An Adaptive Meshless Method for Magnetic Field Computation

Qiang Li and Kok-Meng Lee

George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405

Design of electromagnetic (EM) actuators often involves solving a magnetic field problem. This paper presents an adaptive meshless method (MLM) that inherits many advantages of the finite-element method (FEM) but needs no explicit mesh structure for design of EM actuators. Specifically, the paper offers a technique to estimate the distribution of numerical errors and a scheme that automatically inserts additional nodes to improve computational accuracy and efficiency. It gives several examples. The first three numerical examples, where exact solutions are available, provide a means to validate the adaptive MLM and evaluate its effectiveness against a regular MLM with a uniform node distribution. The other examples, where magnetic forces are computed from Lorenz's law, illustrate the use of adaptive MLM for practical design of an EM actuator. The paper compares the computed forces against published experimental results.

***Index Terms*—Actuator design, adaptive meshless method, electromagnetic, finite element, magnetic field.**

I. Introduction

RAPID growing interests in developing electric cars, compact appliances, and intelligent robots for traditional and nontraditional industries (such as agriculture, food processing, medical service, and entertainment industries) coupled with the widely available high-coercive rare-earth permanent magnets (PMs) at low cost have motivated the development of novel electromechanical actuators that are geometrically compact and highly energy-efficient. The trends have been further accelerated by the ever increasing cost of energy. Design automation of the electromagnetic actuators involves solving a magnetic field problem.

With the advent of computational technologies, many engineering problems can be solved with numerical methods such as finite-element method (FEM), boundary-element method (BEM), and finite-difference method (FDM). Among these, FEM has been most widely used as it can handle complicated geometry with the help of a mesh generation program. However, computation accuracy of FEM depends on the quality of its mesh. Despite considerable effort has been devoted to improve the design of the mesh and the algorithm to generate it, the creation of a proper element structure remains a challenge; human involvement is still unavoidable for most of engineering analyses with FEM. Furthermore, the need to model both the dimensionally very small air gaps where energy conversion takes place and the remaining electromagnetic structure presents a significant challenge.

Recently, meshless methods (MLM), which inherit many advantages of FEM and yet need no explicit mesh structure to discretize geometry, have been proposed [1]–[3] and applied to some magnetic field computation [4]–[10]. The MLM method based on a similar theoretical framework as FEM has some unique advantages: firstly, it requires only scatter nodes (instead of elemental structure to discretize geometry), which significantly eases the preprocessing task. Secondly, it uses smooth shape functions to interpolate the field variables at a global level, which results in a smooth solution requiring no post-processing. Last but not least, it is an ideal method for adaptive computation since no element reconstruction is needed in the process of nodal insertion.

Most of the earlier research in MLM focuses on proposing new methods for constructing basis functions. Little attention has been focused on solving practical applications which exploit the advantages of MLM. Recently, some research efforts have been seen in solving two technical problems related to adaptive MLM. The first problem is to estimate computational error in MLM. Methods such as residual technique in [11] and recovery technique in [12] are effective, but they are often mathematically difficult to derive and relatively complicated to apply in practice. The second problem is the development of a nodal insertion algorithm that is needed to reconstruct integration cells after the nodes are inserted. Most existing adaptive MLMs use a background cell technique, which demands a significant amount of computation time particularly when the nodal distribution becomes irregular. Methods (such as the quadtree technique) have been proposed to improve the efficiency of the reconstruction process [13], [14]. However, the addition of the computational load cannot be totally eliminated. In [15], a stabilized conforming nodal integration technique has been proposed to avoid the need for constructing background cells. This method has some successful applications in adaptive computation, for example [16], but its extension to three-dimensional computations remains a challenge.

For the above reasons, we offer here an alterative adaptive-MLM for magnetic field computation for designs of electromagnetic actuators. Specifically, this paper offers the following.

1) An Error Estimation Technique Is Offered for Adaptive MLM: We extend the *posteriori* error estimation technique to MLM. This technique was originally developed for FEM based on the observation that the computational results at certain locations (such as nodes) are more accurate than at other locations, and has achieved some successes in adaptive FEM [17], [18]. However, the basis function in MLM is, in general, not a polynomial; the *posteriori* error estimation technique developed for FEM cannot be directly applied to MLM. We present here a modified error estimation built on two different support sizes of

Digital Object Identifier 10.1109/TMAG.2006.876126

a basis function. As will be illustrated with a one-dimensional (1-D) example, this modified error estimation characterizes the true error remarkably well, and its computation in MLM is simpler than in FEM.

2) Practical Issues Related to the Nodal Insertion Are Discussed: We present an automatic node insertion method based on a Voronoi plot technique along with the partition unity integration [19] scheme for obtaining the discretized system of weak-form formulated equations.

While the MLM does not need elements to perform numerical integration as in FEM, most of MLM divide their computational domain into small numerical integration cells (called background cells). When the nodal density increases at a local area, the density of these background cells in that area must increase accordingly in order to ensure the computational accuracy; this makes MLM lose some of its advantages. To overcome this drawback, we introduce a different numerical integration scheme (called the partition unity integration). In this proposed method, a new integration cell is automatically created when a new node is inserted.

3) The Adaptive ML Computation is Validated: Three numerical examples (that have exact solutions) are given validating the computation of adaptive MLM. They also illustrate the processes of error estimation and automatic node insertion, and demonstrate the effectiveness of the adaptive MLM on the convergence. In this paper, the weak form equations are derived using the Galerkin method in the MLM.

4) Applications for Design Analysis of Electromagnetic Actuators Are Illustrated: We illustrate the use of adaptive MLM for design of electromagnetic actuators with high coercive permanent magnets. With two examples, magnetic forces are computed using the Lorenz's force law and compared the computational results against a set of published experimental results. Additionally, we show how MLM can be used to improve the torque-to-weight ratio in the pole design of a three-degree-of-freedom (DOF) spherical motor.

II. Error Estimation in MLM

One of most common methods to improve the accuracy of the numerical approximation is to reduce the nodal space (or increase the density of the nodes). The simplest way is to uniformly increase the nodal density in the whole computational domain. However, if large numerical errors occur only in certain local regions, this method is inefficient since extra nodes in small error regions do not help improve the overall computational accuracy but they would simply lengthen the computation time. Thus, it is desired to have an estimate of the overall error distribution of the computation so that additional nodes can be effectively inserted accordingly, or more specifically, into the large error regions.

The exact numerical error e can be defined as follows:

$$e(\mathbf{x}) = \Phi_e(\mathbf{x}) - \Phi_a(\mathbf{x}) \tag{1}$$

where Φ_{e} and Φ_{a} are the exact potential field distribution and the approximated solution of the MLM respectively. However, Φ_{e} is often unavailable in practice. Thus, a modified form is used to estimate the numerical error:

$$\hat{e}(\mathbf{x}) = \Phi_h(\mathbf{x}) - \Phi_l(\mathbf{x}) \tag{2}$$

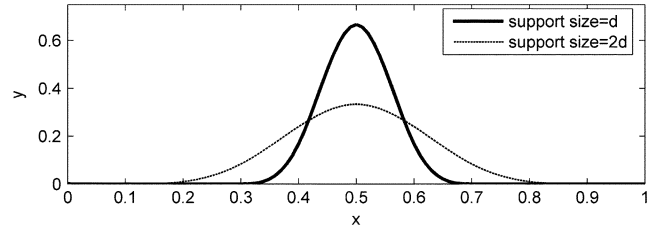


Fig. 1. RKP basis function with two different support sizes.

where Φ_{h} and Φ_{l} are both numerical results but Φ_{h} is more accurate than Φ_{l}. As an example, Φ_{l} is a solution obtained with $n \times n$ number of nodes and Φ_{h} with $2n \times 2n$ nodes. However, it is desired that Φ_{h} can be computed without recalculating with the denser number of nodes for computational efficiency.

We present an alternative error estimation based on two different support sizes of a basis function to locate regions of large numerical errors for the adaptive MLM:

$$\tilde{e}(\mathbf{x}) = \sum_{i=1}^{n} \Psi_{i,d}(\mathbf{x})\Phi_{i,d} - \sum_{i=1}^{n} \Psi_{i,2d}(\mathbf{x})\Phi_{i,2d} \tag{3}$$

where $\Psi_{\mathrm{i,d}}$ and $\Psi_{\mathrm{i,2d}}$ denote the basis functions at the ith node with a support size d and $2d$ respectively; $\Psi_{\mathrm{i,d}}$ is the solution solved in the previous computation step; and $\Phi_{\mathrm{i,2d}}$ is the fitted result using the basis function with a support size of $2d$. As an example, we include a commonly used MLM basis function (known as the reproducing kernel particle, or the RKP basis function) in the Appendix. If the basis function is noninterpolating (as often the case in MLM), $\Phi_{\mathrm{i,2d}}$ can be solved from the following system of linear equations:

$$\Phi_{2d} = \mathbf{E}^{-1}\Phi_d \tag{4}$$

where the elements of the matrix $\mathbf{E}$ are given as

$$\varepsilon_{i,j} = \Psi_{j,2d}(\mathbf{x}_i). \tag{5}$$

The rationale for (3) can be explained with the aid of Fig. 1, which compares two different support sizes of a RKP basis function. As shown in Fig. 1, the larger the support size the smoother is the basis function. In general, it is more difficult for the basis function with a larger support size to approximate a function with an abrupt change in the solution. Thus, regions of large errors can be characterized by comparing the approximation solutions solved using the two different basis functions. Numerical experiments have confirmed this finding.

Once the error is estimated from (3), locations of large errors are identified as follows:

$$\forall \mathbf{x}_a : \tilde{e}(\mathbf{x}_a) > e_p \tag{6}$$

where $\mathbf{x}_a$ is the test location; and e_p is a specified error threshold. We illustrate the error estimation with an example.

Example 1. 1-D Problem Illustrating the Error Estimation: We illustrate the method here using a 1-D problem characterized by the second-order ordinary differential equation:

$$d^2y/dx^2 = f(x) \tag{7}$$

where $f(x) = -6x - e^{-(2x-1)^2/4\alpha^2}[2 - (2x-1)^2/\alpha^2]/\alpha^2$. The boundary conditions are

$$y(0) = -e^{-1/4\alpha^2} \quad \text{and} \quad y(1) = -1$$

where α is a constant used to control the shape of the solution. The exact solution is given by $y = -x^3 - e^{-(2x-1)/4\alpha^2}$.

To illustrate the error estimation, we solve (7) numerically using MLM with weak form formulation (WFF). The weak form equation is obtained by substituting the ML approximation

$$\hat{y}(x) = \sum_{i=1}^{n} \Psi_i(x) y_i$$

into (7) and integrating the result by parts, which yields

$$\int_0^1 \frac{d\Psi_j}{dx} \sum_{i=1}^{n} \frac{d\Psi_i}{dx} y_i dx = -\int_0^1 \Psi_j f(x) dx. \tag{8}$$

For the purpose of inserting additional nodes, we compare the estimated error at the midpoint between two adjacent nodes against the exact error. As shown in Fig. 2(a), the solution has a high gradient region around $x = 0.5$. Fig. 2(b) shows that the results of the MLM (with a uniform distribution of 21 nodes) has a relatively large error around the high gradient region of $x = 0.5$.

In order to insert additional nodes efficiently, the error estimation must identify this large error region faithfully with reasonable accuracy. As compared in Fig. 2(b), the estimated error characterizes the true error remarkably well, and its computation in MLM is simpler than in FEM.

III. Adaptive Node Insertion for MLM

Additional nodes can be inserted into the computational domain using the Voronoi plot [20] technique that constructs one Voronoi cell for each node.

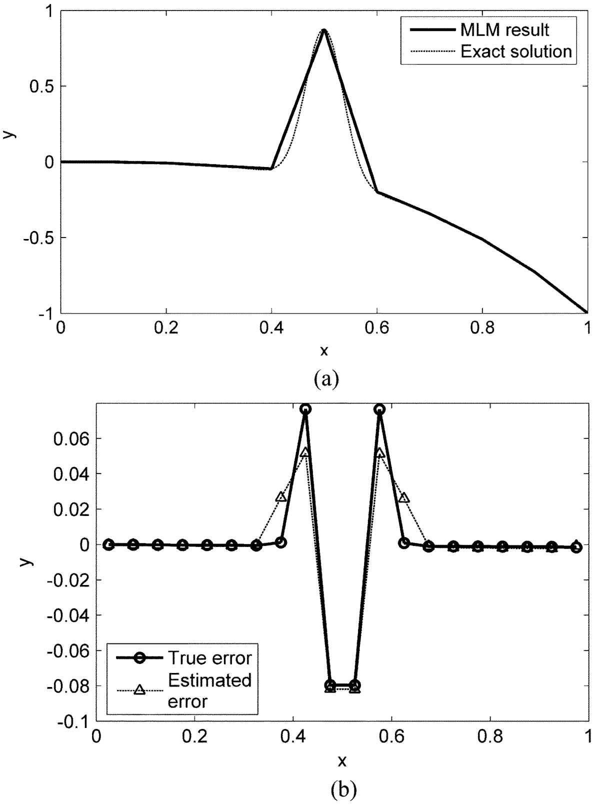
A. Node Insertion Scheme

An example Voronoi plot for a 2-D computational domain is shown in Fig. 3, where the solid dots represent the nodes and the dashed lines are boundaries of the Voronoi cells. As shown in Fig. 3, a Voronoi cell is a polygon containing all the points closest to the node that it surrounds. The error at the vertexes of each Voronoi cell is computed from (3). If the error at a corner point satisfies criterion (6), a new node is created at that point as illustrated in Fig. 3 where the three triangles at the corners of a Voronoi cell are example regions of large numerical errors.

B. Support Size

The support size of the inserted node is calculated using (9) as the maximum distance from the node to its surrounding nodes whose Voronoi cell is adjacent to this node:

$$r_i = a_p \cdot \max(\|\mathbf{x}_j - \mathbf{x}_i\|) \tag{9}$$



(a)

(b)

Fig. 2. Comparison between exact and estimated errors. (a) Exact versus computed solutions. (b) Exact versus estimated error.

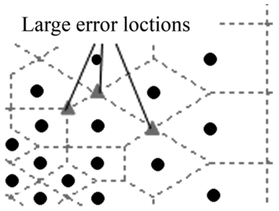


Fig. 3. Voronoi plot with three large error points.

where r_i is the support radius for ith node; $\mathbf{x}_\mathrm{i}$ and $\mathbf{x}_\mathrm{j}$ are the coordinates of ith and jth nodes, respectively. The Voronoi cell of the jth node is adjacent to the Voronoi cell of the ith node. In (9), a_p is a constant coefficient normally taken a value between 1 to 3. For the newly inserted node, the support size of its basis function must be chosen carefully considering the following tradeoffs. 1) The support radius must be sufficiently large to cover enough nodes for constructing the ML basis function. On the other hand, it is desired to localize the effect of the newly inserted nodes, and thus the support radius should be kept small. 2) Computational load increases with the support radius.

C. Partition Unity Integration

The partition unity integration performs the numerical integration based on the support size of the basis function. When a new node is inserted, a new integration cell is automatically created as illustrated in Fig. 4 and thus, this numerical integration scheme is very suitable for adaptive computation.

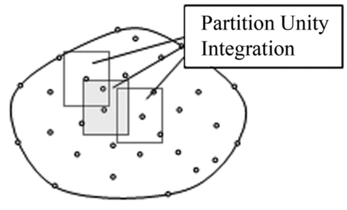


Fig. 4. Partition unity integration cells.

Most of the basis functions used in MLM (including RKP basis function) have the partition unity property:

$$\sum_{i=1}^{n}\Psi_i(\mathbf{x}) = 1 \qquad (10)$$

with which the integration for an arbitrary function $f(\mathbf{x})$ in the computational domain can be computed as follows:

$$\int\limits_{\Omega} f(\mathbf{x})d\mathbf{x} = \int\limits_{\Omega} f(\mathbf{x})\sum_{i=1}^{n}\Psi_i(\mathbf{x})dx = \sum_{i=1}^{n}\int\limits_{\Omega} f(\mathbf{x})\Psi_i(\mathbf{x})dx \qquad (11)$$

where Ω is the computational domain. To exclude points outside the computational domain, (11) is written such that the integration is within the support domain S_i of ith basis function:

$$\sum_{i=1}^{n}\int\limits_{\Omega} f(\mathbf{x})\Psi_i(\mathbf{x})dx = \sum_{i=1}^{n}\int\limits_{S_i} f(\mathbf{x})P(\mathbf{x})\Psi_i(\mathbf{x})dx \qquad (12)$$

where

$$P(\mathbf{x}) = \begin{cases} 1, & \text{when } \mathbf{x} \in \Omega \\ 0, & \text{when } \mathbf{x} \notin \Omega. \end{cases}$$

The global integration for the whole computational domain is divided into n subintegration domains and performed upon the support domain of n basis functions. Because the support domain of the basis functions, in general, has a regular shape, the conventional numerical integration scheme such as Gaussian quadrature can be applied easily.

Example 2: Effect of Adaptive Node Insertion on High Gradient: Fig. 5 visually demonstrates the converging process of the adaptive MLM for the 1-D example problem given in Example 1 (Fig. 2), where the initial computation (of 11 uniformly distributed nodes) was compared with the exact solution. At the end of each computation, the error at the midpoint of every two adjacent nodes is estimated, which is shown in Fig. 5(a). New nodes are inserted at regions where the estimated errors exceed a predefined threshold. In this research, the average of the error distribution is used as the error threshold. The results of automatic node insertion are given in Fig. 5(b). As shown in Fig. 5, all inserted nodes concentrate around high gradient region indicating that the automatic node insertion correctly locates the large error region. We then apply the method of partition unity integration to (8):

$$\sum_{k=1}^{n}\oint\limits_{L_k}\Psi_k P(x)\frac{d\Psi_i}{dx}\sum_{i=1}^{n}\frac{d\Psi_i}{dx}y_i dx = \sum_{k=1}^{n}\oint\limits_{L_K}\Psi_k P(x)\Psi_j f(x)dx. \qquad (13)$$

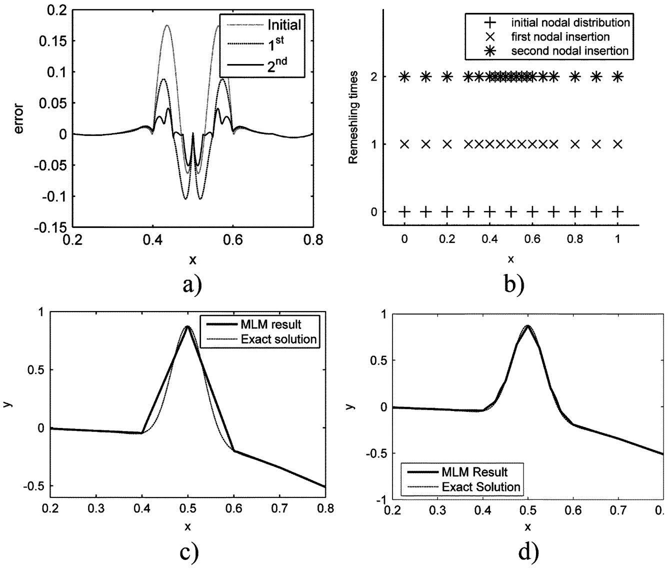


Fig. 5. Converging process of a 1-D problem. (a) Computational error, (b) nodal insertion process, (c) first adaptive (15 nodes), and (d) second adaptive (19 nodes).

As seen in Fig. 5(c) and (d), where the support radius of 1.2 is chosen for the adaptive computation, the results of the adaptive MLM converges to the exact solution rapidly after two adaptive computations, each of which uses four additional nodes.

Example 3. Effect of Adaptive Insertion on Converging Speed: Consider the 2-D problem

$$\nabla^2 u = \nabla^2 f(x,y) \qquad (14)$$

with the following boundary conditions:

$$\begin{aligned} &u(x,0) = 0; u(x,1) = 0; \\ &u(0,y) = 0; u(1,y) = 0. \end{aligned} \qquad (14a)$$

The exact solution is given by

$$f(x,y) = 5x^2y^2(1-x)^2(1-y)^2(e^{10x^2}-1)(e^{10y^2}-1). \qquad (15)$$

We investigate here the effects of the adaptive node insertion on the converging speed by comparing it against a commonly used weak-form-formulated MLM (with a globally uniform distribution of nodes). In other words, the nodes of the uniform-node MLM are increased uniformly in the computational domain while the adaptive MLM increases its nodes according to the estimated errors. Both methods start with an initial computation of 6 × 6 nodes. Three successive insertions are performed for each method. For the uniform-node MLM, the three successive node distributions are 8 × 8, 9 × 9, and 13 × 13. The node insertion of the adaptive MLM is automatically generated using the estimated error criterion (6) and is demonstrated in Fig. 6.

For the purpose of comparison against the exact solution plotted in Fig. 7(a), we define the computation error as follows:

$$\text{error} = \sqrt{\oiint (u_{\text{computed}} - u_{\text{exact}})^2 d\Omega}. \qquad (16)$$

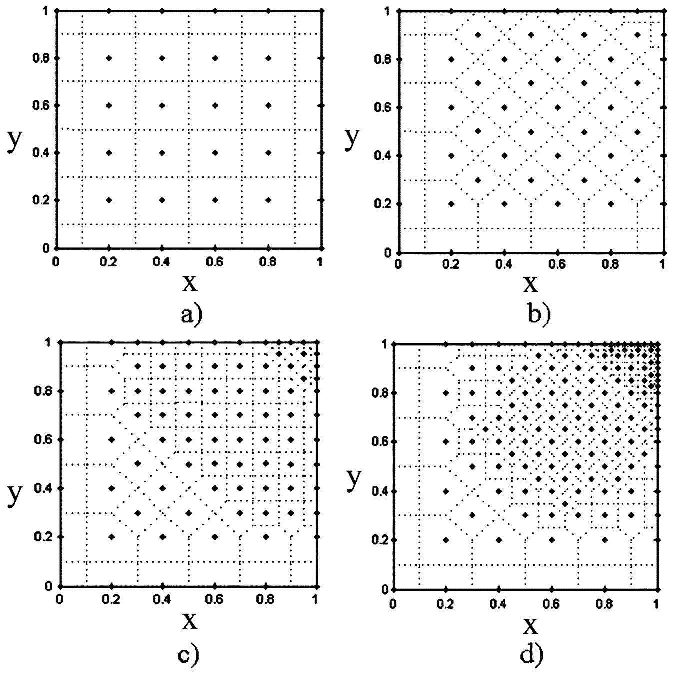


Fig. 6. Process of adaptive nodal insertion. (a) Initial nodal distribution, (b) first nodal insertion (54), (c) second nodal insertion (84), and (d) third titme nodal insertion (146).

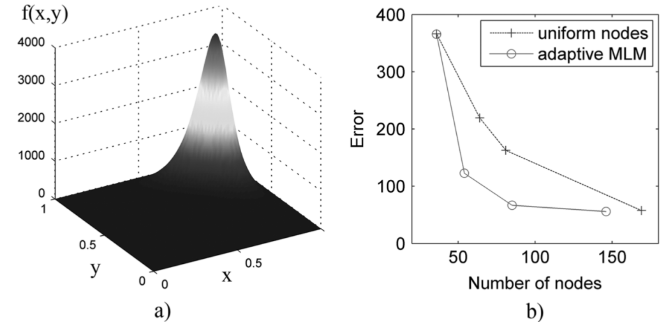


Fig. 7. Exact solution of the example and error comparison. (a) Exact solution. (b) Error comparison.

Since the computational time is directly proportional to the number of nodes, the comparison is made by plotting the computational errors versus the number of nodes in Fig. 7(b). The following observation can be made from the converging process of two methods.

1) Both methods tend to converge to the exact solution as the number of nodes increase. However, the adaptive MLM has a significantly higher converging rate.
2) The adaptive MLM effectively identifies the large error regions occurring around the regions of high gradient as expected.
3) The first two nodal insertions result in rapid error reduction as compared to the third insertion. As the number of nodal insertions increase, the error caused by highly irregular nodal distribution may gradually outweigh the benefit generated by additional nodes hinting that the number of nodal inserting iterations should not be too high in order to maintain the efficiency of algorithm.

IV. Applications to EM Actuators

The prediction of magnetic forces involved in the design of an electromagnetic (EM) actuator relies on the solution of its magnetic field. Due to the field discontinuity at the material interface, and the fact that the air gap between the stator and rotor poles is often very small as compared to the dimension of the overall field distribution, it is often difficult for the regular MLM with a smooth basis function to achieve satisfactory accuracy around the air gaps where energy conversion takes place. The following example shows how adaptive MLM can be used effectively to improve the field accuracy around these regions. Specifically, the objectives of this example are as follows.

1) Illustrate the use of adaptive node insertion to approximate discontinuities around the material interface.
2) For validation of the adaptive MLM computation, the magnetic force between a permanent magnet (PM) and an electromagnet (EM) calculated using Lorenz Law with the computed field is compared against published experimental results.
3) Demonstrate how the adpative MLM can be used as an effective tool to analyze designs. As an example, we consider the pole design of a spherical motor which is capable of providing three-DOF motion in a single joint [21]. By comparing two different designs, we illustrate the effects of design geometry on the torque-to-weight characteristics.

For the above objectives, we use the adaptive MLM to solve for the magnetic field intensity $\mathbf{H} = -\nabla\Phi$ around a cylindrical permanent magnet (uniformly magnetized along its axis, $\mathbf{M} = M_o\mathbf{i}_z$) in free space, where Φ is the magnetic scalar potential. The results provide a basis for computing the magnetic force under the influence of an electromagnet using Lorenz's law. In cylindrical coordinates, Φ can be solved from the Laplacian equation:

$$\nabla^2\Phi = \frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Phi}{\partial\rho}\right)+\frac{1}{\rho^2}\frac{\partial^2\Phi}{\partial\theta^2}+\frac{\partial^2\Phi}{\partial Z^2}=0 \qquad (17)$$

where $\rho = r/R, Z = z/R$, and R is the radius of the magnet. For this axisymmetric problem; $\partial\Phi/\partial\theta = 0$. The BC at the infinity far boundary is

$$\Phi_{\mathbf{x}\to\infty} = 0. \qquad (17a)$$

At the material interface, $\mathbf{H}$ is continuous along the tangential direction; and the flux density $\mathbf{B}$ is continuous along its normal. In terms of scalar potential functions

$$\Phi_p = \Phi_q \qquad (17b)$$

$$(\mu_{\mathrm{rq}}\nabla\Phi_q - \mu_{\mathrm{rp}}\nabla\Phi_p)\cdot\mathbf{n} = (\mathbf{M}_q - \mathbf{M}_p)\cdot\mathbf{n} \qquad (17c)$$

where the subscripts p and q denote as two different regions. In addition, for the symmetry

$$\partial\Phi/\partial r = 0 \quad \text{at } r = 0. \qquad (17d)$$

The weak form of this is shown in (18)

$$\oint_{\Gamma_m}\hat{\Psi}_j(\mathbf{M}_1-\mathbf{M}_2)\cdot\mathbf{n}d\Gamma - \oint_{\Omega}\mu_r\nabla\hat{\Psi}_j(\mathbf{x})\nabla\left(\sum_{i=1}^{n}\hat{\Psi}_i(\mathbf{x})\Phi_i\right)d\Omega = 0. \qquad (18)$$

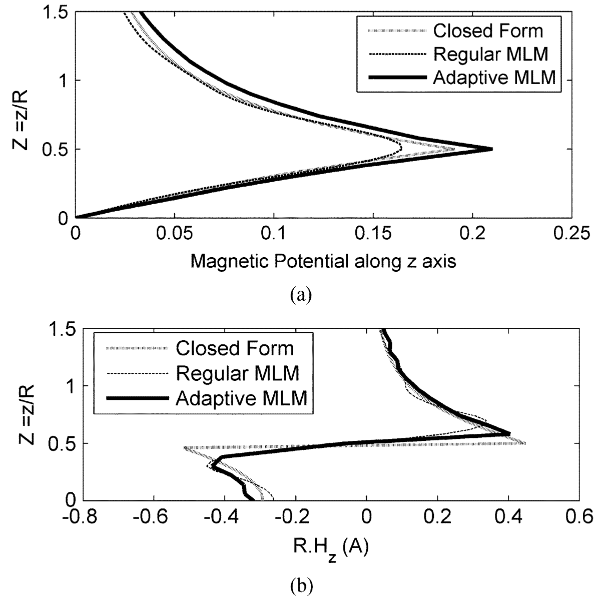


Fig. 8. Effect of adaptive node on handling of discontinuity. (a) Magnetic scalar potential. (b) Magnetic field intensity.

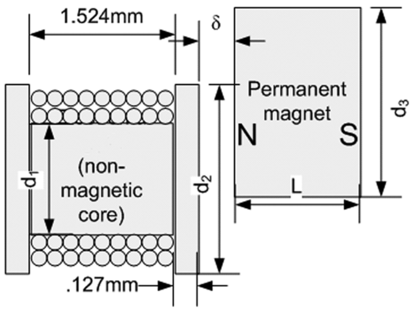


Fig. 9. Experiment configuration.

Once the magnetic field is computed, the force can be obtained by integrating the force density exerted on the current carrying conductor by its interaction with the magnetic field:

$$\mathbf{F} = \int_V (\mathbf{J} \times \mathbf{B}) \mathrm{rdrd}\theta dz \tag{19}$$

where $\mathbf{J}$ is current density and V is the volume of the current conductor.

A. Approximation of a Discontinuity Magnetic Field

The magnetic scalar potential and field intensity along the z axis are obtained by the adaptive MLM and compared against the closed form solution [22] in Fig. 8(a) and (b), respectively. In this example, 41×41 nodes are used in the regular MLM. The adaptive MLM starts with a uniform nodal distribution of 221 (uniform 13×17) nodes and increases to 658 nodes after three successive insertions. As in Example 2, all the additional nodes are inserted into large gradient regions around the magnet pole.

TABLE I
DIMENSIONS OF EXAMPLE 4

Configurations	Large	Small
d_1 (mm)	3.048	1.524
d_2 (mm)	3.962	3.175
d_3 (mm)	2.998	1.6
L (mm)	1.6	0.8128
Coil res. (Ω)	57	32
Wire length (m)	3	1.68
Samarium-Cobalt magnet; $\mu_0 M_0$=1.02 T		

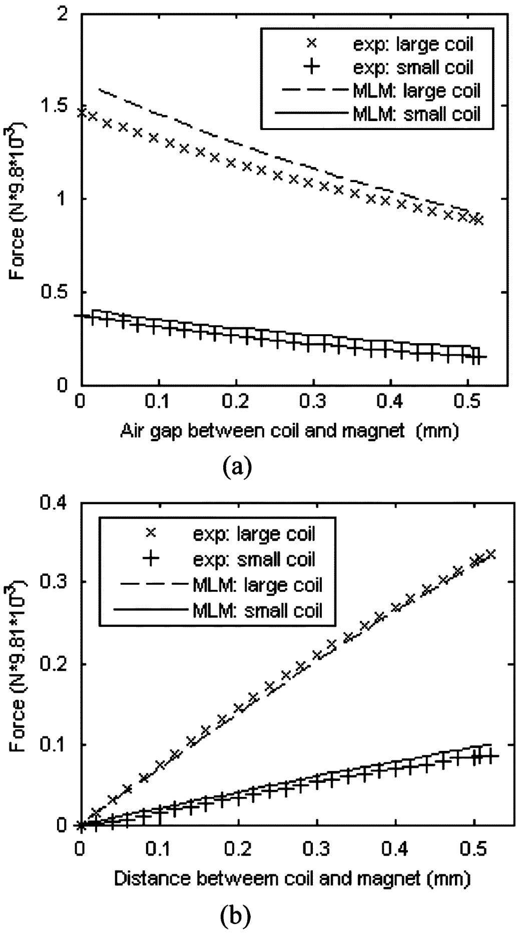


Fig. 10. Comparison between computed and experimental results. (a) Axial force. (b) Restoring force.

While the result of the uniform-node MLM offers a very good overall prediction but exhibits some oscillations near the discontinuous interface. As compared in Fig. 8, the potential and field intensity computed by the adaptive MLM with only 658 nodes (less than half of the number of nodes used by the uniform-node MLM) match the closed form solution very well at the material interface. With a higher nodal concentration at the material interface, the adaptive MLM is able to approximate the discontinuity satisfactorily with a continuous basis function. This suggests that the adaptive MLM is a good alternative for solving problem with field discontinuity.

Example 4: Force Between a Permanent Magnet and an Electromagnet: This example is selected from one of the T.E.A.M. problems [23], where the experimental setup is shown in Fig. 9 and Table I.

Fig. 10 compares the computed forces against published experimental data [23]. The computed restoring force matches the experimental results very well while the computed axial

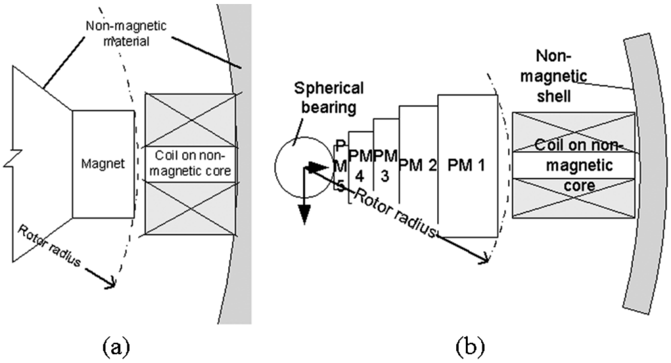


Fig. 11. Comparison of pole designs (not to scale). (a) Design 1 [24], [25]. (b) Design 2 [24], [25].

TABLE II
PARAMETERS USED IN SIMULATION

Design	Rotor radius, mm	Stator Coil OD×ID×L (mm)	# of turns	PM rotor pole OD×L (mm)
1	37.5mm	19.05×9.53×25.4	1050	12.7×12.7
2	93	18×4×27	900	25×10, 20×5, 16×6, 12×3, 8×3

Air gap = .5mm; Magnetization $\mu_0 M_0$=1.27T

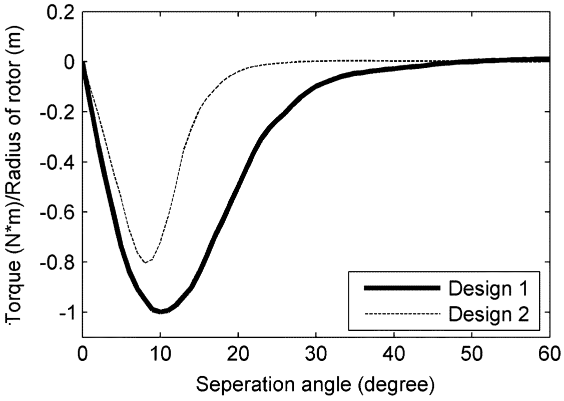


Fig. 12. Comparison of torque per unit radius.

force is slightly larger than the measured force but within 15% difference.

Example 5: Pole Design of a Three-DOF Spherical Actuator: Fig. 11 compares the pole design of two different prototype spherical motors [24], [25]. The geometry and layout of the pole have a significant influence on the torque performance of a spherical motor [21]. The rotor of Design #1 [24] consists of two rows of eight small PMs whereas Design #2 [25] uses one row of eight large PMs as rotor poles. Detailed geometries of the two pole designs are given in Table II.

The torque between the EM and PM is computed using Lorenz law from the adaptive-MLM computed field for the two configurations. To provide a common basis for comparison, we compare the torque output per unit radius for a single EM-PM pole-pair. As compared in Fig. 12, the torque per unit radius for Design #1 is significantly higher although Design #2 uses a much larger PM rotor pole for similar input power. In addition, the compact pole geometry in Design #1 allows for a larger number of rotor poles to be used, which could further improve the torque-to-volume ratio for a specified input power.

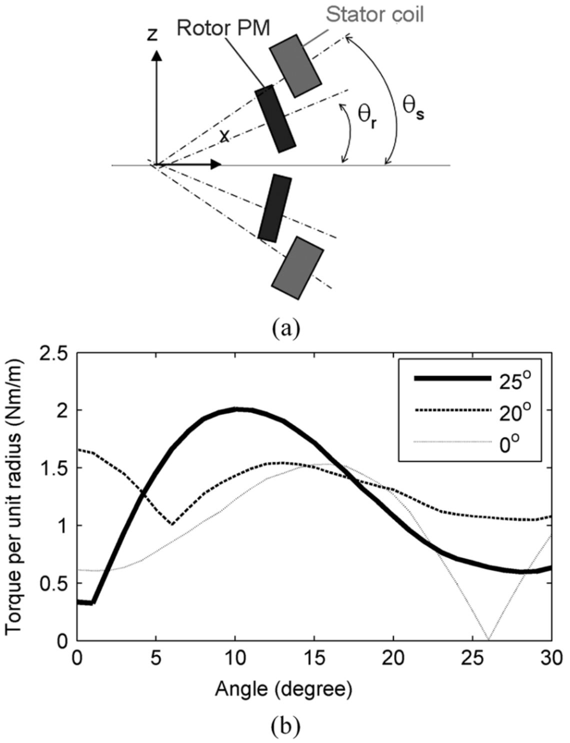


Fig. 13. Effect of pole layout on torque generated. (a) Pole geometry. (b) Torque for different $\theta_r (\theta_s = 26°)$.

To investigate the effect of the PM-pole layout (denoted as θ_r in Fig. 13) on the torque profile for Design #1, we compute the torque about the y axis for three different rotor pole layouts for a specified pair of EM poles spaced at $\theta_s = 26°$. The three different layouts are $\theta_r = 0°$, 20°, and 25°. For the case $\theta_r = 0°$, only one PM that has the same length but twice the volume is used. As shown in Fig. 13(b), the pole layout has a significant effect on the torque profile (both its maximum magnitude and the smoothness of its shape). As illustrated in Figs. 12 and 13(b), the maximum magnitude of the torque and the range of its influence can be doubled but only with a carefully selected set of design parameters. These results illustrate how the adaptive MLM can be effectively used to analyze the effects of pole design on the torque performance of an EM actuator.

V. CONCLUSION

A relative complete adaptive computational method for MLM has been presented and illustrated with several examples. This practical method, which has been validated by comparing against exact solutions, overcomes two technical difficulties associated with MLM; namely, error estimation and nodal insertion. Our results show that the method can faithfully locate large error regions, automatically insert nodes to these regions without human involvement, and improve the computational efficiency significantly. Comparing the converging speed of the adaptive MLM against a regular MLM (with uniform distributed nodes) shows that the adaptive MLM is effective and computationally efficient. Additionally, we compare the electromagnetic forces computed using Lorenz's law with the field results predicted by the adaptive MLM against published experimental results, which show excellent agreement.

APPENDIX
REPRODUCING KERNEL PARTICLE (RKP) BASIS FUNCTION

The RKP basis function can be expressed as

$$\Psi_i(\mathbf{x}) = C(\mathbf{x};\mathbf{x}-\mathbf{x}_i)\Lambda(\|\mathbf{x}-\mathbf{x}_i\|/d) \tag{A1}$$

where $\Lambda(\|\mathbf{x}-\mathbf{x}_i\|/d)$ is a kernel (or weight) function centered at $\mathbf{x}_i$; the support size d is a design parameter that influences the effective region of the kernel function.and $C(\mathbf{x};\mathbf{x}-\mathbf{x}_i)$ is a set of enrichment functions that vary with the location of approximation $\mathbf{x}$. The following cubic B-spline function [14] is chosen for the kernel function:

$$\Lambda(\mathbf{x}-\mathbf{x}_i) = \begin{cases} 2/3 - 4p^2 + 4p^3, & \text{for } 0 \le p \le 1/2 \\ 4(1-3p+3p^2-p^3)/3, & \text{for } 1/2 \le p \le 1 \\ 0, & \text{for } p \ge 1 \end{cases} \tag{A2}$$

where $p = \|(\mathbf{x}-\mathbf{x}_i)\|/d$; The function $C(\mathbf{x};\mathbf{x}-\mathbf{x}_i)$ is given by [14]:

$$C(\mathbf{x};\mathbf{x}-\mathbf{x}_i) = \mathbf{h}^{\mathrm{T}}(0)\mathbf{P}^{-1}(\mathbf{x})\mathbf{h}(\mathbf{x}-\mathbf{x}_i) \tag{A3}$$

where $\mathbf{h}^{\mathrm{T}}(\mathbf{x}-\mathbf{x_i}) = [1\ (\mathbf{x}_1-\mathbf{x}_{i1})\ \cdots\ (\mathbf{x}_n-\mathbf{x}_{\mathrm{in}})^n]$; and $\mathbf{h}^{\mathrm{T}}(0) = [1\ 0\ 0\ \cdots\ 0]$. When a new node is inserted, the basis functions of the original nodes are also updated. In order to minimize computation, the moment matrix $\mathbf{P}$ is updated using its value from the previous computational step:

$$\mathbf{P}(\mathbf{x}) = \mathbf{P}_o(\mathbf{x}) + \sum_{i=1}^{n_a} \mathbf{h}(\mathbf{x}-\mathbf{x}_i)\mathbf{h}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_i)\Lambda(\mathbf{x}-\mathbf{x}_i) \tag{A4}$$

where n_a is the number of newly added nodes.

To facilitate the procedure for imposing the boundary conditions in WFF, a modified basis function $\hat{\Psi}_i(\mathbf{x})$that can be computed from the RKP basis function (A1) and has an interpolation property is introduced:

$$\hat{\Psi}_i(\mathbf{x}) = \sum_{j=1}^{n} \Psi_i(\mathbf{x}) L_{ij}^{-\mathrm{T}} \tag{A5}$$

where the element L_{IJ} is defined by

$$L_{ij} = \Psi_i(x_J) \tag{A6}$$

such that

$$\tilde{\Phi}(\mathbf{x}) = \sum_{i=1}^{n} \hat{\Psi}_i(\mathbf{x})\Phi_i. \tag{A7}$$

ACKNOWLEDGMENT

This work was supported in part by the Georgia Agriculture Technology Research Program and in part by the U.S. Poultry and Eggs Association.

REFERENCES

[1] I. Babuska and J. M. Melenk, "The partition of unity method," *Int. J. Numer. Methods Eng.*, vol. 40, pp. 727–758, 1997.

[2] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: An overview and recent developments," *Comput. Methods Appl. Mech. Eng.*, vol. 139, p. 3, 1996.

[3] C. A. Duarte and J. T. Oden, "h-p adaptive method using clouds," *Comput. Methods Appl. Mech. Eng.*, vol. 139, p. 237, 1996.

[4] V. Cingoski, N. Miyamoto, and H. Yamashita, "Element-free Galerkin method for electromagnetic field computations," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3236–3239, Sep. 1998.

[5] K. Do Wan and K. Hong-Kyu, "Point collocation method based on the FMLSRK approximation for electromagnetic field analysis," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1029–1032, Mar. 2004.

[6] X. Liang, Z. Zhiwei, B. Shanker, and L. Udpa, "Element-free Galerkin method for static and quasi-static electromagnetic field computation," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 12–20, Jan. 2004.

[7] S. Liu, "Improvement of the element-free Galerkin method for electromagnetic field calculation," *IEEE Trans. Appl. Supercond.*, vol. 14, no. 2, pp. 1866–1869, Jun. 2004.

[8] Y. Marechal, "Some meshless methods for electromagnetic field computations," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3351–3354, Sep. 1998.

[9] I. Nishiguchi and K. Haseyama, "Application of element-free Galerkin method to 2-D and 3-D magnetostatic problems," in *Proc. Int. Symp. Non-linear Electromagnetic Systems (ISEM'99)*, Pavia, Italy, 2000, pp. 327–330.

[10] Y. Shiyou, N. Guangzheng, J. R. Cardoso, S. L. Ho, and J. M. Machado, "A combined wavelet-element free Galerkin method for numerical calculations of electromagnetic fields," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1413–1416, May 2003.

[11] S.-H. Park, K.-C. Kwon, and S.-K. Youn, "A posteriori error estimates and an adaptive scheme of least-squares meshfree method," *Int. J. Numer. Methods Eng.*, vol. 58, pp. 1213–1220, 2003.

[12] R. Rossi and M. K. Alves, "An h-adaptive modified element-free Galerkin method," *Eur. J. Mech. A, Solids*, vol. 24, p. 782, 2005.

[13] T. Rabczuk and T. Belytschko, "Adaptivity for structured meshfree particle methods in 2D and 3D," *Int. J. Numer. Methods Eng.*, vol. 63, p. 1559, 2005.

[14] A. Tabarraei and N. Sukumar, "Adaptive computations on conforming quadtree meshes," *Finite Elements Anal. Des.*, vol. 41, p. 686, 2005.

[15] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You, "Stabilized conforming nodal integration for Galerkin mesh-free methods," *Int. J. Numer. Methods Eng.*, vol. 50, p. 435, 2001.

[16] Y. You, J. S. Chen, and H. Lu, "Filters, reproducing kernel, and adaptive meshfree method," *Comput. Mech.*, vol. 31, p. 316, 2003.

[17] O. C. Zienkiewicz and J. Z. Zhu, "Superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity," *Int. J. Numer. Methods Eng.*, vol. 33, p. 1365, 1992.

[18] O. C. Zienkiewicz and J. Z. Zhu, "Superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique," *Int. J. Numer. Methods Eng.*, vol. 33, p. 1331, 1992.

[19] A. Carpinteri, G. Ferro, and G. Ventura, "The partition of unity quadrature in meshless methods," *Int. J. Numer. Methods Eng.*, vol. 54, p. 987, 2002.

[20] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, p. 469, 1996.

[21] K. M. Lee and C. K. Kwan, "Design concept development of a spherical stepper for robotic applications," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 175–181, Feb. 1991.

[22] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[23] N. I. J. P. A. Bastos, "Forces in Permanent Magnets Team Workshop Problem 23," [Online]. Available: http://www.compumag.co.uk/team.html.

[24] K.-M. Lee and H. Son, "Torque model for design and control of a spherical wheel motor," in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM2005)* Monterey, CA, 2005, pp. 335–340.

[25] L. Yan, I.-M. Chen, C. K. Lim, G. Yang, W. Lin, and K.-M. Lee, "Torque modeling of a DC spherical actuator based on lorentz force law," presented at the Int. Conf. Control, Automation, Robotics and Vision, Singapore, 2002.

Manuscript received March 4, 2006; revised April 24, 2006. Corresponding author: K.-M. Lee (e-mail: kokmeng.lee@me.gatech.edu).