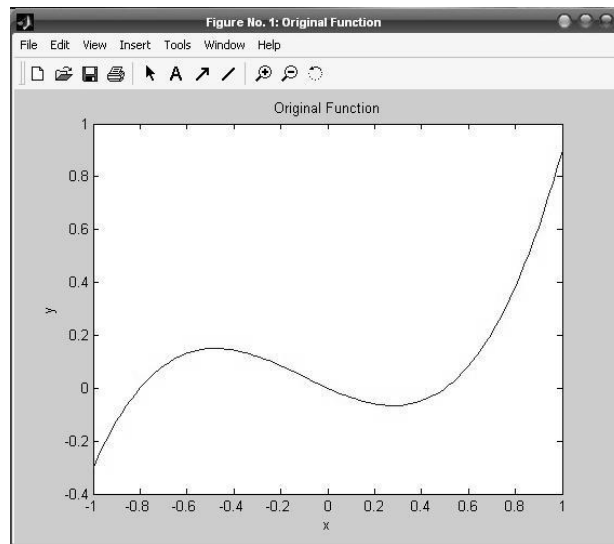


Figure 1. The original function for simulation one

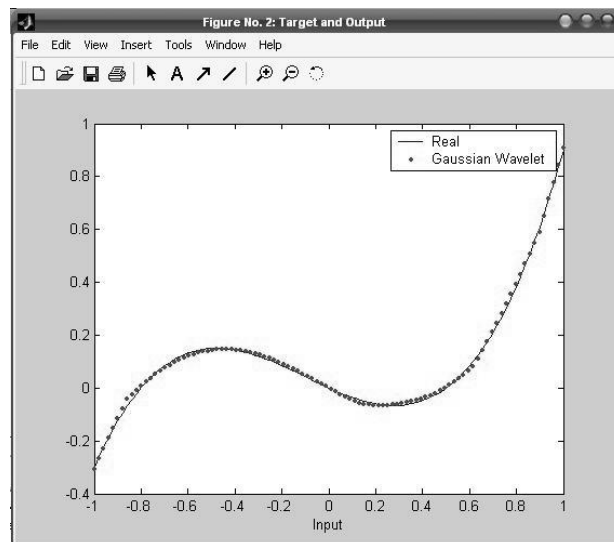


Figure 2. The difference between the Gaussian Wavelet function and the original function

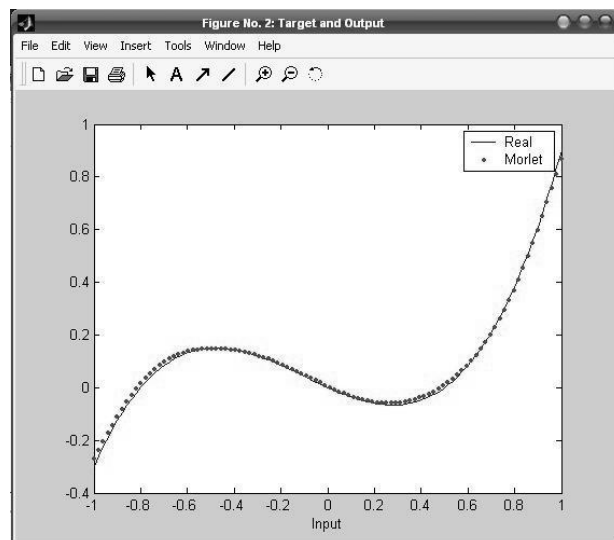


Figure 3. The difference between the Morlet function and the original function

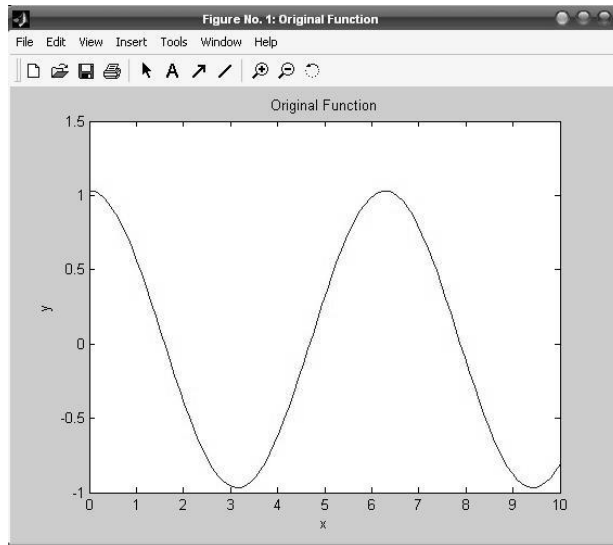


Figure 4. The original function for simulation two

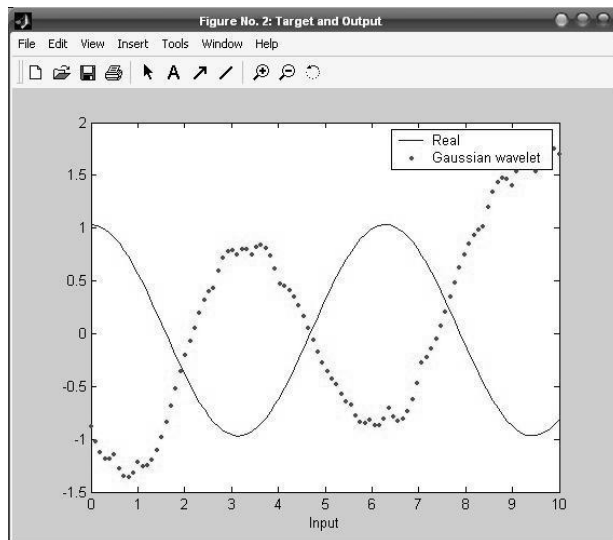


Figure 5. The difference between the Gaussian Wavelet function and the original function

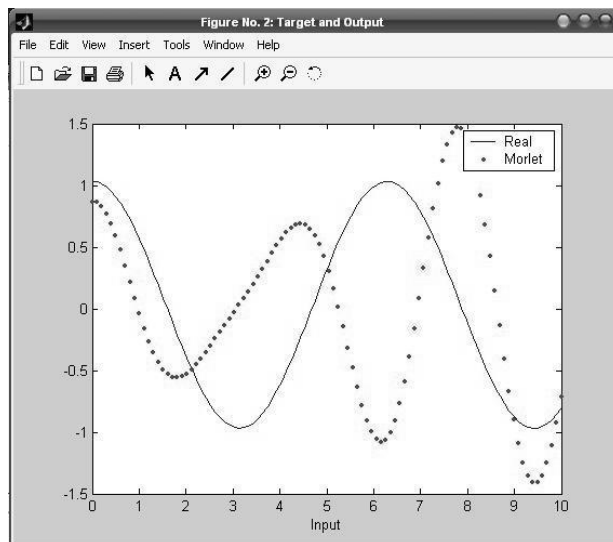


Figure 6. The difference between the Morlet function and the original function

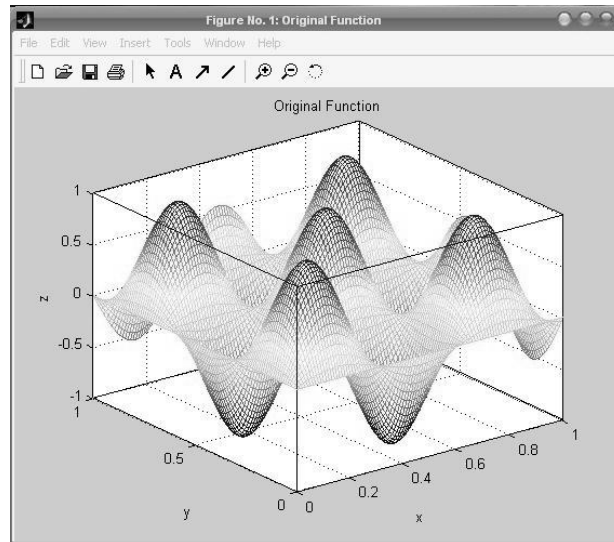


Figure 7. The original function surface for simulation four

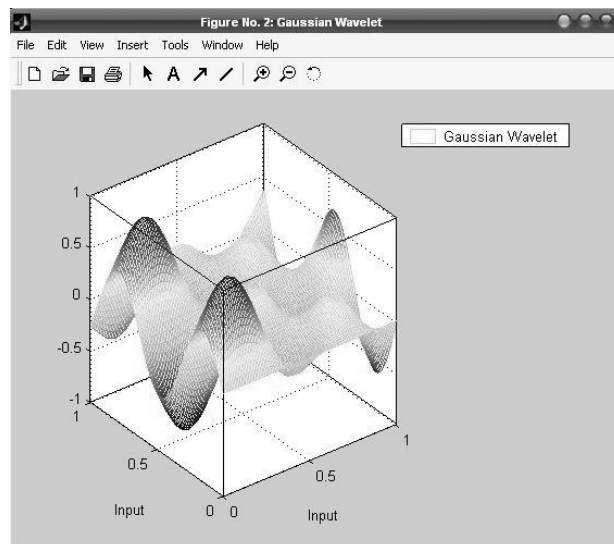


Figure 8. The Gaussian Wavelet surface

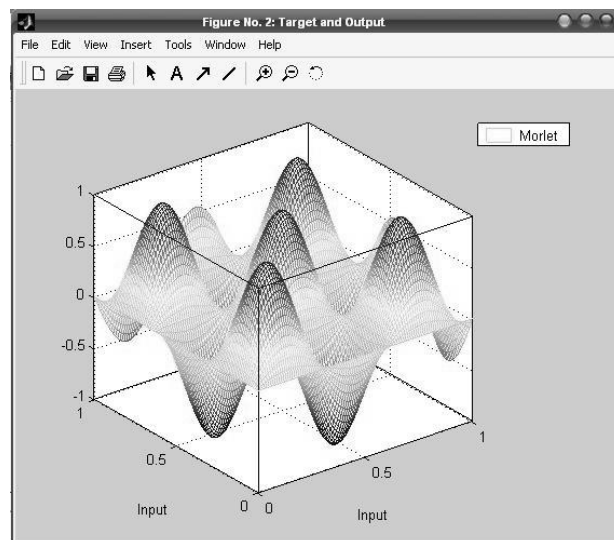


Figure 9. The Morlet function surface



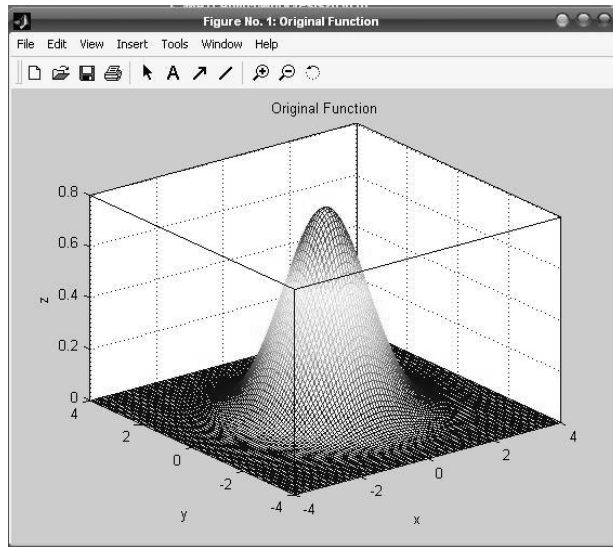


Figure 10. The original function surface for simulation five

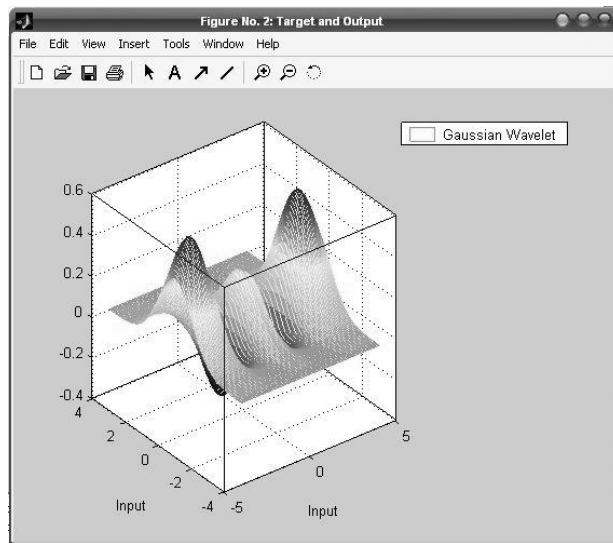


Figure 11. The Gaussian Wavelet surface

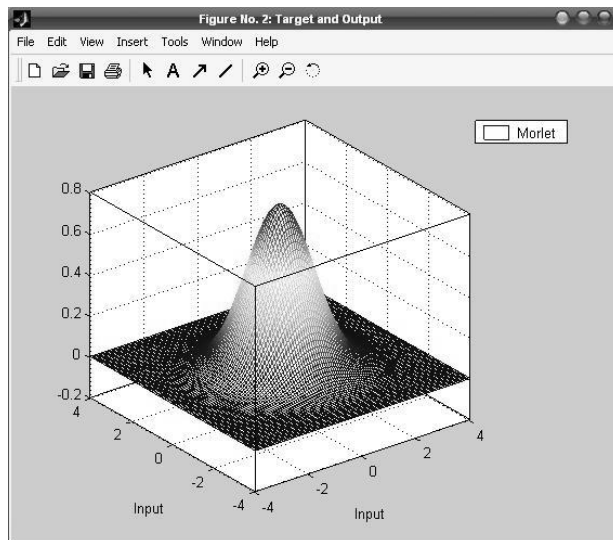


Figure 12. The Morlet function surface