

# Analysis and Design of Parallel 3-D Mesh Refinement Dynamic Load Balancing Algorithms for Finite Element Electromagnetics With Tetrahedra

Dennis D. Giannacopoulos and Da Qi Ren

Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada

**We develop a simulation-based approach for the computational analysis and design of dynamic load balancing algorithms in parallel three-dimensional unstructured mesh refinement with tetrahedra. A Petri Nets model is implemented based on a random polling algorithm and the target multiprocessor architecture, which simulates the behavior of the parallel mesh refinement. Subsequently, estimates for performance measures are derived from discrete event simulations. The benefits of this new approach for developing high-performance parallel mesh refinement algorithms are demonstrated with results for an example geometric mesh refinement model.**

**Index Terms**—Adaptive systems, electromagnetic analysis, finite element method (FEM), parallel processing, software methodology.

## I. INTRODUCTION

**T**HE finite element method (FEM) is a powerful numerical technique for the approximate solution of continuum electromagnetic problems. However, efficient and accurate solutions for some three-dimensional (3-D) modern applications require extremely large numbers of elements. In such cases, parallel processing is beneficial for each stage of the FEM including mesh refinement [1]–[3]. Due to the computational complexity of 3-D, parallel, unstructured mesh generation and refinement, the performance of the method is highly dependent on several factors, e.g., the underlying algorithm, the inter-processor communication pattern, the synchronization of tasks, etc. The goal of modeling and simulating parallel mesh refinement is to examine the specific parallel system architecture and software techniques in advance, in order optimize its design and thus achieve the best possible performance for a given cost.

Today, techniques used for parallel mesh refinement performance analysis are, typically, based on benchmarking programs on known environments. Unfortunately, these types of deterministic evaluations are inefficient for performance studies in the early design stages of a parallel system. For example, various analysis and design methodologies reported in the literature have been used to address separately specific load balancing algorithms or mesh refinement models. In other words, these approaches focus, typically, on only one particular aspect of a whole parallel mesh refinement system [3]–[5]. Thus, such separately focused analyses cannot predict the performance of the whole parallel mesh refinement system accurately. One promising route for overcoming these limitations is based on extending a preliminary approach developed previously by the authors for using Petri Nets to simulate parallel mesh refinement [6]. In order to obtain more accurate simulation results for the performance analysis of a whole parallel mesh refinement system, the distributed data structure, geometric mesh refinement model, dynamic load balancing (DLB) algorithm, parallel system architecture, and the inter-processor communication details must all be incorporated in the Petri Nets model in a realistic fashion. This type of holistic simulation is

an innovative approach that we believe will lead to improved analysis and design methods for parallel 3-D mesh refinement DLB algorithms for finite element electromagnetics.

In this paper, we develop a new approach for the modeling, analysis and design of DLB algorithms in parallel finite element mesh refinement that utilizes Petri Nets. Petri Nets-based models allow for a relatively detailed description of a system due to their formal syntax and functional semantics, and can reveal key characteristics of system performance stochastically. While Petri Nets have been used for discrete event-based simulation of various applications, to our knowledge, they have not been considered previously for parallel 3-D mesh refinement DLB for finite element electromagnetics with tetrahedra [6]–[8]. In addition, we use the proposed approach for the design of a random polling (RP)-DLB algorithm for a specific 3-D parallel mesh refinement model suitable for FEM electromagnetics with tetrahedra [6], [8], [9].

## II. GEOMETRIC MESH REFINEMENT MODEL

For 3-D electromagnetic analysis and design with the FEM, tetrahedra are often employed to achieve the geometric discretization of the problem domain, because they possess several desirable computational modeling properties. However, one difficulty is the potential geometric complexity involved in mesh refinement, which is often necessary to improve the solution accuracy to within required tolerances. There are several tetrahedral mesh refinement strategies. To solidify concepts, consider the subdivision of a tetrahedron as shown in Fig. 1(a). This method consists of cutting every edge into two and every face into four triangles, resulting in four tetrahedra, each a half-scale duplicate of the original, and an octahedron [9]. The octahedron is kept in an element list, and it is temporarily split into four tetrahedra just for matrix assembly purposes if necessary, as in Fig. 1(c). These four tetrahedra are not similar to the original one and they cannot be used for further subdivision because this may result in the progressive deterioration of mesh quality. In order to maintain the original mesh quality, the octahedra kept from the element list are subdivided in the next iteration by bisecting each edge, i.e., cut into six smaller size octahedra and eight tetrahedra as shown in Fig. 1(b). These eight tetrahedra are similar to the original tetrahedron, though reduced by a factor of four in each dimension. The

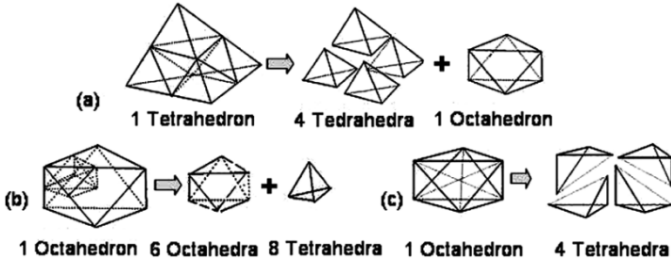


Fig. 1. Mesh refinement model. (a) Tetrahedron subdivision. (b) Primary octahedron subdivision. (c) Secondary octahedron subdivision.

four temporary tetrahedra are then discarded. In finite element applications, the subdivisions of Fig. 1(a) and (b) are repeated until all of the new tetrahedra satisfy specified mesh criteria. Any remaining octahedra are each cut into four additional tetrahedra as shown in Fig. 1(c). This mesh refinement model is considered because of its potential to allow refinement to proceed simply without changing the mesh quality as well as it makes matrix assembly straightforward in FEM computation [3]. It may be noted that the tetrahedral and octahedral refinement rules of Fig. 1(a) and (b) generate tetrahedra of the same quality as the original [9]; however, this is not necessarily the case when the subdivision rule of Fig. 1(c) is applied to terminate the mesh refinement process for FEM applications, and other mesh optimization techniques can be applied to improve the quality of the resulting elements [1], [3], [9], [10].

### III. RP-DLB PARALLEL MESH REFINEMENT MODEL

In this section, the RP-DLB modeling strategy is briefly explained, and key algorithmic details are given. A master-slave parallel computing scheme is considered for this work.

#### A. RP-DLB Algorithm

1) *Overview of Parallel Mesh Refinement With RP-DLB*: Fig. 2 shows a conceptual outline of the key elements for our parallel mesh refinement with RP-DLB algorithm. To begin, the input data are read, parsed, and checked by the master processing element (PE). The master PE partitions the initial geometric mesh model into subdomains based on the anticipated work load of each slave PE, then broadcasts the subdomain assignments to corresponding slave PEs, as shown in Fig. 2. The slave PEs executing the tetrahedral-octahedral subdivision algorithm (Fig. 1) each work in parallel on their respective domains during this phase, and a RP-DLB mechanism is applied [4]. While the work is not finished, all slave PEs will work in parallel asynchronously; however, if a PEs local tasks are done, it will repeat sending a request  $R$  to other randomly determined PEs until  $R$  is not rejected. The “polled” PE will split its remaining tasks and reinitialize them asynchronously as some are sent to the PE that initiated  $R$ .

A discrete events model for the operation of a slave PE performing the RP-DLB mesh refinement is shown in Fig. 3; it is constituted of five states and eight transitions. The slave PE starts with tasks assigned by the master PE, and subsequently its state changes over the transitions which represent the occurrence of the possible events. This model will be analyzed using the Petri Nets formalism in next subsection.

2) *Performance Measurement Parameters*: Let  $T_i^{(k)}$  and  $O_i^{(k)}$  represent the quantity of tetrahedra and octahedra produced, respectively, in iteration  $i$  by PE  $P_k$ . In Fig. 1(a) and (b)

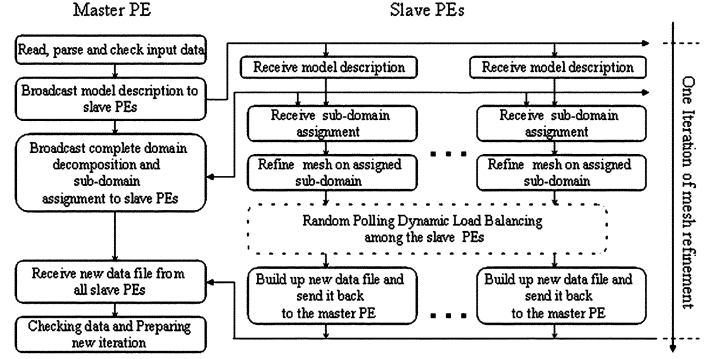


Fig. 2. Conceptual outline of parallel mesh refinement with RP-DLB.

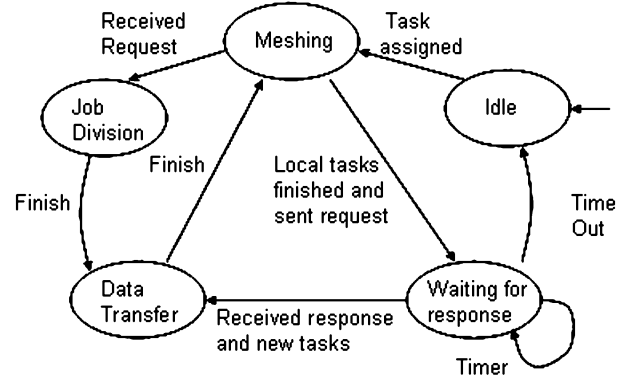


Fig. 3. Discrete events chart for a slave PE.

subdivisions, for iteration  $i$  each tetrahedron of iteration  $i - 1$  can be subdivided into 4 smaller tetrahedra and 1 octahedron, and each octahedron of iteration  $i - 1$  can be subdivided into 8 tetrahedra and 6 smaller octahedra. Thus, we have

$$T_i^{(k)} = 4T_{i-1}^{(k)} + 8O_{i-1}^{(k)} \quad (1)$$

$$O_i^{(k)} = T_{i-1}^{(k)} + 6O_{i-1}^{(k)}. \quad (2)$$

In any iteration of the mesh refinement, if the matrix assembly is required, all the octahedra in the element list will be subdivided into four tetrahedra each as in Fig. 1(c), and in this case

$$T_i^{(k)} = T_{i-1}^{(k)} + 4O_{i-1}^{(k)} \quad (3)$$

$$O_i^{(k)} = 0. \quad (4)$$

Let  $t_{\text{part}}$  and  $o_{\text{part}}$  be the time required for a tetrahedron or octahedron subdivision, as shown in Fig. 1(a) and (b), respectively. For iteration  $i$ , the computation time  $t_i^{\text{comp}}$  and communication time  $t_i^{\text{comm}}$  for  $P_k$  are given by (5) and (6), respectively. Here  $t_{\text{startup}}$  represents the message startup time and  $t_{\text{data}}$  is the transmission time to send the data for one element. These parameters can be adjusted to simulate their effect on the parallel computing environment performance. For  $n$  PEs in a master-slave model, there will be  $n - 1$  slave PEs in charge of  $n - 1$  subdomains. The time  $t_i$  for the slave PEs to complete the mesh refinement for iteration  $i$  satisfies (7), (since the PEs start the refinement iteration  $i$  at the same time in our model) [11]. The proof is shown in Fig. 4

$$t_i^{\text{comp}}(p_k) = 5T_{i-1}^{(k)} \cdot t_{\text{part}} + 14O_{i-1}^{(k)} \cdot o_{\text{part}} \quad (5)$$

$$t_i^{\text{comm}}(p_k) = t_{\text{startup}} + (T_i^{(k)} + O_i^{(k)}) \cdot t_{\text{data}} \quad (6)$$

$$t_i \leq \max(t_i^{\text{comp}}(p_k)) + \sum_{k=1}^{n-1} (t_i^{\text{comm}}(p_k)). \quad (7)$$

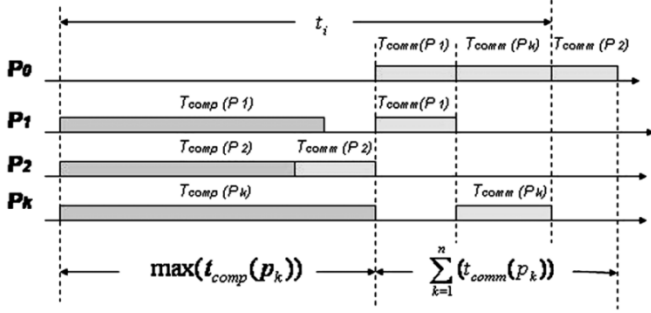


Fig. 4. Timing diagram for proof of (7).

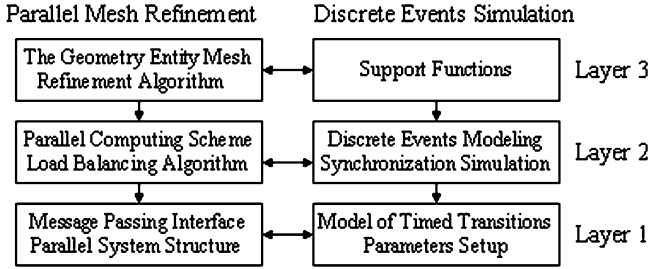


Fig. 5. Framework for mesh refinement Petri Nets model.

A standard performance measure is the parallel efficiency  $E$

$$E = \frac{w}{n \cdot t_{\text{parallel}}(n, w)} \quad (8)$$

where  $w$  is the size of the mesh refinement process measured in units of sequential execution time and  $t_{\text{parallel}}(n, w)$  is the parallel execution time for a system consisting of  $n$  identical PEs [4]. In this work, we examine the performance of our approach both with respect to parallel speedup and efficiency.

### B. Modeling Framework

Our Petri Nets model is a specification of the parallel mesh refinement system with RP-DLB in terms of a set of states and events. Performance simulation involves modeling the occurrence of events as they evolve in time and recognizing their effects as represented by transitions of states during the parallel mesh refinement process [7], [8]. We map the algorithm with the supporting formula (1)–(7) into the Petri Nets model by modifying the parameters related to the states, events and transition delays.

The parallel mesh refinement process can be conceived at three levels, each corresponding to one layer of our Petri Nets model, as shown in Fig. 5. The first layer comprises the parallel computing environment module. The parameters involved in this layer are the communication timing delay and computation cost. The estimated values of the parameters are derived from the system implementation (we use the MPI of SUN HPC5.0). The corresponding part in the Petri Nets model is the timed transitions. Layer 2 is the parallelization and load balancing algorithm module. The specification of the processor interactions and the DLB schemes is included in this layer. In the Petri Nets model, this part is the real-time discrete events model that underlies the dynamic behavior of the DLB algorithm. Layer 3 is the application layer. The geometrical properties of the tetrahedral and octahedral subdivisions are specified in this module. On the right, for the same layer of the Petri Nets model, we use the transition-arc-weight to model the geometrical mesh refinement scheme. The details of the model are provided below.

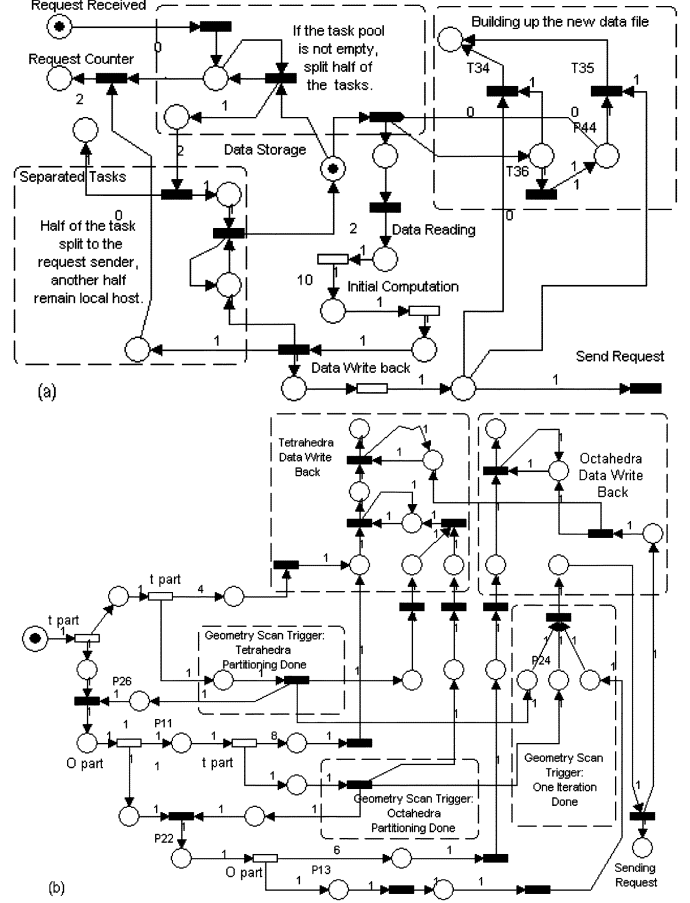


Fig. 6. Petri Nets Module. (a) RP-DLB task division. (b) Tetrahedron and octahedron subdivision.

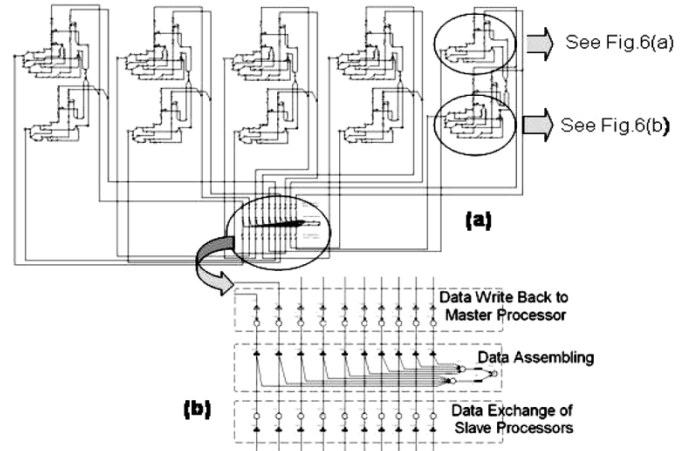


Fig. 7. Petri Nets model of parallel mesh refinement in a 6 PE system. (a) Overall model structure. (b) Mesh refinement in slave PE.

### C. Petri Nets Model

1) *RP-DLB Protocol:* Fig. 6(a) shows the Petri-Nets model of the RP-DLB protocol: in a PE, tasks to be executed are stored in Data\_Storage, while the geometric computations are being processed. When a request  $R$  arrives, the PE will send half of the tasks in Data\_Storage to the sender of  $R$ . When the PE has executed all the tasks in Data Storage, it will send its request  $R$  to another randomly selected PE and simultaneously start to build up a new data file for its mesh refinement results.

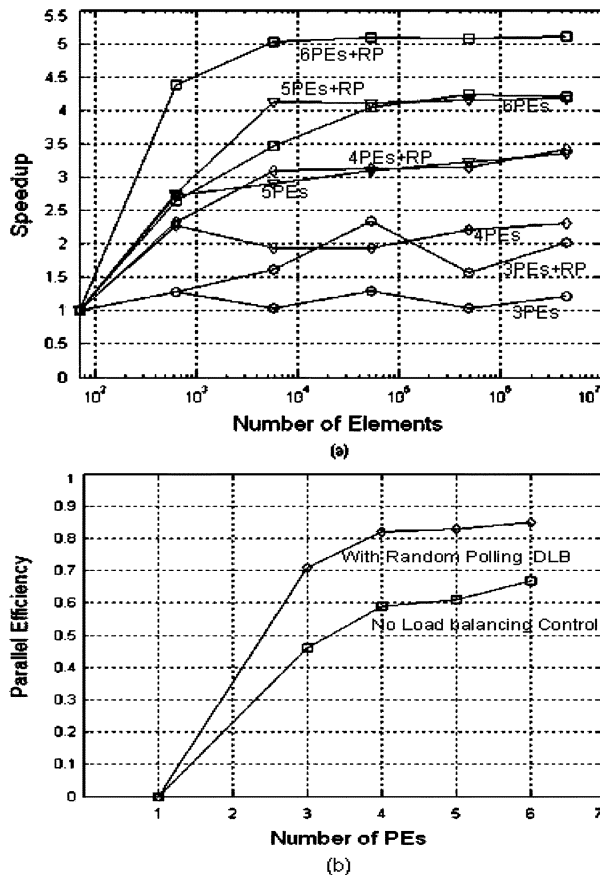


Fig. 8. RP-DLB performance results. (a) Speedup versus number of elements for different numbers of PEs. (b) Parallel efficiency versus number of PEs.

2) *Meshing Computation*: The Petri Nets model in Fig. 6(b) shows the procedure of tetrahedral and octahedral subdivision: this starts with a scan of tetrahedral/octahedral entities; next the refinement rule is applied to each individual tetrahedron/octahedron. Once an individual element is processed, a signal is generated by scan trigger [Fig. 6(b)] for loading the next geometrical entity.

3) *Parallel System Model*: The overall model we developed is shown schematically in Fig. 7. It involves six modules representing one master and five slave PEs that we have in a symmetric multiprocessor. The communication costs are defined by transitions that connect the PEs in the system, as shown in the figure. The system parameters  $t_{part}$ ,  $o_{part}$ ,  $t_{data}$ , and  $t_{startup}$ , are defined in the transition delays in each stage of the mesh refinement model.

#### IV. RESULTS

The efficiency of a RP-DLB algorithm specifically designed for the mesh refinement model described in this work is examined in this section. Performance results for the RP-DLB parallel mesh refinement simulation are shown in Fig. 8. It may be noted from Fig. 8(a), that the parallel speedup for different numbers of PEs differs with increasing problem size, as the mesh refinement progresses. In each case, RP-DLB improves the performance compared to the same number of PEs without DLB. Furthermore, it can benefit the system by saving PEs: e.g., 5 PEs with RP is as good, or better, than 6 PEs without RP. Fig. 8(b)

shows the parallel efficiency versus the number of PEs in the system. The results are based on the mean speedups observed over the entire range of the number of elements produced during the refinement procedure. Clearly, the parallel efficiency of the new RP-DLB mesh refinement model is better than without load balancing. In addition, note that the parallel efficiency increases as the number of PEs increases up to the 6 PEs in our model.

#### V. CONCLUSION

A new approach for the modeling, analysis and design of DLB algorithms in parallel finite element mesh refinement that utilizes Petri Nets has been proposed and evaluated. This new simulation-based approach allows for a relatively detailed description of a system and can reveal key performance characteristics. The results for the 3-D parallel mesh refinement model considered demonstrate the benefits of the new approach for developing RP-DLB algorithms for the target parallel architecture. Future work should include: validating the Petri Nets model by comparing the simulation performance with real codes; further performance optimization of the new RP-DLB algorithm for systems of heterogeneous multiprocessors; and through a more detailed communication cost analysis. Finally, the new modeling approach may be extended to other aspects of the FEM, such as matrix assembly and solution methods.

#### ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the Fonds Québécois de la Recherche sur la Nature et les Technologies.

#### REFERENCES

- [1] P. J. Frey and P. L. George, *Mesh Generation: Application to Finite Elements*. Oxford, U.K.: Hermes Science Publishing, 2000.
- [2] L. Oliker, R. Biswas, and H. N. Gabow, "Parallel tetrahedral mesh adaptation with dynamic load balancing," *Parallel Comput.*, vol. 26, no. 12, pp. 1583–1608, Nov. 2000.
- [3] B. H. V. Topping, J. Muylle, P. Iványi, R. Putanowicz, and B. Cheng, *Finite Element Mesh Generation*. Kippen, U.K.: Saxe-Coburg, 2004.
- [4] P. Sanders, "Asynchronous random polling dynamic load balancing," in *Proc. Dig. 10th Int. Symp. Algorithms Comput.*, 1999, pp. 37–48.
- [5] K. Vipin, Y. G. Ananth, and R. Venpaty, "Scalable load balancing techniques for parallel computers," *J. Parallel Distrib. Comput.*, vol. 22, pp. 60–79, Jul. 1994.
- [6] D. Q. Ren and D. D. Giannacopoulos, "A Preliminary approach to simulate parallel mesh refinement with petri nets for 3-D finite element electromagnetics," in *Proc. ANTEM*, 2004, pp. 127–130.
- [7] C. Girault and R. Valk, *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*. Berlin, Germany: Springer-Verlag, 2002.
- [8] C. Lindemann, *Performance Modeling with Deterministic and Stochastic Petri Nets*. New York: Wiley, 1998.
- [9] G. Greiner and R. Grosso, "Hierarchical tetrahedral-octahedral subdivision for volume visualization," *Vis. Comput.*, vol. 16, pp. 357–369, Oct. 2000.
- [10] M. Dorica and D. Giannacopoulos, "Impact of mesh quality improvement systems on the accuracy of adaptive finite element electromagnetics," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1692–1695, May 2005.
- [11] A. Grama, A. Gupta, G. Karypis, and V. Kumar, *Introduction to Parallel Computing*, 2nd ed. Reading, MA: Addison-Wesley, 2003.