Alternate Parallel Processing Approach for FEM

David M. Fernández, Maryam Mehri Dehnavi, Warren J. Gross and Dennis Giannacopoulos
Department of Electrical and Computer Engineering, McGill University
3480 University Street, Montreal (QC), H3A 2A7, Canada
Email: david.fernandezbecerra@mail.mcgill.ca, maryam.mehridehnavi@mail.mcgill.ca, warren.gross@mcgill.ca,
dennis.giannacopoulos@mcgill.ca

***Abstract*—In this work we present a new alternate way to formulate the finite element method (FEM) for parallel processing based on the solution of single mesh elements. The key idea is to decouple the solution of a single element from that of the whole mesh, thus exposing parallelism at the element level. Individual element solutions are then superimposed node-wise using a weighted sum over concurrent nodes. A classic 2D electrostatic problem is used to validate the proposed method obtaining accurate results. The original mesh was refined to study the iterations scaling behavior, which proved to grow linearly with the number of elements. A 13 times speedup was observed for a GPU implementation over a sequential CPU version.**

I. Introduction

Solving increasingly complex electromagnetic (EM) problems using modern computing resources inevitably requires employing parallel programming paradigms in response to the current trend of advances in microprocessor architecture. The advent of the multicore/manycore processors brings about an important turning point in programming practices; in particular, for EM practitioners and the scientific community in general this translates to rewriting legacy libraries and applications in parallel terms to efficiently realize the performance benefits offered by these modern computing resources as shown recently in [1], [2]. This work focuses on the finite element method (FEM), a popular numerical simulation technique, and proposes an alternate way to solving the linear systems derived that is well suited for parallel manycore implementations.

II. New FEM Single Element Solution Method

The classic FEM formulation can be thought of as a seven step process as shown in Fig. 1. Traditionally, the solution of FEM has been parallelized in three ways: a) partitioning and solving in parallel the derived algebraic system [1]–[3]; b) employing domain decomposition techniques [3]–[5]; and c) using multigrid techniques [5]. However, a greater amount of parallelism is sought to take advantage of the aforementioned manycore trend. Thereof, we propose to decouple the element solution from that of the whole mesh by directly computing on the element stiffness matrices concurrently going from step three to five in Fig. 1, each subject to boundary conditions. Such disconnected solutions are then averaged node-wise using a weighted sum over all concurrent nodes in an iterative fashion until convergence is achieved. Furthermore, this approach does not require building a global coefficient matrix skipping step four in Fig.1. A similar approach was proposed in [6] where the solution is computed by nodes. The mathematical formulation for the proposed decoupled single element solution (FEM-SES) approach is presented next.

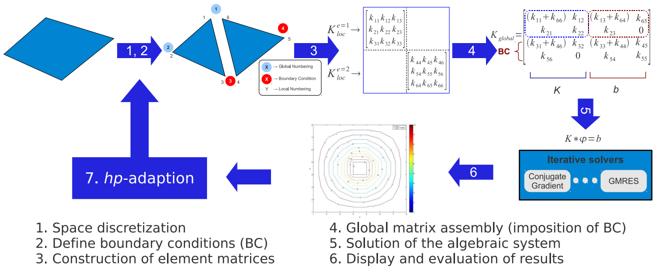

Fig. 1. Steps in the classic finite element method (FEM).

A. Mathematical Formulation

Equations (1-3) present the classic FEM variational formulation for a static EM boundary value problem that will be used for simplicity, without loss of generality. Here $F(\varphi)$ represents the functional to minimize, φ the unknowns and p the boundary conditions (BC) applied.

$$\delta F(\varphi) = 0 \tag{1}$$

$$\varphi = p, \text{ on the boundary } \Gamma \tag{2}$$

$$F(\varphi) = \frac{1}{2}\iint\limits_{\Omega}\left[\left(\frac{\partial\varphi}{\partial x}\right)^2 + \left(\frac{\partial\varphi}{\partial y}\right)^2\right] d\Omega \tag{3}$$

The functional can then be applied to each element in the discretized domain as shown in (4-5) where the superscript e refers to the element index.

$$F(\varphi) = \sum_{e=1}^{n} F^e(\varphi^e) \tag{4}$$

$$F^e(\varphi^e) = \frac{1}{2}\iint\limits_{\Omega_e}\left[\left(\frac{\partial\varphi^e}{\partial x}\right)^2 + \left(\frac{\partial\varphi^e}{\partial y}\right)^2\right] d\Omega \tag{5}$$

Next the local functionals are minimized and BCs are enforced element-wise independently, see equation (6). This is where the new method departs from the classic FEM.

$$\left\{\frac{\partial F^e}{\partial\varphi^e}\right\}_{BC\_reduced} = \{K^e\}\{\varphi^e\} - \{b^e\} = \{\emptyset\} \tag{6}$$

To obtain the global solution from (6) a *2-step iterative relaxation* approach is proposed. As presented in Fig.2, the first step updates the local element solutions independently using a relaxation method and the second step couples the local solutions using a weighted average to produce the global

Step 1: Update local estimate using a given relaxation criteria (e.g. Jacobi: $x^{new}=Mx^{old}+b$)

For elements with 1 BC:

$$\begin{bmatrix}\varphi_1^{new}\\ \varphi_2^{new}\end{bmatrix}=\begin{bmatrix}1/k_{11} & 0\\ 0 & 1/k_{22}\end{bmatrix}\times\left\{\begin{bmatrix}b_1-k_{13}\boxed{\varphi_3^{p}}\\ b_2-k_{23}\boxed{\varphi_3^{p}}\end{bmatrix}-\begin{bmatrix}0 & k_{12}\\ k_{21} & 0\end{bmatrix}\begin{bmatrix}\varphi_1^{prev}\\ \varphi_2^{prev}\end{bmatrix}\right\} \quad (9)$$

For elements with 2 BC (not computed iteratively):

$$\left[\varphi_6^{new}\right]=\left[1/k_{66}\right]\times\left[b_6-(k_{64}\boxed{\varphi_4^{p}}+k_{65}\boxed{\varphi_5^{p}})\right] \quad (10)$$

For interior elements (no boundary conditions):

$$\begin{bmatrix}\varphi_1^{new}\\ \varphi_2^{new}\\ \varphi_2^{new}\end{bmatrix}=\begin{bmatrix}1/k_{11} & 0 & 0\\ 0 & 1/k_{22} & 0\\ 0 & 0 & 1/k_{33}\end{bmatrix}\times\left\{-\begin{bmatrix}0 & k_{12} & k_{13}\\ k_{21} & 0 & k_{23}\\ k_{31} & k_{32} & 0\end{bmatrix}\begin{bmatrix}\varphi_1^{prev}\\ \varphi_2^{prev}\\ \varphi_3^{prev}\end{bmatrix}\right\} \quad (11)$$

Step 2: Couple local solutions using weighted average enforcing continuity

$$\begin{bmatrix}\varphi_1^{new}\\ \varphi_2^{new}\end{bmatrix}_{global}=[C]\begin{bmatrix}\varphi_1^{new}\\ \varphi_2^{new}\\ \varphi_6^{new}\end{bmatrix}_{local} \quad (12)$$

C: modified connectivity information (using a weighted average)

$$\varphi_{1(global)}^{new}=\varphi_{1(local)}^{new}\left(\frac{k_{11}}{k_{11}+k_{66}}\right)+\varphi_{6(local)}^{new}\left(\frac{k_{66}}{k_{11}+k_{66}}\right) \quad (13)$$

Check for convergence to desired precision.
→ *If* (converged): **exit**.
→ *else:* **repeat step 1**.

Fig. 2. *2-step iterative relaxation* method: Step-1 shows update rationale for elements with or without BCs and Step-2 shows a weighted average example.

iterate. Finally, a convergence check is performed to either exit or repeat the process.

B. Sources of Parallelism, Advantages and Disadvantage

Sources of parallelism identified in the new approach are:

- Element stiffness matrices can be built in parallel and preserved in distributed CPU/cores to be computed later.
- Elements solutions may be computed in parallel independently of any other element.

Two drawbacks of the proposed *2-step iterative relaxation* method can be identified: a) it will have slow convergence similar to that of Jacobi iterative method, but a great deal of parallelism is obtained in exchange; and b) computing the global solution requires a single synchronization per iteration. Among other advantages, the proposed FEM-SES method does not require special numbering, no global coefficient matrix is built, uses the same information as the classic FEM, and good scaling is expected considering that the element connectivity is almost constant as the problem dimensions grows.

III. Results and Concluding Remarks

A 2D electrostatic coaxial cable problem (see Fig. 3) was implemented to validate the new method and study its convergence behavior. Tests were conducted on an 2.4GHz Intel Core2 Quad processor, with 4GB of global memory and running 64-bit Linux system. First, sequential implementations for both traditional FEM and the proposed *2-step iterative relaxation* method were done. Fig. 3 compares the FEM energy results with those of the new FEM-SES method demonstrating good agreement of the results for different number of unknowns; thus proving the validity of the the new method.

Next, the original mesh was refined to empirically study the convergence scaling of the proposed method. The dotted line in Fig. 4 represents a reference linear scaling (1:1 slope) and the solid line shows the iteration count results. These results empirically prove a sub-linear iteration scaling of the proposed FEM-SES method as the number of unknowns increases, which is a desirable scaling property of iterative methods.

Considering that recently manycore (i.e. graphic processing units-GPUs) processors have become an important computing resource available in almost all computing systems, which have demonstrated significant speedup of important scientific kernels [2], a straight forward GPU version of the proposed method was implemented using an NVIDIA GT 8800, an early generation CUDA [7] enabled GPU. This GPU has 14 streaming processors (SM) each with 8 scalar processors clocked at 1.5GHz and 512MB of global memory. A speedup of up to 13 was obtained with respect to a sequential version on the CPU referred above demonstrating the potential of the proposed method for parallel manycore computing.

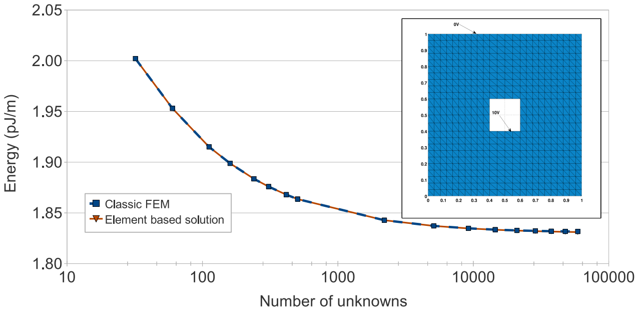

Fig. 3. Energy comparison for the classic FEM and the proposed FEM-SES.

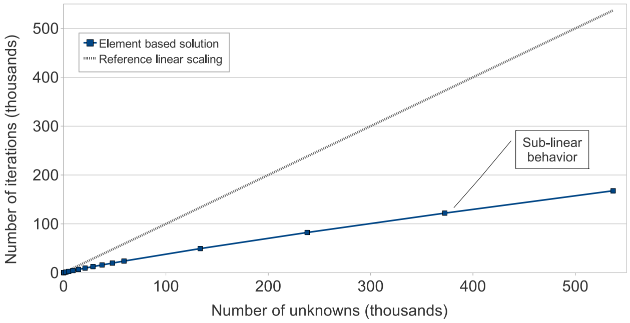

Fig. 4. Iterations scaling with the increase of problem size.

This work presents a new element-based solution approach for FEM called FEM-SES, that uses the same information as the classic FEM, while exposing low level parallelism well suited for modern manycore-GPU processors. The results shown prove the validity of the method for electrostatic boundary value EM problems and its potential for manycore parallelism with up to 13 speedup of a GPU implementation over a sequential CPU implementation. In the extended work both sequential and GPU implementation details will be given as well as further optimized results and comparisons.

References

[1] D. Fernández, D. Giannacopoulos and W. J. Gross, “Multicore acceleration of CG algorithms using blocked-pipeline-matching techniques,” *IEEE Trans. on Mag.*, vol.46, no.8, pp. 3057–3060, 2010.

[2] M. Mehri Dehnavi, D. Fernández and D. Giannacopoulos, “Finite element sparse matrix vector multiplication on GPUs,” *IEEE Trans. on Mag.*, vol.46, no.8, pp. 2982–2985, 2010.

[3] T. Itoh, et. al., *Finite Element Software for Microwave Engineering*, Wiley-Interscience, New York, 1996, pp. 385-400.

[4] A. Takei, et. al., “Full Wave Analyses of Electromagnetic Fields With an Iterative Domain Decomposition Method,” *IEEE Trans. on Mag.*, vol.46, no.8, pp. 2860–2863, 2010.

[5] L. Yuanqing and Y. Jiansheng, “A finite element domain decomposition combined with algebraic multigrid method for large-scale electromagnetic field computation,” *IEEE Trans. on Mag.*, vol.42, no.4, pp. 655–658, 2006.

[6] J. P. A. Bastos and N. Sadowski, “A new method to solve 3-d magnetodynamic problems without assembling an Ax=b system,” *IEEE Trans. on Mag.*, vol. 46 no. 8, pp. 3365–3368, 2010.

[7] NVIDIA CUDA, http://developer.nvidia.com/object/cuda.html.