ASME Early Career Technical Journal
2012 ASME Early Career Technical Conference, ASME ECTC
November 2 – 3, Atlanta, Georgia USA

COMPARISON OF THE LENGTH FACTOR ARTIFICIAL NEURAL NETWORK AND FINITE ELEMENT METHODS FOR SOLVING BOUNDARY VALUE PROBLEMS

Kevin McFall
Southern Polytechnic State University
Marietta, GA, USA

Patrick McEnroe
The Pennsylvania State University
State College, PA, USA

ABSTRACT

The length factor artificial neural network method (LFANNM) has been demonstrated as an alternative to the finite element method for solving boundary values problems (BVPs). Besides optimizing on a rectangular grid rather than the typical triangular mesh of the FEM, a primary advantage of the LFANNM is that the resulting continuous approximate solution eliminates the need for interpolation between grid points. This manuscript compares the accuracy of the LFANNM and FEM for BVPs with known analytical solutions, both for single partial differential equations (PDEs) and for coupled systems of PDEs. Unlike the FEM, results show that error for the LFANNM on and between grid points is roughly equivalent. This superior built-in interpolation allows the LFANNM to outperform the FEM in most cases even when trained with as much as an order of magnitude fewer nodal points.

INTRODUCTION

Problems in science and engineering often involve differential equations (DEs) sufficiently complicated to require numerical techniques for approximating their solutions. Traditionally, the finite difference [1], finite element [2], and boundary element [3] methods have been employed to numerically solve DEs. Although powerful and widespread, these tools do have drawbacks including problematic discretization of the problem domain, the need for interpolation between nodes, and complications in solving nonlinear DEs. Artificial neural networks (ANNs) have emerged as an alternative method for numerical solution of DEs, with a survey [4] citing roughly 50 different publications employing ANNs to solve DEs in any number of application areas. Solving boundary value problems (BVPs) with ANNs requires optimizing an approximate solution to satisfy the DE in question under the constraint imposed by the boundary conditions (BCs). The length factor artificial neural network method (LFANNM) [5], [6] removes the BC constraint by posing the approximate solution in a form where the BCs are automatically satisfied regardless of ANN output.

Typically, researchers justify the value of ANN methods by showing they rival or even outperform traditional methods. The comparison is often done for a single combination of design parameters [7] preferentially chosen for high performance. While this does provide an understanding for the best possible ANN approximation, it does not necessarily reflect typical performance nor provide insight into how to properly choose design parameters. When solving BVPs with the LFANNM, the practitioner must select the number and location of training points, the number of hidden neurons in the ANN, and the starting values for ANN parameters. This manuscript aims to compare the LFANNM against the FEM when solving three different BVPs of varying complexity, and explore the dependence of performance against the FEM on all three design parameters.

THE LENGTH FACTOR METHOD

The LFANNM reduces the BVP to an unconstrained optimization problem by posing the trial approximate solution as

$$\psi_t(\mathbf{x},\boldsymbol{\theta}) = A(\mathbf{x}) + L(\mathbf{x})N(\mathbf{x},\boldsymbol{\theta}) \tag{1}$$

where $\mathbf{x}$ is the position vector in the domain, $\boldsymbol{\theta}$ is the ANN parameter vector including weights and biases, A is a problem specific ansatz designed to satisfy the BVP's Dirichlet BCs, L is the length factor, and N is the ANN output. The ansatz A can be a direct analytical expression known to satisfy the BCs or can be defined using thin plate spline (TPS) interpolation [8] on domains of arbitrary shape [5]. The length factor L can also be defined by inspection or with TPS interpolation. In either case, the length factor must have a zero value on the domain boundary and non-zero values inside the domain. As such, Equation (1) reduces to the exact BCs for all $\mathbf{x}$ on the boundary irrespective of N since the contribution due to the ANN vanishes when multiplied by the zero length factor, leaving only the correct BC value encoded in A.

Two-dimensional TPS interpolation takes the form

$$f(x,y) = \sum_{i=1}^{m} F_i r_i^2 \ln r_i^2 + F_{m+1} + F_{m+2}x + F_{m+3}y \tag{2}$$

The quantities

$$r_i(x,y) = \sqrt{(x-\bar{x}_i)^2 + (y-\bar{y}_i)^2 + d^2} \qquad (3)$$

represent a measure of distance for a given point (x,y) from the i^{th} control point $(\bar{x}_i, \bar{y}_i)$, where d is a small number chosen to be 0.1 for all calculations here. The TPS is analogous to a point force F located at each of the m control points. The net force on the plate should be zero, as should the net moment about the x and y axes. These three conditions on the m+3 coefficients F are combined with the specification that $f(x,y)$ be set to known values at each of the at m control points. The resulting $(m+3)\times(m+3)$ linear system is then solved to determine the values for F.

Control points for generating A are chosen on the domain boundary and the value of $f(x,y)$ at each is specified according to the Dirichlet BCs. When using TPS interpolation to define the length factor, control points are also chosen on the domain boundary with values for $f(x,y)$ set to zero there. A final control point is selected inside the domain with $f(x,y)$ set to unity to avoid the interpolation from producing the trivial solution of $f(x,y)=0$.

The ANN architecture is a feed-forward, fully-connected, multi-layer perceptron network with the position vector components as inputs, H neurons in the hidden layer employing logistic sigmoid transfer functions, and a single output N computed with a linear transfer function. The ANN parameters, including weights and biases, are initialized to small random values and then optimized using the Levenberg-Marquardt [9] gradient descent method to minimize the error function

$$E = \sum_{i=1}^{n} G\left(\mathbf{x}_i, \psi_t(\mathbf{x}_i), \nabla\psi_t(\mathbf{x}_i), \nabla^2\psi_t(\mathbf{x}_i)\right)^2 \qquad (4)$$

where G is the differential operator defining the desired differential equation

$$G\left(\mathbf{x}, \psi(\mathbf{x}), \nabla\psi(\mathbf{x}), \nabla^2\psi(\mathbf{x})\right) = 0 \qquad (5)$$

for the exact analytical solution ψ. The error in Equation (4) is evaluated at the collocation of n training points chosen on a square grid falling within the domain of the problem.

Systems of DEs are solved in the LFANNM [6] by defining an approximate solution of the form of Equation (1) for each unknown function in the system with its own ANN. The total error function in comprised of the sum of contributions in Equation (4) for each DE to be satisfied. This total error is a function of the ANN parameters for each of the approximate solutions, and is minimized similarly using the same Levenberg-Marquardt training algorithm.

Results are dependent on the number of training points n, the number of hidden neurons H, and the starting ANN parameters.

EXPERIMENTAL DESIGN

Results are evaluated for values of H = 20, 30, 40, and 50 and training points chosen on 10×10, 20×20, 30×30, 40×40, and 50×50 square grids. The twenty combinations of values for H and n are then trained with 100 different random starting values for ANN parameters. The ANN results are then compared with the Galerkin FEM [2] using a meshing algorithm [10] designed to generate triangular elements of roughly equal area. Performance of both the LFANNM and the FEM as a function of the number of training points is compared. Comparing performance with n for the LFANNM and the same number of mesh knots for the FEM is justified since both methods involve solving linear systems of the same size. Figure 1 illustrates the FEM mesh with 99 knots for a square domain as well as the points used to train the ANN with n = 100.

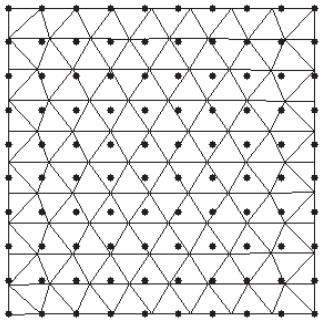

Figure 1: Training mesh with 99 knots for the FEM on a square domain, with comparable n = 100 ANN training points.

The average ANN performance over the 100 runs is reported in order to provide an indication of the typical error expected from the LFANNM. Box plots [11] are also generated, indicating the spread of performance over the 100 runs as well as the maximum and minimum possible errors with the LFANNM.

The FEM produces discrete approximations at the mesh knots, which are linearly interpolated to produce approximations between knots. Error in the FEM is therefore expected to be higher for points between knots. Two types of error are computed: training error measured at the mesh knots, and evaluation error measured at all points falling on a 75×75 square grid. Interpolation is unnecessary in the LFANNM since the approximate solution in Equation (1) is continuous over the entire domain. As a result, training and evaluation errors are not significantly different, with evaluation error typically less than 1.1 times training error. For this reason error is only reported for the LFANNM on the 75×75 evaluation grid since error at the training points is only slightly better.

PROBLEM 1: SQUARE DOMAIN LAPLACE EQUATION

The first BVP explored is the Laplace equation

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=0 \tag{6}$$

on a square domain with boundary conditions

$$\psi(0,y)=\psi(1,y)=\psi(x,0)=0 \text{ and } \psi(x,1)=\sin(\pi x) \tag{7}$$

which has the analytical solution

$$\psi=\frac{\sin(\pi x)\sinh(\pi y)}{\sinh(\pi)} \tag{8}$$

The ansatz

$$A=y\sin(\pi x) \tag{9}$$

satisfies the BCs in Equation (7) and the length factor

$$L=xy(x-1)(y-1) \tag{10}$$

is zero on the boundary and positive inside the domain.

Figure 2 illustrates the mean absolute error over the domain for the FEM and LFANNM. Error with the FEM is between 2 and 10 times worse on the evaluation grid than at mesh knots, something that could be improved with interpolation more sophisticated [12] than the linear method used. The LFANNM still outperforms the FEM by an order of magnitude compared with the best case error at the FEM knots. Note that LFANNM performance is not strongly dependent on the number of hidden neurons although slightly worse with H = 20.

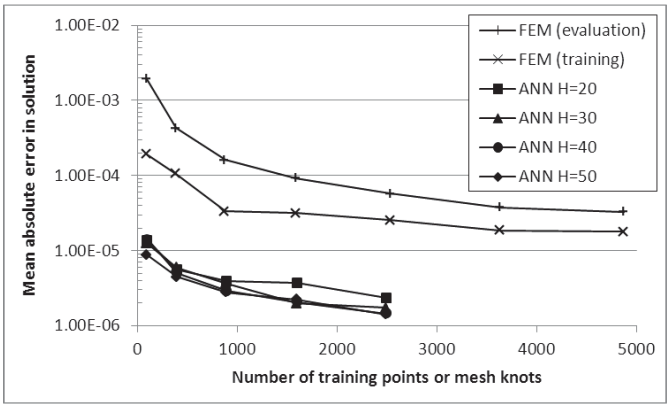

Figure 2: Error in approximating Problem 1 as a function of training points or mesh knots using the FEM as well as the average error over 100 runs with the LFANNM using various numbers of hidden nodes.

PROBLEM 2: IRREGULAR DOMAIN LAPLACE EQUATION

The second BVP considered is also the Laplace equation but on a domain defined by two non-concentric circles. The domain, including a sample of mesh knots and ANN training points, appears in Figure 3. The analytical solution

$$\psi=\frac{1}{2\ln 2}\ln\left(\frac{16r^2+1+8r\cos\theta}{r^2+16+8r\cos\theta}\right) \tag{11}$$

defined in polar coordinates satisfies Equation (6) with BCs of unity value on the outer circle defined by

$$(x-1)^2+y^2=2.5^2 \tag{12}$$

and zero value on the inner circle defined by

$$x^2+y^2=1^2 \tag{13}$$

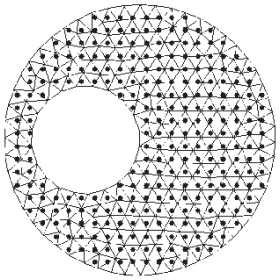

Figure 3: Training mesh with 262 knots for the FEM on the domain of Problem 2, with comparable n = 276 ANN training points.

The ansatz A in Figure 4 used for this problem was developed using TPS interpolation [8] with control points as shown. The length factor

$$L=\left[2.5^2-(x-1)^2-y^2\right]\left(x^2+y^2-1^2\right) \tag{14}$$

was determined by inspection in order to be zero on both the inside and outside circles with positive values in between.

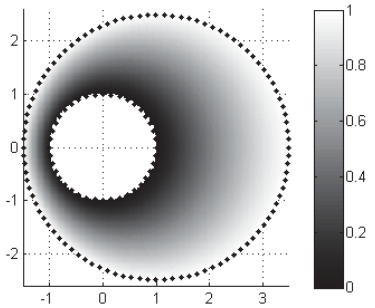

Figure 4: Ansatz A developed using TPS interpolation at the marked control points, and used to satisfy the Problem 2 BCs of zero on the inside circle and unity on the outside circle.

Figure 5 presents the error in both methods for Problem 2. Performance with the LFANNM is not overwhelmingly superior to the FEM, although error on the evaluation grid is better than with the FEM for less than 1000 training points when using at least 40 hidden neurons. This domain shape is challenging for the ANN approach since choosing training points on a square grid is not conducive to evenly spaced points in the small space between the circles, as is apparent in Figure 3. Interestingly enough, increasing the number of training points beyond 200 apparently does not increase performance with the LFANNM. While not categorically better than the FEM, the LFANNM with sufficiently many hidden neurons and training points produces acceptably low errors on the order of 10^{-4}; results are comparable to evaluation error when using the FEM with similar numbers of mesh knots.

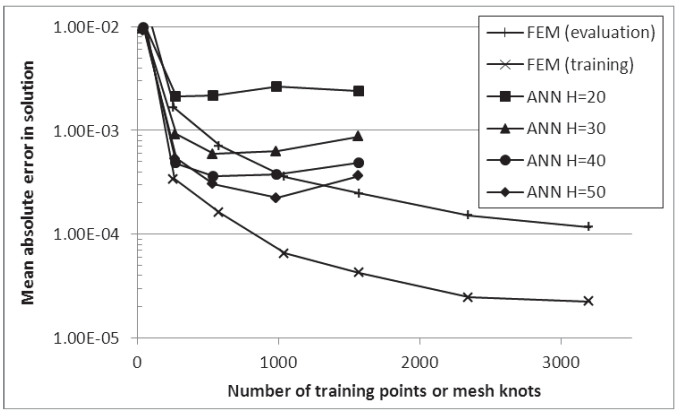

Figure 5: Error in approximating Problem 2 as a function of training points or mesh knots using the FEM as well as the average error over 100 runs with the LFANNM using various numbers of hidden nodes.

PROBLEM 3: SYSTEM OF TWO COUPLED DES

The final BVP investigated involves the coupled differential equations

$$\begin{aligned} \frac{\partial^2 \psi_1}{\partial x^2} + \frac{\partial \psi_1}{\partial y} + \frac{\partial \psi_2}{\partial x} + \frac{\partial \psi_2}{\partial y} &= 0 \\ \frac{\partial \psi_1}{\partial x} + \frac{\partial^2 \psi_2}{\partial y^2} + B_1 e^{A_1 x} + B_2 e^{A_2 y} &= 0 \end{aligned} \tag{15}$$

with analytical solutions

$$\begin{aligned} \psi_1 &= -\frac{B_1}{A_1} e^{A_1 x} + \frac{B_2}{A_2^2} e^{A_2 y} \\ \psi_2 &= B_1 e^{A_1 x} - \frac{B_2}{A_2} e^{A_2 y} \end{aligned} \tag{16}$$

where $A_1 = 0.85$, $A_2 = 0.2$, $B_1 = B_2 = 0.3$ in the case reported here. BCs are determined by evaluating (16) on the limaçon

$$r = 1.5^2 + 1.5\cos\theta \tag{17}$$

TPS interpolation was employed to produce A_1 and A_2 appearing in Figures 6a and 6b, respectively, designed to satisfy Equation (16) on the domain boundary defined by Equation (17). Figure 7 illustrates the length factor used, also developed by TPS interpolation.

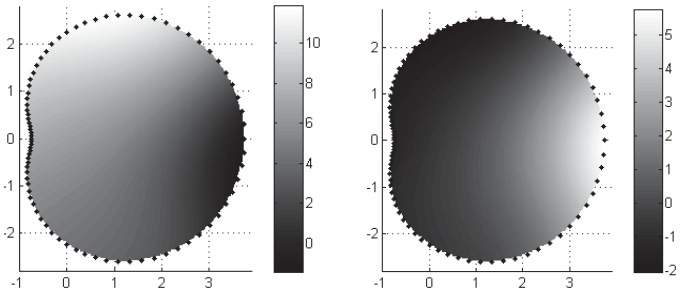

Figure 6: TPS interpolated ansatz functions to satisfy the BCs for a) ψ_1 and b) ψ_2 in Problem 3. TPS control points are marked with dots.

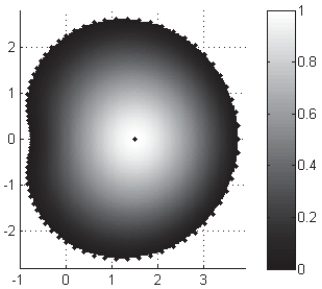

Figure 7: TPS interpolated length factor for Problem 3 with values at boundary control points set to zero and the value at the internal control point set to unity. TPS control points are marked with dots.

Figures 8 and 9 display error in Problem 3 when approximating the solutions of ψ_1 and ψ_2, respectively. With sufficiently many hidden neurons, the average LFANNM run outperforms the FEM for both ψ_1 and ψ_2, even for error evaluated at the mesh knots.

RESULTS

Figures 2, 5, 8, and 9 indicate that increasing the number of training points beyond a minimum value does not significantly improve performance with the LFANNM. The indication is that training points selected on a 30×30 rectangular grid, at roughly 250 to 300 points depending on the geometry, is sufficient to solve two-dimensional BVPs of varying complexity. The superior built-in interpolation achieved by the continuous approximate solution in Equation (1) eliminates the need to use excessively many training points. On the other hand, performance of the FEM continues to improve until several thousand mesh knots are included. The LFANNM has been shown to achieve similar or better performance with an order of

magnitude fewer training points. This translates to a sizable savings in computational complexity since factorization of linear systems grows at $O(n^3)$.

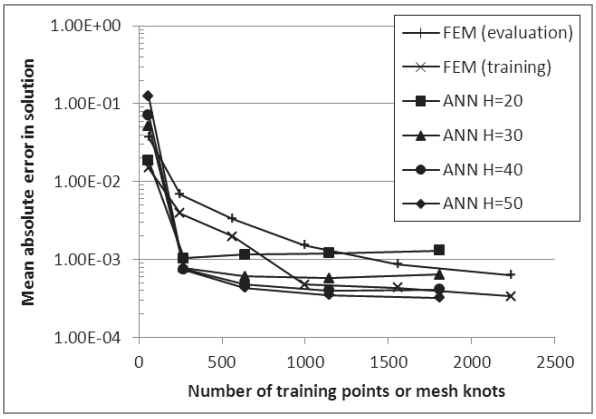

Figure 8: Error in approximating ψ_1 in Problem 3 as a function of training points or mesh knots using the FEM as well as the average error over 100 runs with the LFANNM using various numbers of hidden nodes.

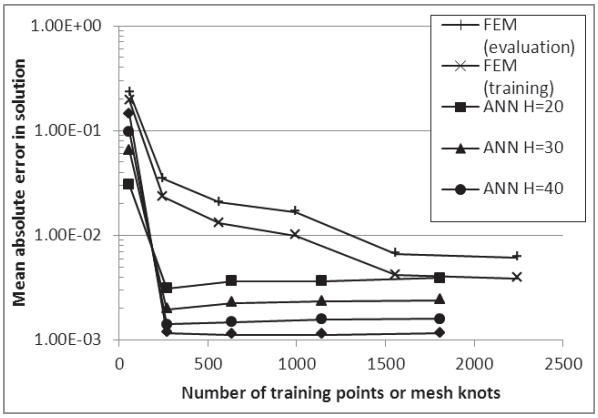

Figure 9: Error in approximating ψ_2 in Problem 3 as a function of training points or mesh knots using the FEM as well as the average error over 100 runs with the LFANNM using various numbers of hidden nodes.

Like the number of training points, results show that a minimum number of hidden neurons, H = 40, is necessary to guarantee acceptable performance. Experience has indicated that training points selected on a 30×30 rectangular grid and 40 hidden neurons are the minimum required design parameters. Using smaller values risks inferior performance, and using more unnecessarily increases training time.

Initial ANN weight and bias values are the final parameter affecting performance. Figures 2, 5, 8, and 9 present error averaged over 100 runs with different random starting weights. As such, those errors represent performance for a typical run when any given run could produce higher or lower error depending on the location of local minima in the error space. Figure 10 displays box plots for the 100 runs using minimum design parameters for H and n in each of the three problems explored. Outlier runs are marked with a + and error at the mesh knots for the FEM are included as a comparison. Open circles represent FEM error with a number of mesh knots comparable to the n used in the LFANNM, and filled circles indicate FEM error with the maximum number of mesh knots evaluated. These circles represent the lowest possible FEM error assuming an interpolation scheme could reduce error between mesh knots to the same level as at the knots.

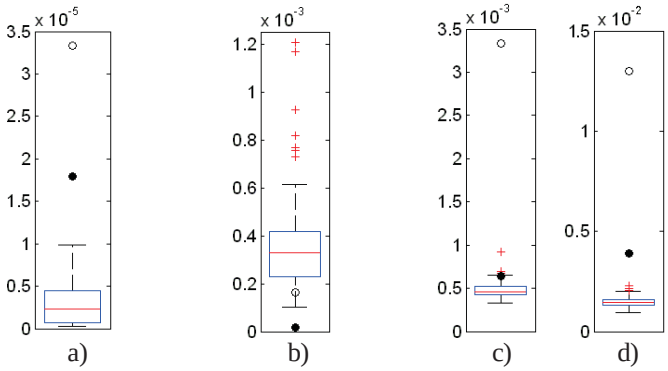

Figure 10: Box plots representing the mean absolute error in the solution for the LFANNM over 100 runs using the minimum design parameters. Performance includes a) Problem 1, b) Problem 2, and Problem 3 solutions c) ψ_1 and d) ψ_2. FEM error at mesh knots for a similar number of training points is marked with an open circle and FEM error at mesh knots with the maximum number of training points marked with a filled circle.

The worst runs for Problems 1 and 3 (excepting a couple outliers) perform better than the best possible FEM, even with a nearly unlimited number of mesh knots. LFANNM performance in Problem 2 is typically somewhat worse than the best case FEM, primarily due to selecting training points on a square grid for the circular domain shape. However, 6% of the LFANNM runs do result in lower error than the FEM with a similar number of mesh knots.

CONCLUSION

In general, the LFANNM is shown to significantly outperform the FEM, even when using as many as an order of magnitude fewer training points than mesh knots. The LFANNM is able to more accurately model the analytical solution to BVPs using relatively few training points due to the built-in interpolation from a continuous rather than discrete approximate solution. As long as sufficiently many training points are included (two-dimensional problems require points chosen on at least a 30×30 square grid) the continuous nature of the approximate solution successfully models the solution in the intervening spaces between training points. In fact, error between training points is nearly identical to that at the training points. Indeed, ANNs are known to be universal approximators [13], which can model any function to arbitrary levels of accuracy. Using more than a 30×30 grid of training points does not improve error much, in contrast with the FEM which

benefits significantly from increased numbers of mesh knots and the accompanying increased computational complexity.

Like the number of training points, results indicate a lower limit on the number of hidden neurons at about 40. Increasing this number can improve performance somewhat, but all problems were solved with acceptably low errors using no more than 40 hidden neurons.

One of the strongest criticisms of ANN methods is the dependence of performance on the choice of initial network weights. Even the worst runs of the LFANNM outperformed the FEM for two of the three problems explored here. FEM approximation was somewhat better in the other problem, presumably because the triangular mesh more accurately represented the circular domain shape than points chosen on a square grid.

ANN methods for solving BVPs have become more popular in recent years due to the simplicity of choosing training points, their continuous approximate solutions, and their ease of use with complex nonlinear DEs. This manuscript exhibits that this list of benefits should also include superior approximation in general compared with the more traditional FEM approach. Additionally, the increased performance is achieved with a significantly lower computational complexity due to requiring far fewer training points than mesh knots.

While the example problems explored in this manuscript represent relatively simple situations, the LFANNM has been shown to accurately solve problems in three dimensions [5] and more realistic thermal-fluids problems such as the Blasius boundary layer and the entrance length problem [6]. Even the apparently simplistic Laplace equation on a domain of non-concentric circles in Problem 2 could represent heat dissipation in a the insulation of a current carrying conductor where the solution produces the temperature distribution rather than simply a shape factor for computing overall heat transfer [14]. Comparison for this study was limited to linear DEQs easily solved with the FEM. Future studies will look to confirm similar performance of the LFANNM with more complicated DEQs modeling more realistic phenomena.

REFERENCES

[1] G. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford, UK: Clarendon, 1978.

[2] G. R. Liu and S. S. Quek, *The Finite Element Method: A Practical Course*. Boston: Butterworth-Heinemann, 2003.

[3] C. Brebbia, J. Telles, and L. Wrobel, *Boundary Element Techniques: Theory and Applications in Engineering*. Berlin: Springer-Verlag, 1984.

[4] M. Kumar and N. Yadav, "Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey," *Computers & Mathematics with Applications*, vol. 62, no. 10, pp. 3796–3811, Nov. 2011.

[5] K. McFall and J. Mahan, "Artificial Neural Network Method for Solution of Boundary Value Problems With Exact Satisfaction of Arbitrary Boundary Conditions," *Neural Networks, IEEE Transactions on*, vol. 20, no. 8, pp. 1221–1233, 2009.

[6] K. McFall, "Solving Coupled Systems of Differential Equations Using the Length Factor Artificial Neural Network Method," *American Society of Mechanical Engineers Early Career Technical Journal*, vol. 9, pp. 27–34, 2010.

[7] G. P. Zhang, "Avoiding Pitfalls in Neural Network Research," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 1, pp. 3–16, 2007.

[8] L. Zagorchev and A. Goshtasby, "A comparative study of transformation functions for nonrigid image registration," *Image Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 529–538, 2006.

[9] K. L. Priddy and P. E. Keller, *Artificial neural networks: an introduction*. SPIE Press, 2005.

[10] Per-Olof Persson and G. Strang, "A Simple Mesh Generator in MATLAB," *SIAM Review*, vol. 46, no. 2, pp. 329–345, Jun. 2004.

[11] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of Box Plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, Feb. 1978.

[12] H. Späth, *Two Dimensional Spline Interpolation Algorithms*. A K Peters/CRC Press, 1993.

[13] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *NEURAL NETW*, vol. 2, no. 5, pp. 359–366, 1989.

[14] Incropera, et. al. *Fundamentals of Heat and Mass Transfer*, Hoboken, NJ: Wiley, 2007