Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Physics Communications 174 (2006) 948–960

Computer Physics Communications

www.elsevier.com/locate/cpc

Development of a parallel Poisson's equation solver with adaptive mesh refinement and its application in field emission prediction

K.-H. Hsu, P.-Y. Chen, C.-T. Hung, L.-H. Chen, J.-S. Wu *

Department of Mechanical Engineering, National Chiao-Tung University, Hsinchu 30050, Taiwan

Received 2 November 2005; received in revised form 18 January 2006; accepted 19 January 2006

Available online 13 March 2006

Abstract

A parallel electrostatic Poisson's equation solver coupled with parallel adaptive mesh refinement (PAMR) is developed in this paper. The three-dimensional Poisson's equation is discretized using the Galerkin finite element method using a tetrahedral mesh. The resulting matrix equation is then solved through the parallel conjugate gradient method using the non-overlapping subdomain-by-subdomain scheme. A PAMR module is coupled with this parallel Poisson's equation solver to adaptively refine the mesh where the variation of potentials is large. The parallel performance of the parallel Poisson's equation is studied by simulating the potential distribution of a CNT-based triode-type field emitter. Results with ∼100 000 nodes show that a parallel efficiency of 84.2% is achieved in 32 processors of a PC-cluster system. The field emission properties of a single CNT triode- and tetrode-type field emitter in a periodic cell are computed to demonstrate their potential application in field emission prediction.

Keywords: Parallel Poisson's equation; Galerkin finite element method; Parallel adaptive mesh refinement; Field emission

1. Introduction

Field emission display (FED) has attracted tremendous attention in the past decade [1–6]. The advantages of applying FED in display technology include lower driving voltage, higher lighting frequency, and possibly, better display resolution. From the Fowler–Nordheim law [7], the magnitude of the electron flux emitted from the surface depends upon the local electric field at the surface and the work function of the solid. In addition to finding materials with lower work functions, enhancing the local electric field near the surface is one of the most critical tasks in improving field emission properties. As a trial-and-error method is often expensive in terms of time and cost, a computer simulation may speed up the design process by revealing the detailed physics with the FED. In practice, the geometry of the field emitter and the gates involved in the FED design is three dimensional and often very complicated [8–10]. Thus, it is important to develop a simulation tool which is accurate, fast, and capable of handling complicated geometry for predicting the distribution of the electrical field around the emitters. In the current study, we intend to present a simulation tool using the finite-element method that has the above-mentioned important features for field emission prediction.

In the past, several numerical studies have been conducted for the prediction of field emission properties, e.g., [11–16]. Most of these studies use either the 2D or 3D finite difference method [12–16], or the 2D finite element approach [11] for discretizing the electrostatic Poisson's equation. As mentioned earlier, a practical FED design often involves three-dimensional objects with a complicated geometry, rendering the use of the finite-difference method as very difficult or unsuitable. The finite-element or finite-volume method using unstructured grids should represent the best choice for the numerical method in this regard. In addition, parallel processing can be necessary in simulating the practical three-dimensional design of field emitters or when including space-charged effect with high emission currents in the particle-in-cell (PIC) method [14–16]. Otherwise, in Refs. [14–16], the computational time for a typical run to emit only a few electrons can

* Corresponding author. Tel.: +886 3 573 1693; fax: +886 3 572 0634.
E-mail address: chongsin@faculty.nctu.edu.tw (J.-S. Wu).

0010-4655/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2006.01.006

take up to one week. Also, the accuracy of the electron-flux prediction from the emitters strongly depends on the accuracy of the local electrical field at the surface, which makes the grid resolution at the surface a critical issue in the simulation. This concern necessitates the use of an adaptive mesh refinement to achieve accuracy in predicting the electrical field at the surface, which is the main concern of the current study.

In this study, we present a parallel three-dimensional Poisson's equation solver using the Galerkin finite-element method coupled with parallel adaptive mesh refinement using an unstructured tetrahedral mesh. In addition, the applications used to predict the field emission properties are demonstrated in this paper.

The finite-element method for modeling the Poisson's equation is described next, followed by a detailed description of the proposed parallel implementation. The general procedures of PAMR are then enumerated step by step. In addition, the coupling of this parallel Poisson's equation solver with a parallel adaptive mesh refinement is described. Then the results of the parallel performance of this solver and the coupling of PAMR are also discussed, respectively. Finally, the study is summarized with some important conclusions and recommendations for future researches.

2. Numerical method

2.1. Poisson's equation

Poisson's equation for the electrostatic distribution can be written as,

$$\nabla^2\phi = -\frac{\rho}{\varepsilon_0}, \tag{1}$$

where ϕ is the electric potential, ρ is the volume density of the free charges, and ε_0 is the permittivity of free space. Without considering the space charge effect caused by the emitted electrons around the emitters, Poisson's equation is reduced to a Laplacian equation. However, in the current study, we still discretize the Poisson's equation, since our interest lies in including the space charge effect using the particle-in-cell (PIC) [17] method for the prediction of field emission and low-temperature plasma in the very near future.

2.2. Finite-element discretization of Poisson's equation

By applying the Galerkin weighted residual method with the trial solution ($\phi^{(e)}$) assumption, the Poisson's equation is then discretized in an element as

$$\begin{bmatrix} K_{11}^{(e)} & K_{12}^{(e)} & \cdots & K_{1n}^{(e)} \\ K_{21}^{(e)} & K_{22}^{(e)} & \cdots & K_{2n}^{(e)} \\ \vdots & \vdots & \cdots & \vdots \\ K_{n1}^{(e)} & K_{n2}^{(e)} & \cdots & K_{nn}^{(e)} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} F_1^{(e)} \\ F_2^{(e)} \\ \vdots \\ F_n^{(e)} \end{bmatrix}, \tag{2}$$

where

$$K_{ij}^{(e)} = \iiint \left[\left(\frac{\partial N_i^{(e)}}{\partial x} \frac{\partial N_j^{(e)}}{\partial x} \right) + \left(\frac{\partial N_i^{(e)}}{\partial y} \frac{\partial N_j^{(e)}}{\partial y} \right) + \left(\frac{\partial N_i^{(e)}}{\partial z} \frac{\partial N_j^{(e)}}{\partial z} \right) \right] \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z,$$

$$F_i^{(e)} = \iiint \left(\frac{\rho}{\varepsilon_0} \right) N_i^{(e)}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z - \iint_A N_i^{(e)} \vec{\tau}_n^{(e)}\,\mathrm{d}A, \tag{3}$$

$$\phi^{(e)}(x,y,z) = \sum_{i=1}^{n} a_i N_i^{(e)}(x,y,z), \tag{4}$$

where $\vec{\tau}_n^{(e)}$ is the inward-normal flux to the element face. Note that $N_i^{(e)}(x,y,z)$ is the shape function, and $n = 4$ in Eq. (2) since we use a tetrahedral element in the current study. Employing the linear shape function with

$$N_i^{(e)}(x,y,z) = \frac{1}{6V_c}\left(a_i^{(e)} + b_i^{(e)}x + c_i^{(e)}y + d_i^{(e)}z\right), \quad i = 1\text{–}4, \tag{5}$$

where V_c is the element volume, and the coefficients of $a_i^{(e)}$, $b_i^{(e)}$, $c_i^{(e)}$, and $d_i^{(e)}$ can be easily determined from the definition of the shape function in the finite-element theory, in which $N_i^{(e)}$ is unity at the supporting node i and zero at other nodes in an element.

By assembling all element equations throughout the computational domain, the system matrix equation becomes,

$$K_{i,j}^{(s)} a_j = F_i^{(s)}, \quad i, j = 1, \ldots, n, \tag{6}$$

where $K_{i,j}^{(s)}$ is the coefficient matrix, and $F_i^{(s)}$ is the loading vector. This system matrix equation is then solved using the conjugate gradient (CG) method with nonzero entries stored in a compressed sparse row (CSR) format that is computationally efficient both in storage and matrix operations [18]. In the CSR format, a one-dimensional primary array is used to store nonzero entries only, and another two indexing arrays are constructed to indirectly address the nonzero entries in the primary array to the indices of the coefficient matrix. This CSR format is very efficient in matrix-by-vector product operations, which is the most time-consuming part for the most iterative scheme.

2.3. Parallel implementation of Poisson's equation solver

In the current parallel FEM for the Poisson's equation, a geometrical non-overlapping subdomain-by-subdomain (SBS) method is used [18]. The global coefficient matrix is stored as a partitioned matrix, and the dominant matrix-by-vector product and inner product of two vector operations of the coefficient matrix in the conjugate gradient method are performed on the SBS basis. In the SBS method, we first decompose the computational domain Ω into p non-overlapping subdomains as

$$\Omega = \bigcup_{i=1}^{p} \Omega_i \tag{7}$$

and

$$\Omega_i \cap \Omega_j = \{\} \quad \text{when } i \neq j. \tag{8}$$

Since domain Ω has been partitioned, the unknowns, $\bar{a}$, are ordered in the following manner. Each of the interior vertices in Ω_1 (e.g., $\bar{a}_1$) is followed by each of the interior vertices in Ω_2 (e.g., $\bar{a}_2$), etc. up to each of the interior vertices in Ω_p (e.g., $\bar{a}_p$). It then follows that the system equation which may be expressed in a block matrix is formed as

$$\begin{bmatrix} \bar{\bar{K}}_1 & & & & \bar{\bar{B}}_1 \\ & \bar{\bar{K}}_2 & & & \bar{\bar{B}}_2 \\ & & \ddots & & \vdots \\ & & & \bar{\bar{K}}_p & \bar{\bar{B}}_p \\ \bar{\bar{B}}_1^{\mathrm{T}} & \bar{\bar{B}}_2^{\mathrm{T}} & \cdots & \bar{\bar{B}}_p^{\mathrm{T}} & \bar{\bar{K}}_s \end{bmatrix} \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \vdots \\ \bar{a}_p \\ \bar{a}_s \end{bmatrix} = \begin{bmatrix} \bar{F}_1 \\ \bar{F}_2 \\ \vdots \\ \bar{F}_p \\ \bar{F}_s \end{bmatrix}. \tag{9}$$

In Eq. (9), each of the submatrices in the block-arrowhead structure of the coefficient matrix stems from the non-overlapping domain decomposition. The formation of this block-arrowhead structure of the coefficient matrix depends on how we number the nodes in the current study. In each subdomain, we number the internal nodes first followed by the interprocessor nodes.

During the process of element assembly, both the blocks $\bar{\bar{K}}_i$ and $\bar{F}_i$ have contributions only from the internal nodes in each processor i. For each processor i, the blocks $\bar{\bar{B}}_i$ and $\bar{\bar{B}}_i^{\mathrm{T}}$ have contributions from the internal nodes to the interprocessor nodes and vice versa. Both the blocks $\bar{\bar{K}}_s$ and $\bar{F}_s$ have contributions only from all of the interprocessor nodes and require communication among processors for element assembly. Once each processor has concurrently assembled each of the blocks $\bar{\bar{B}}_i$, $\bar{\bar{B}}_i^{\mathrm{T}}$, $\bar{\bar{K}}_i$, and $\bar{F}_i$, the system equation (9) is then stored in a distributed manner. It is then ready to be solved by *parallel CG*, the details of which can be found in Saad's book [18] but are skipped here for brevity. In the current study, the convergence criterion of the relative residual in parallel CG is 10^{-7}, unless otherwise specified. In addition, a parallel multilevel graph-partitioning library [19] is used to partition the computational domain whenever necessary since the unstructured mesh is adopted in the current study.

2.4. *Parallel adaptive mesh refinement (PAMR)*

Fig. 1 shows the proposed overall procedures of parallel adaptive mesh refinement for an unstructured tetrahedral mesh. Only the general procedures are described in this paper, while the details and results of the parallel implementation can be found elsewhere [20]. Basically, the parallel mesh refinement procedures in Fig. 1 are similar to those presented earlier for serial mesh refinement in this journal [21]. In the serial mesh refinement, the cells are first examined to identify if cell refinement is necessary. If so, then they are refined "isotropically" into eight child cells. The generated hanging nodes are then removed following the procedures proposed in Wu et al. [21] in which the cells are further refined into two, four, or eight child cells.

However, the detailed procedures and related data structure become more complicated than those in serial mesh refinement because of the parallel processing. Domain decomposition is also used in line with parallel implementation of the current Poisson's equation solver. Each spatial subdomain belongs to a specific processor in practice. The overall procedure shown in Fig. 1 can be summarized as follows:

1. Preprocess the input data at the host processor, and distribute them to all other processors.
2. Index the cells which require refinement based on the refinement criteria. In the current study, we use the variation of potentials among elements as the criterion for cell refinement which, in practice, is equivalent to a generally accepted error estimator as will be shown in the next section.
3. Check if further mesh refinement is necessary. If it is, then proceed to the next step. If not, proceed to Step 9.
4. Add new nodes into those cells that require refinement.
   (a) Add new nodes onto all edges of isotropic cells.
   (b) Add new nodes into the anisotropic cells which require further refinement as decided upon in the following steps.
   (c) Communicate the hanging-node data to corresponding neighboring processor if the hanging nodes are located at IPB.
   (d) Remove the hanging nodes following the procedures as shown in Wu et al. [21]. The basic idea is to remove the hanging nodes for all kinds of conditions, and then refine the cell into two, four, or eight child cells.
5. Unify the global node and cell numberings caused by the newly added nodes in all processors.
   (a) Add up the number of the newly added nodes in each processor, excluding those located at interprocessor boundaries (IPBs).
   (b) Gather this number from all other processors, and add them up to obtain the updated total number of nodes, including old and new nodes, but excluding the newly added nodes at IPBs.
   (c) Build up the updated node-mapping and corresponding cell-mapping arrays for those newly added nodes in the interior part of each subdomain based on the results in Step 5(b).
   (d) Communicate the data of newly added nodes at the IPBs among all processors.
   (e) Build up the node-mapping array for the new nodes received at IPBs in each processor.
6. Build up new connectivity data for all cells to include the newly added nodes.
7. Build up the new neighbor-identifying array based on the new connectivity data obtained in Step 6.
   (a) Reset the neighbor-identifying array.
   (b) Build up the neighbor-identifying arrays for all cells based on the new connectivity data, excluding the data associated with the faces lying on the IPBs that require the updated information of the global cell number which is not yet known at this stage.
   (c) Record all the neighbor-identifying arrays that have not been rebuilt in Step 7(b).
   (d) Broadcast all the recorded data in all processors.
   (e) Build up the neighbor-identifying arrays on the IPBs, considering the overall connectivity data structure.

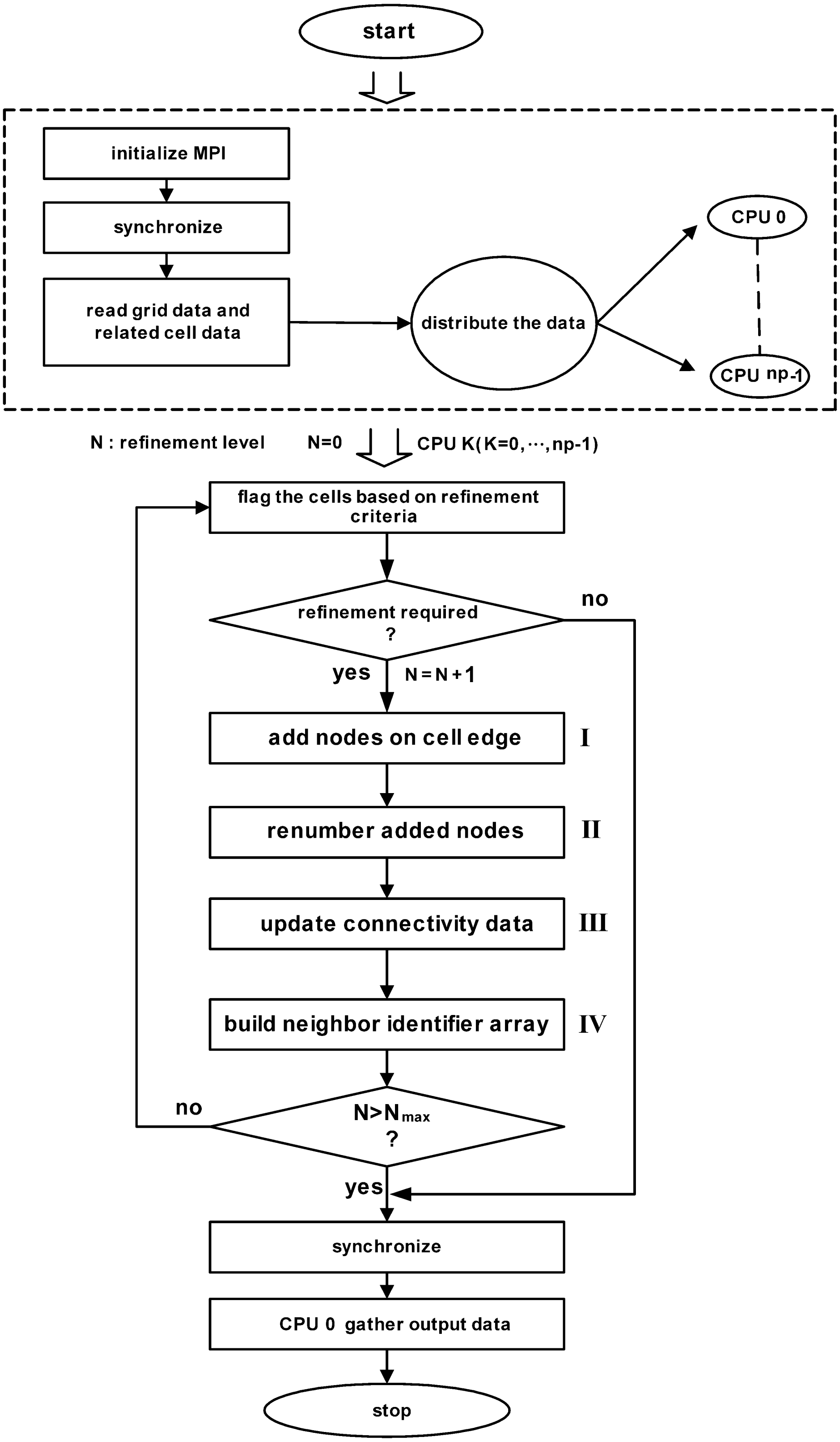


Fig. 1. Flowchart of the parallel mesh refinement module.

8. Decide if it reaches the preset maximum number of refinement. If it does, then proceed to the next step. Otherwise, return to Step 3.
9. Synchronize all processors.
10. The host processor gathers and outputs the data.

In the current study, by coupling the PAMR with the parallel Poisson's equation solver as stated in Step 3, the maximum number of refinement is set to be "one", since the option whether further refinement is necessary is decided outside the PAMR, as can be seen in the next section.

2.5. Coupling of PAMR with parallel Poisson's equation solver

The PAMR presented in the previous section can be easily coupled to the current parallel Poisson's equation solver since both utilize 3D unstructured tetrahedral mesh and MPI for data communication. One can readily wrap up the PAMR as a library and insert it into the source code of any parallel numerical solver to be used. However, some problems may occur due to memory conflicts between the inserted library and the numerical solver itself that could reduce the problem size one can handle in practice. As such, a simple coupling procedure, written in shell script (Fig. 2) that is standard on all Unix-like systems, can be prepared to link the PAMR and the current parallel Poisson's equation solver. In doing so, we can keep the source codes intact and without alterations. Indeed, it is especially justified if only a steady state of the physical problem is sought, in which normally only several times of mesh refinement is enough to have a fairly satisfactory solution. Thus, the total I/O time, which is in proportion to the number of couplings in switching between two codes, can be reduced to a minimum in practical applications. In addition, as shown in Fig. 2, after identifying those cells that require refinement before PAMR, the domain is repartitioned based on the new mesh refinement requirements. For example, the weight factors of the cells (vertex in graph theory) are set as eight for those cells which are flagged to be refined; otherwise, they are set as unity. With this distribution of weight factors as the input to ParMetis [19] (a graph-partitioning tool), an approximate (but rather good) load balancing can be achieved in the PAMR module. Then the PPES reads in the output refined mesh from the PAMR module and partitions the new mesh with equal weight factors for all cells, in which the workload is balanced in the PPES.

The current parallel Poisson's equation solver along with PAMR is implemented and tested on a PC-cluster system with the Linux OS at the National Center for High-Performance Computing in Taiwan (64-node, dual processor and 8 GB RAM per node). The standard message-passing interface (MPI) is used for data communication. It is thus expected that the current parallel code will be highly portable among the memory-distributed parallel machines that are running with the Linux (or its equivalent) operating system.

3. Results and discussions

3.1. Validation of the parallel Poisson's equation solver

Many analytical solutions of Poisson's equation are available for comparison either with or without the source term. In the current study, we have selected one problem without a source term and another with a constant source term. The former is a grounded conducting sphere with diameter (D_{sphere}) 2 meters immersed in a uniform electric field ($\vec{E} = 10$ volts/m, ~40 000 elements, 20 processors), while the latter is a uniformly charged distribution between two infinite, grounded conducting plates at $L = 0$ m and $L = 0.02$ m (quasi-1D, number density of singly-charged ions $= 10^{16}\ \text{m}^{-3}$, ~8500 elements, 20 processors). About ~56 000 particles are used. The charge weighting used in this is based on the volume coordinates which originated from the finite element method. The simulation and analytic solutions of these two problems are both in excellent agreement with the analytical solution as shown in Fig. 3(a) and (b), respectively. These results validate the accuracy of the current parallel Poisson's equation solver.

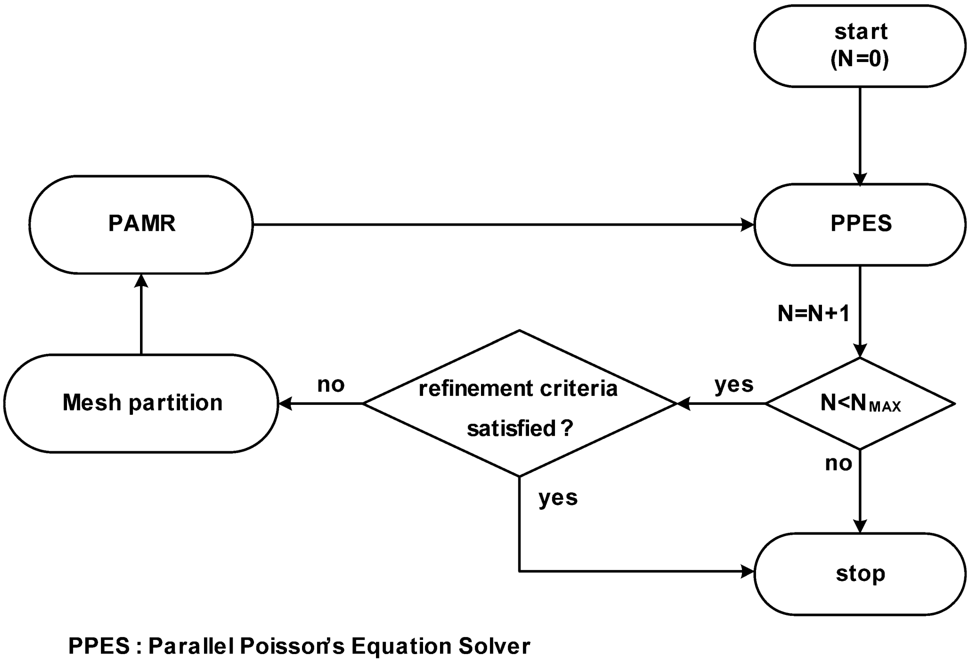
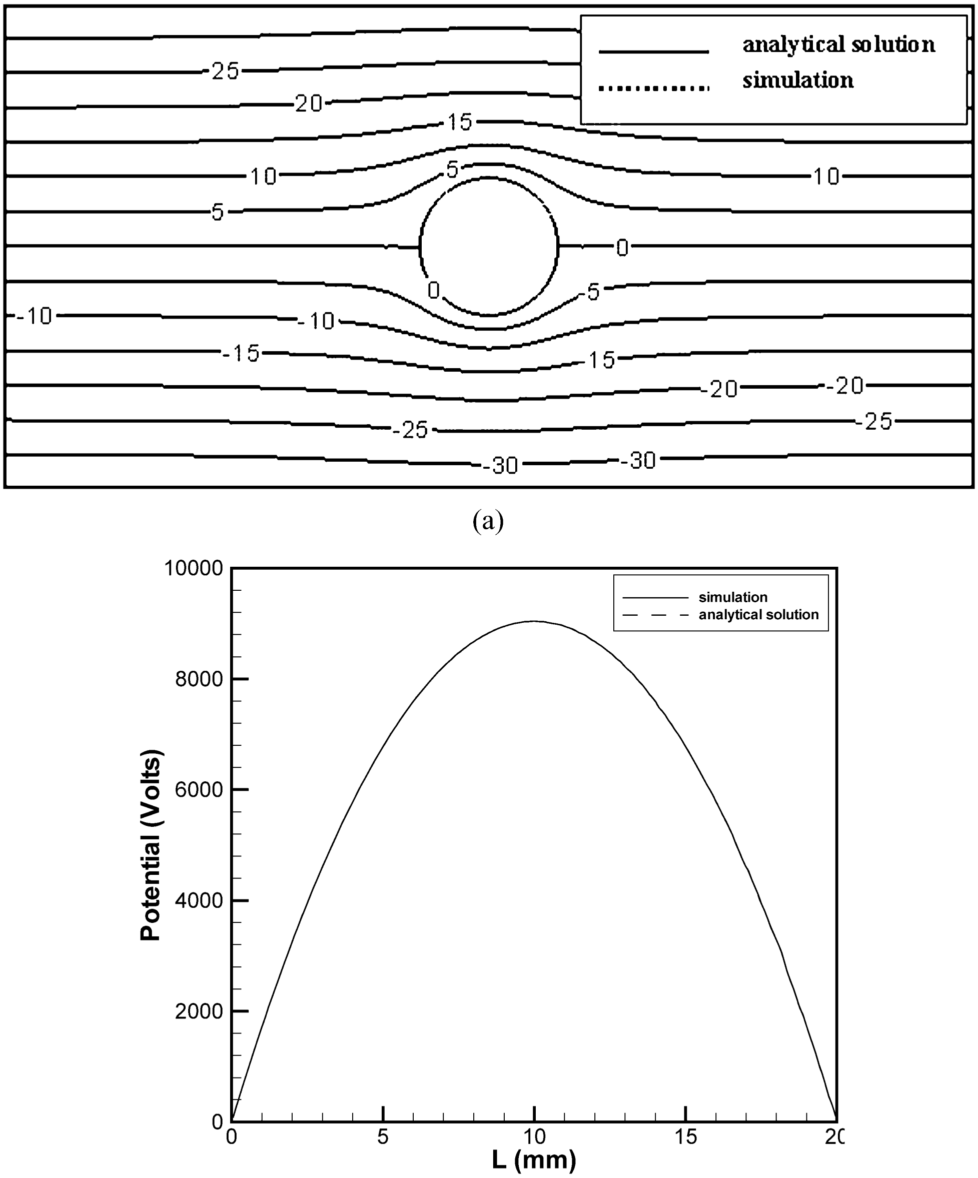


Fig. 2. Flowchart of the coupled PPES-PAMR method.



(a)

(b)

Fig. 3. Contours of the potential distribution of (a) a grounded conducting sphere immersed in a uniform electric field and (b) uniform positive charges distribution between two infinite grounded conducting planes.

3.2. Parallel performance of the Poisson's equation solver

The simulation of a typical single CNT field emitter within a periodic cell using 0.47 million elements (∼97 000 nodes), as shown in Fig. 4, is employed to test the parallel performance of the current Poisson's equation solver. This size of the mesh is typical for further production run as will be presented later. Only $\frac{1}{4}$ of the volume is used for the simulation by taking advantage of the symmetry in this problem. The gate voltage is applied with 150 volts, while the cathode and anode electrodes are grounded and applied with 400 volts, respectively. At the planes of symmetry, Neumann boundary conditions are used. A very refined grid (Fig. 5) is used near the silicon tip to improve the accuracy of the predicted electrical field. No parallel adaptive mesh refinement is used in the simulation since at this stage, we are only interested in obtaining the parallel performance of the Poisson's equation solver.

Fig. 6 illustrates the parallel speedup as a function of the number of processors up to 32. The corresponding time breakdown of various components of the solver along with speedup is summarized in Table 1. The runtime using a single processor is about 138.17 s, while it is reduced to 5.13 s using 32 processors, which results in ∼26.93 of parallel speedup. Most of the time is consumed in the parallel CG matrix solver, in which the percentage of communication time generally increases with the number of processors used. Note that the communication time, including the send/receive and all reduce commands required in a parallel CG solver, is relatively short (∼3.53 s or 4.5% of the total time) at 2 processors which is attributed to the fast access to the same memory by the dual-processor per node architecture

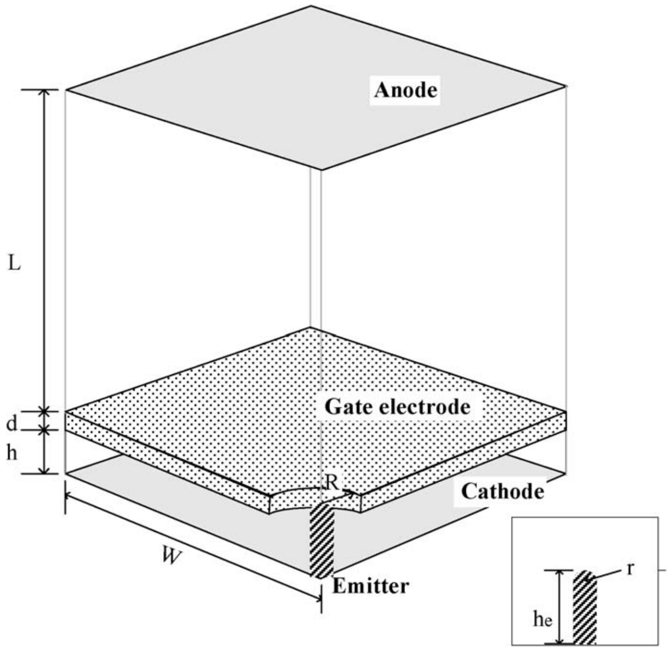


Fig. 4. Schematic diagram of the simulation domain for a typical CNT triode-type field emitter within a periodic cell. The important geometrical parameters are: $R = 500$ nm, $r = 10$ nm, $he = 600$ nm, $h = 500$ nm, $L = 49.3$ μm, $d = 200$ nm and $W = 25$ μm.

of this cluster system. An appreciable portion of the runtime is spent in the communication for a large number of processors, e.g., 35.4% at 16 processors. A further improvement of the solver efficiency by adding a robust parallel preconditioner before the parallel CG solver is highly expected and will be reported elsewhere in the future. Nevertheless, the present results clearly show that the parallel implementation of the Poisson's equation using a subdomain-by-subdomain procedure performs very well for the typical problem size we employ in the field emission prediction. A smaller problem size is not tested in the current study since it is irrelevant for this kind of application. It is expected that the parallel speedup can be even better if a larger problem size is simulated, e.g., for an array of field emitters. Thus, the current parallel implementation can greatly help to reduce the runtime required for the parametric study of optimizing the field emitter design.

3.3. Performance of parallel adaptive mesh refinement

A case with the same boundary conditions as the above test case for parallel performance is used to demonstrate the improvement of prediction using parallel adaptive mesh refinement. Fig. 7 shows a close-up view of the mesh distribution near the single CNT tip using PAMR where the initial mesh is rather coarse (7006 nodes), while the level-5 mesh is very fine (61 241 nodes) near the tip. In this case, an element is refined into eight child elements if the standard deviation of the potentials among the nodes of this element is larger than the value of a preset criterion, ε_{ref}. In this case, ε_{ref} is set to 0.08. Table 2 lists the number of nodes/elements and the corresponding maximal electric field in the simulation domain at different levels of mesh refinement. In addition, the data in the parentheses are obtained by using an *a posteriori* error estimator as proposed by Zienkiewicz and Chu [24]. We have employed a very simple gradient recovery scheme by averaging the cell values of the FE solution to extract the "exact" solution of the electric field in each cell. A prescribed global relative error ε_{pre} of 0.0003 is used to control the level of accuracy. The absolute error in each element is then compared with a current mean absolute error at each level, based on ε_{pre}, to decide if refinement is required. From Table 2, it is clear that the results are nearly the same by using either the variation of potential or the error estimator in the current study, although the implementation of variation of potential is more cost effective. For all the data presented in the present study, mesh refinement based on variation of potential is used throughout the study, unless otherwise specified.

Table 1
Time breakdown and speedup of the different number of processors

Processor no.	1	2	4	8	16	32
Total time (seconds)	138.17	79.17	42.53	14.78	8.21	5.13
CG solver time (%)	98.8	99.1	94.33	76.79	85.14	94.54
Matrix assembling time (%)	0.44	0.36	0.32	0.47	0.42	0.31
Communication time (%)	N/A	4.45	28.1	34.5	35.32	37
Speedup	1	1.74	3.25	9.35	16.83	26.93

Table 2
Evolution of simulation parameters at different levels of mesh refinement. (E_{MAX} is the local maximum electric field strength at the surface of CNT field emitter.)

Refinement level	Number of nodes	Number of elements	E_{MAX} (V/nm)
0	7006 (7006*)	27 814 (27 814)	8.218482 (8.21848)
1	22 750 (24 892)	110 218 (12 1064)	10.20636 (10.20257)
2	34 927 (38 896)	175 254 (196 378)	11.50804 (11.50135)
3	44 080 (47 984)	225 156 (245 975)	11.54894 (11.51166)
4	51 638 (55 488)	264 259 (284 766)	11.32366 (11.32647)
5	61 241 (59 279)	313 092 (306 368)	11.32303 (11.32665)
6	67 173	345 307	11.32324

* Numbers in the parentheses represent numerical data obtained using *a posteriori* error estimator with prescribed global relative error $\varepsilon_{\text{pre}} = 0.0003$.

After level-5 refinement, the maximum value of the electric field near the tip reaches an approximately constant value of 11.323 V/nm. Note that the parallel performance of the PAMR module is not discussed here for brevity purposes but it appears in detail elsewhere [20]. All the cases shown in succeeding sections apply this mesh refinement module for a better resolution near the emitter tip.

3.4. Application to field emission prediction

A completed parallel Poisson's equation solver with parallel adaptive mesh refinement is used to compute the electric field distribution of a CNT-based field emitter without considering space-charge effect. The generally accepted Fowler–Nordheim theory [7] for a clean metal surface relates the field emission's current density, J, to the electric field at the tip surface of the emitter, E, in volts/nm and the work function of the emitter, ϕ,

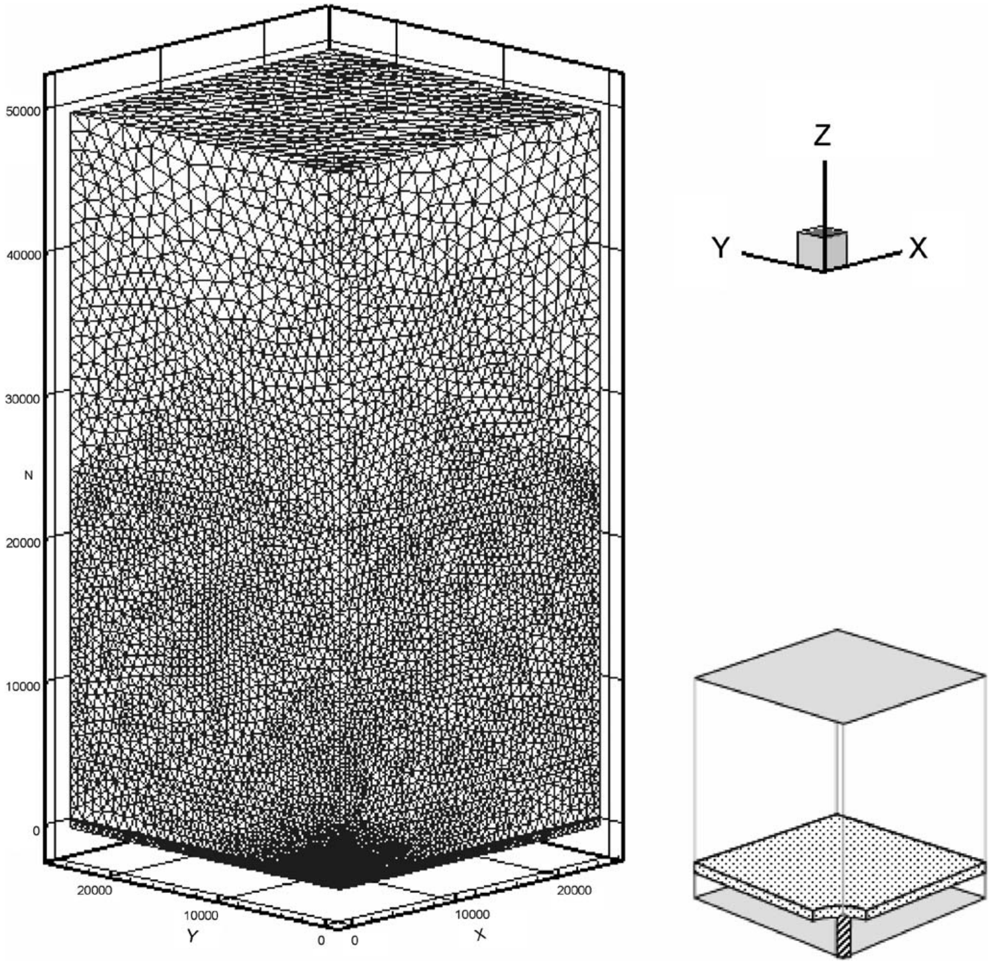


Fig. 5. Surface mesh distribution of a typical single CNT triode-type field emitter within a periodic cell. Only $\frac{1}{4}$ of a periodic cell is simulated for the study of parallel performance of the Poisson's equation solver.

in electron volts (eV) by the equation,

$$J = \frac{AE^2}{\phi t^2(y)} \exp\left(-B\frac{\phi^{3/2}}{E}v(y)\right) \text{Ampere/cm}^2, \qquad (10)$$

where

$$A = 1.5414 \times 10^{-6}, \qquad (10a)$$

$$B = 6.8309 \times 10^{7}, \qquad (10b)$$

$$y = 3.79 \times 10^{-4} E^{1/2}/\phi \qquad (10c)$$

and y is the image charge lowering the contribution to the work function. The functions $t(y)$ and $v(y)$ are approximated by $t^2(y) = 1.1$, $v(y) = 0.95 - y^2$.

The Electron trajectory from the emitter surface to the anode surface is traced on the unstructured mesh based on the computed electric field distribution from the Poisson's equation solver, by using the cell-by-cell particle tracking technique developed previously for DSMC simulation [22]. The current density is then computed as the time average of the accumulated charges due to electron flow reaching the anode surface.

Fig. 4 depicts the simulation domain for a typical CNT triode-type field emitter within a periodic cell. Only $\frac{1}{4}$ of the full emitter is used due to the intrinsic symmetry with Neumann boundary conditions applied at all symmetric planes. Important

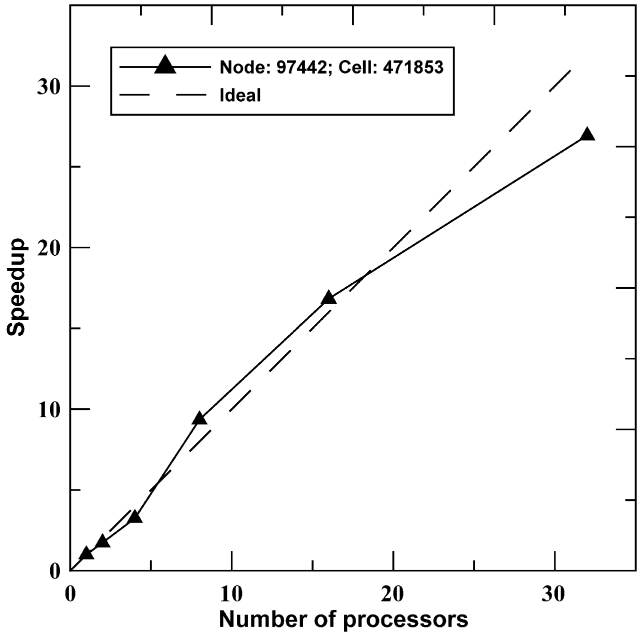


Fig. 6. Parallel speedup as a function of the number of processors on the PC-cluster system (maximum 32 processors) for CNT triode-type field emitter with gate voltage 150 volts, anode voltage 400 volts and the grounded cathode.

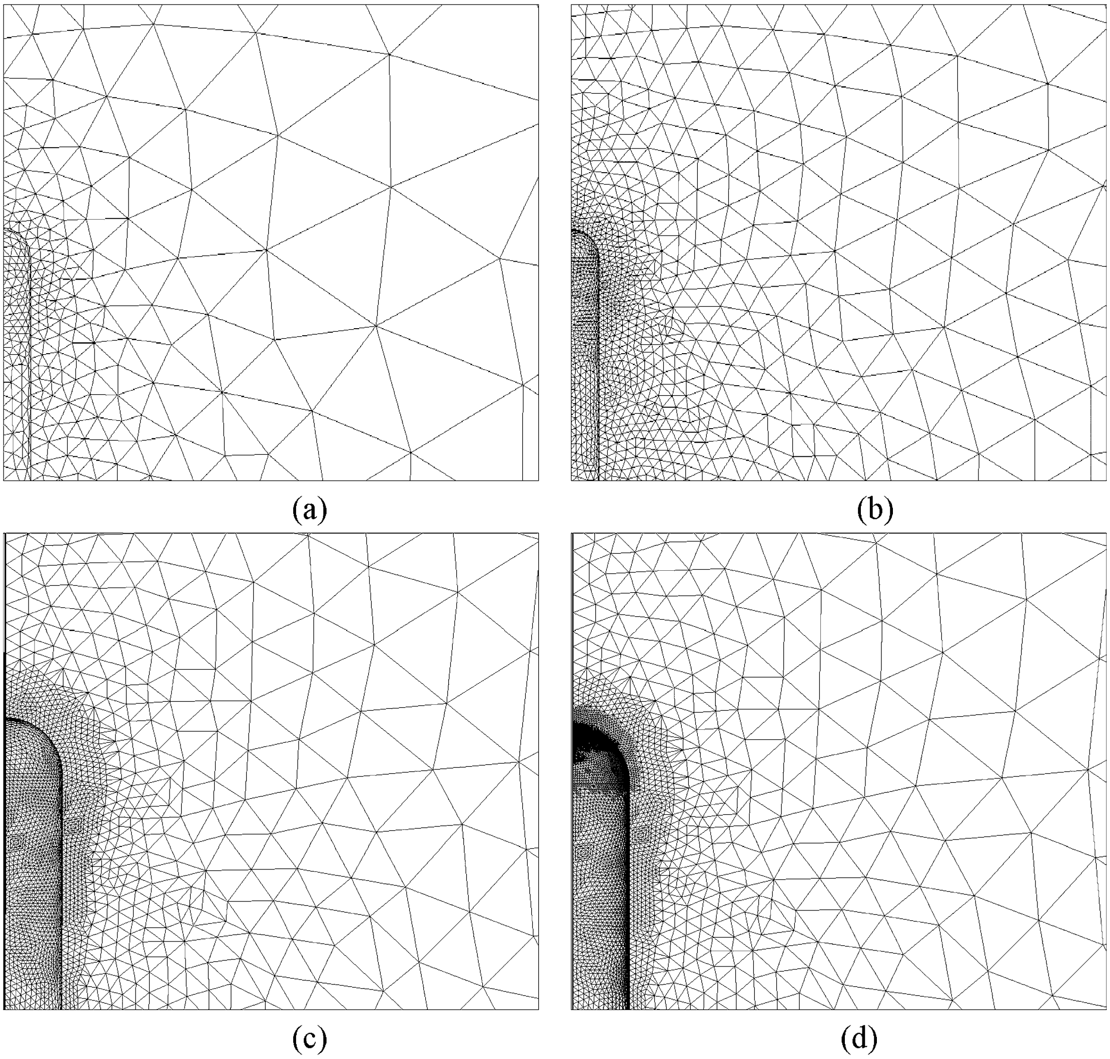


Fig. 7. Close-up view of the unstructured adaptive surface mesh at different levels for a single CNT triode-type field emitter with gate voltage 150 volts, anode voltage 400 volts and the grounded cathode ($\varepsilon_{\text{ref}} = 0.08$). (a) Level-0 (7006 nodes). (b) Level-1 (22 750 nodes). (c) Level-2 (34 927 nodes). (d) Level-5 (61 241 nodes).

geometrical conditions (also summarized in part in Table 3) include a tip radius of 10 nm, an emitter height of 600 and 400 nm, a distance of 0.5 μm between the gate and the cathode, a gate radius of 0.5 μm above the emitter, a distance of 50 μm between the anode and the cathode, a thickness of the gate of 0.2 μm, and the half width of each cell measuring 25 μm. The applied voltage of the gate ranges from 110 to 190 volts, while the cathode and anode are grounded and applied with 400 volts, respectively. The refined final number of nodes used for the simulation is approximately 90 000. The typical results of the predicted potential distribution along with electric field distribution (gate voltage = 150 volts, height = 600 nm) are shown in Fig. 8(a) and (b), respectively. The maximal value of the electric field can reach up to ~11.47 V/nm at the emitter tip when the gate voltage is 150 volts.

The predicted current and voltage data with an emitter height of 600 nm are presented in Fowler–Nordheim format in Fig. 9, with an anode voltage of 400 volts. It is clear that the computed I–V data follow the Fowler–Nordheim law very well as the gate voltage varies from 110 to 160 volts. The fitted field enhancement factor ($\beta = E\frac{d}{V}$) is 26.1, where V is the applied cathode voltage, and d is the vacuum gap in the field emission diode configuration. The corresponding electron trajectories are

Table 3
The important geometrical parameters of CNT triode- and tetrode-type field emitters

	Triode-type (Fig. 7)	Tetrode-type (Fig. 12)
he	600 nm	600 nm
r	10 nm	10 nm
R	500 nm	500 nm
R_f	N/A	1500 nm
d	200 nm	N/A
h	500 nm	N/A
d_1	N/A	200 nm
d_2	N/A	200 nm
h_1	N/A	500 nm
h_2	N/A	500 nm
L	49.3 μm	48.6 μm
W	25 μm	25 μm

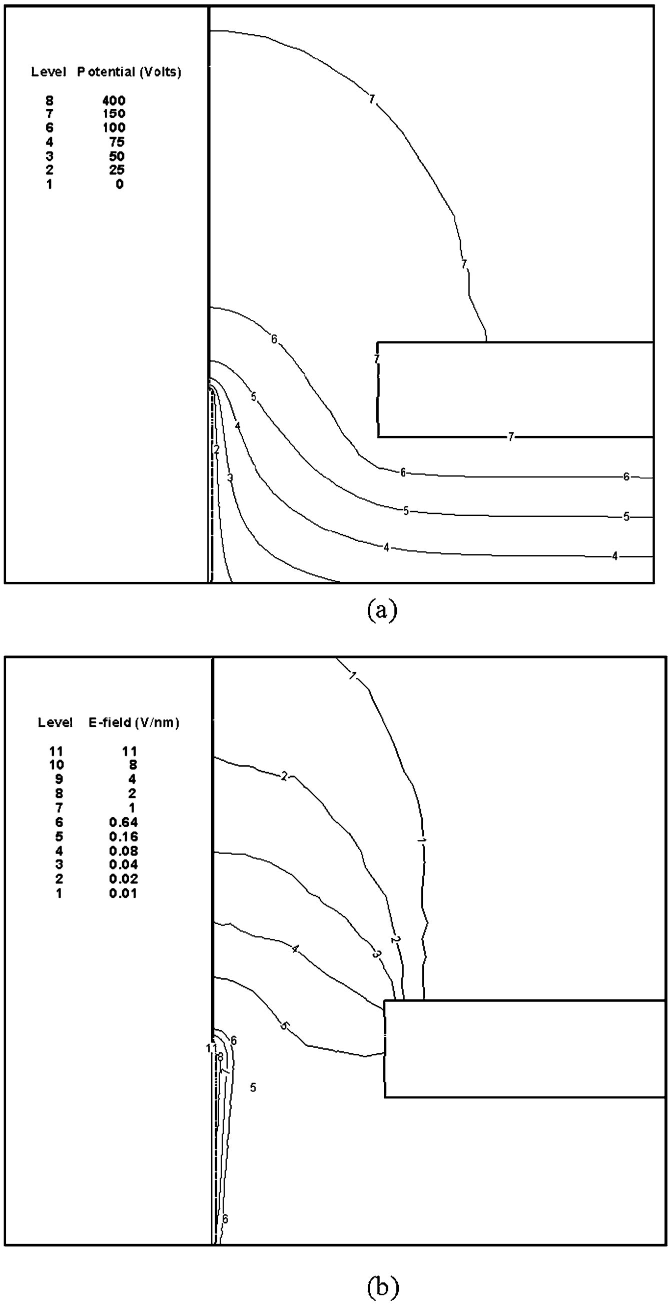


Fig. 8. Contours of the (a) electric potential and the (b) electric field distribution near the tip of the CNT triode-type field emitter with gate voltage 150 volts, anode voltage 400 volts and the grounded cathode.

illustrated in Fig. 10 at two different gate voltages (110 and 160 volts) with a height of 600 nm. The results show that the spreading angle of electrons from the tip increases with the increasing gate voltage. This is attributed to the fact that the area of the tip surface which has a larger local electric field increases as the applied voltage increases, which results in the greater emission of electrons from the side of the emitter near the tip. As will be shown later, adding a focusing gate can help to effectively reduce the spreading angle.

The effects of CNT height and gate voltage to the emission current under an applied voltage of 400 volts are presented in Fig. 11, with the CNT measuring 400 and 600 nm, respectively.

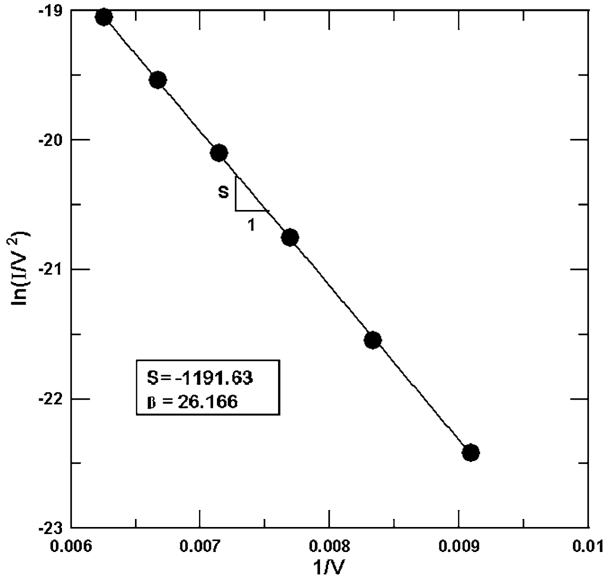


Fig. 9. FN plot of the field emission characteristics of CNT triode-type field emitter (height is 600 nm) with gate voltage 110–160 volts, anode voltage 400 volts and the grounded cathode. ($S \equiv slope = -3244.25\phi^{3/2}/\beta$, $\phi = 4.52$ eV.)

The results show that the turn-on voltage increases with the decreasing height of the CNT emitter. Also, the emission current increases dramatically with the given CNT height. This is reasonable since the larger the height of the CNT, the larger the local electric field which results at the tip surface (shorter anode-cathode distance with the same voltage difference), which in turn induces greater emission of electrons.

Fig. 12 shows schematically the same field emitter as shown in Fig. 4 with an additional focusing gate in-between the gate electrode and anode. Most geometrical conditions (also summarized in part in Table 3) are the same as those in Fig. 4, except for the distance between the focusing electrode and the gate electrode measuring 0.5 μm, the thickness of the focusing electrode measuring 0.2 μm, and the radius of the hole in the center of the focusing electrode which is 1.5 μm. Similar to that in the previous case without the focusing gate, only $\frac{1}{4}$ of a periodic cell is used for the simulation. Fig. 13(b)–(d) present a comparison of the focusing effects of electron trajectories using different focusing electrode voltages (5, 0, −5 volts). Likewise, data involving the absence of focusing electrode are presented for the purpose of comparison (Fig. 13(a)). The results show that the addition of a focusing electrode above the gate electrode can effectively reduce the spreading angle of the electron trajectories, which can possibly increase the resolution and the intensity at the anode. Among the cases simulated, focusing the electrode with 5 volts represents the best choice in focusing the electron flows at the anode.

The above computational examples (Figs. 4 and 12) only serve to demonstrate the capability of the current parallel Poisson's equation solver using FEM with parallel adaptive mesh refinement in predicting field emission properties with complicated geometries. No thorough parametric studies have been explored in the current study, although such are worthy of further investigation.

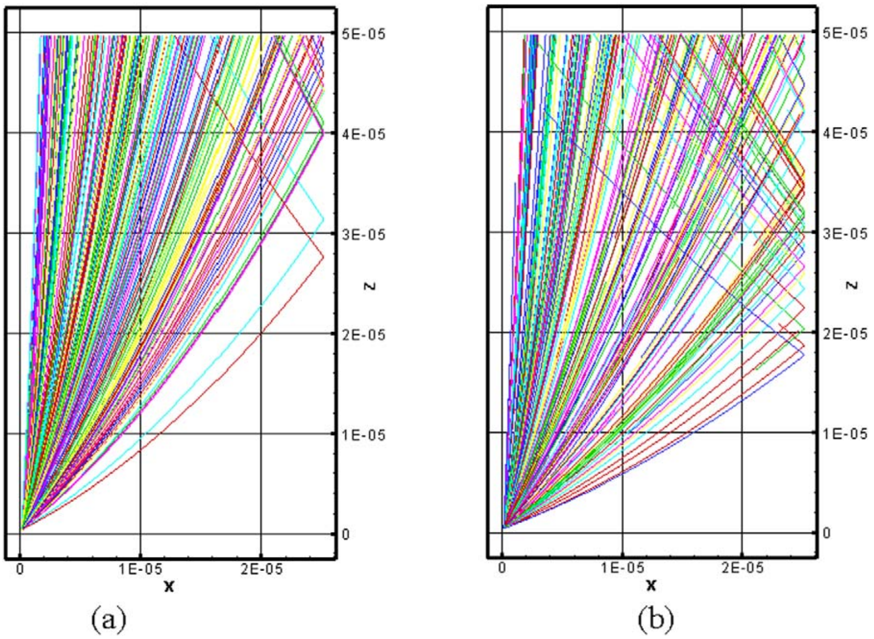


Fig. 10. Trajectories of the emitted electrons inside the periodic cell of CNT triode-type field emitter with the grounded cathode, anode voltage 400 volts and two different gate voltages: (a) 110 volts, (b) 160 volts.

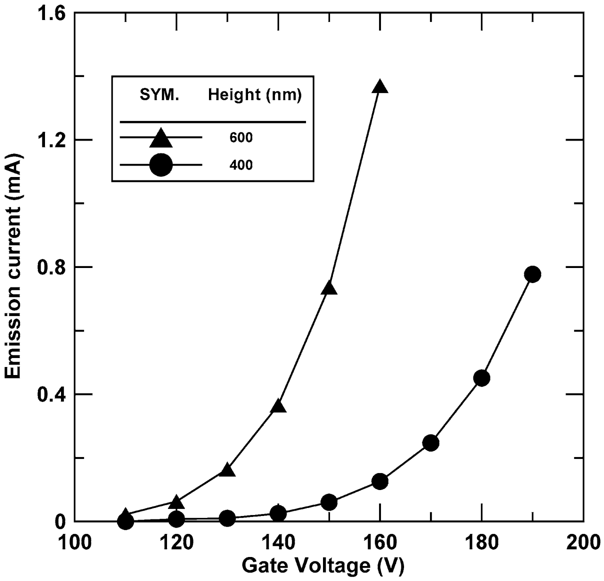


Fig. 11. Effect of the gate voltage on the emission current for two different CNT triode-type field emitter heights with anode voltage 400 volts and the grounded cathode.

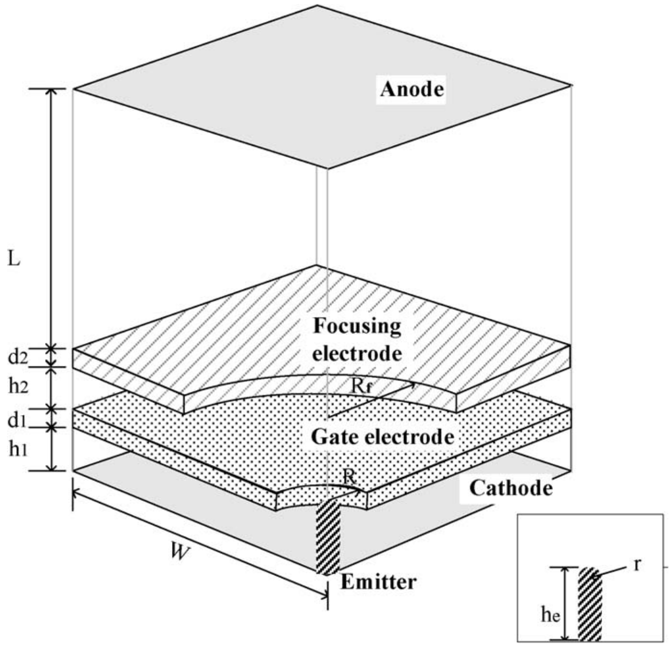


Fig. 12. Schematic diagram of the simulation domain for a typical CNT tetrode-type field emitter within a periodic cell. The important parameters are: $R = 500$ nm, $R_f = 1500$ nm, $r = 10$ nm, $he = 600$ nm, $h_1 = 500$ nm, $h_2 = 500$ nm, $L = 48.6$ µm, $d_1 = 200$ nm, $d_2 = 200$ nm and $W = 25$ µm.

4. Conclusions

A parallel electrostatic three-dimensional Poisson's equation solver using the Galerkin finite-element method coupled with parallel mesh refinement (PAMR) using an unstructured tetrahedral mesh is developed in this paper. The parallel performance of the parallel Poisson's equation is also studied using a triode-type CNT field emitter. The results show that a parallel efficiency of 84.2% is achieved using 32 processors with a problem size pertinent to the application in the prediction of field emission properties. A completed code is then applied to compute the field emission properties of the triode-type CNT field emitter with and without a focusing electrode to demonstrate its capability in predicting field emission properties with complicated geometries. Parametric studies using this code for some practical cases considering multi-CNT emitters are currently in progress and will be reported in the near future. In addition, a study predicting the field emission current by considering the space-charge effect caused by emitted electrons using the particle-in-cell method [23] is also in progress.

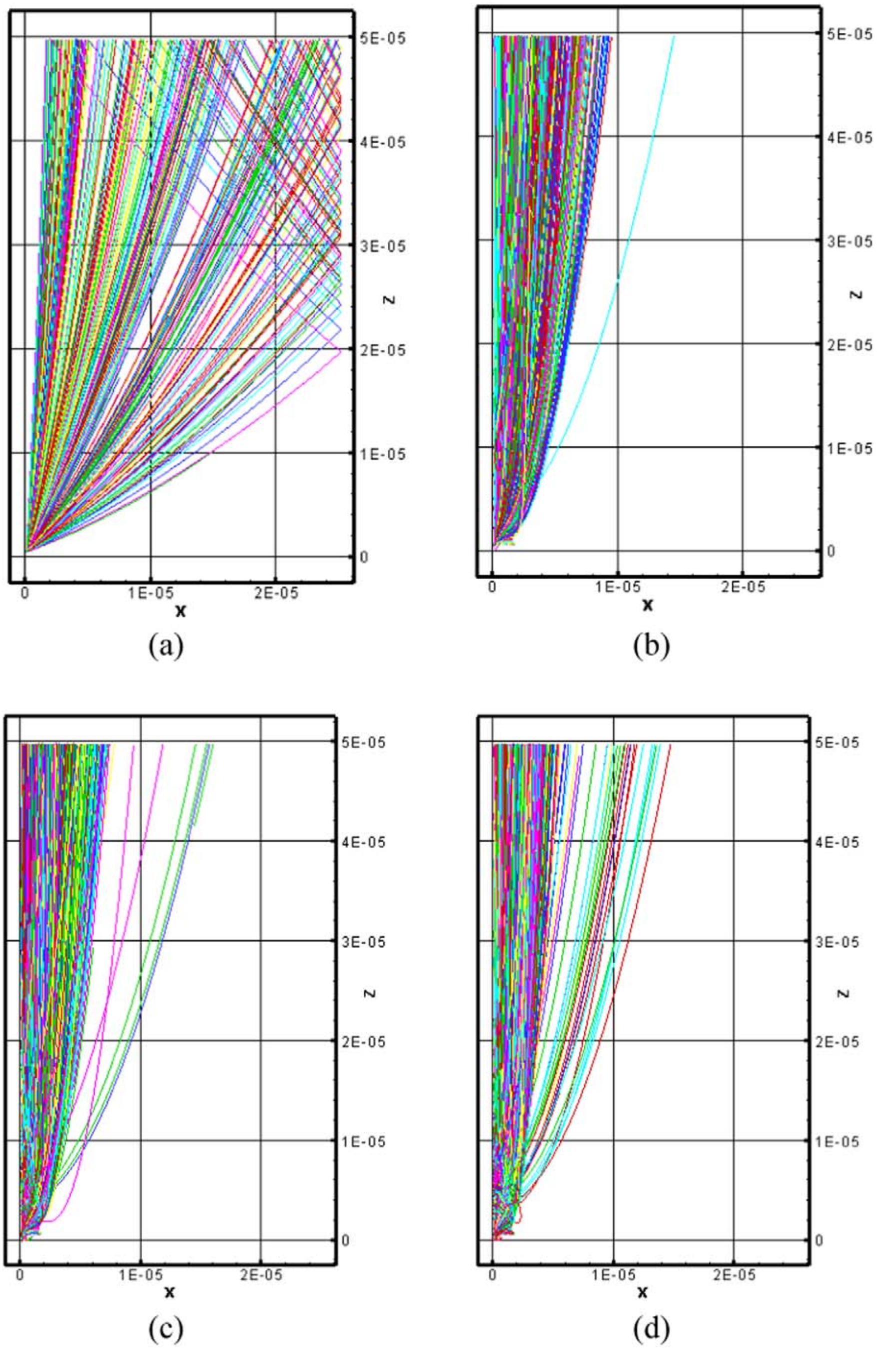


Fig. 13. Comparisons of the trajectories of the emitted electrons between (a) CNT triode-type field emitter with the grounded cathode, anode voltage 400 volts and the gate voltage 150 volts and tetrode-type field emitter with the additional three different focusing voltages: (b) 5 volts, (c) 0 volts, (d) −5 volts.

Acknowledgements

This investigation is supported by the National Science Council of Taiwan, with grants NSC-932212-E-009-015 and NSC 93-2212-E-009-015. The authors would like to thank the National Center for High-Performance Computing of Taiwan for providing the computing resources. Also, they would like to express their sincere gratitude to Prof. Karypis of the University of Minnesota for generously providing references from the partitioning library, ParMetis.

References

[1] W.A. Heer, A. Chatelain, D. Ugarte, Science 270 (1995) 1179.

[2] A.M. Rao, D. Jacques, R.C. Haddon, W. Zhu, C. Bower, S. Jin, Appl. Phys. Lett. 76 (2000) 3813.

[3] K. Nishimura, Z. Shen, M. Fujikawa, A. Hosono, N. Hashimoto, S. Kawamoto, S. Watanabe, S. Nakata, J. Vac. Sci. Technol. B 22 (2004) 1377.

[4] L. Nilsson, O. Groening, C. Emmenegger, O. Kuettel, E. Schaller, L. Schlapbach, H. Kind, J.M. Bonard, K. Kern, Appl. Phys. Lett. 76 (2000) 2071.

[5] S. Itoh, M. Tanaka, T. Tonegawa, J. Vac. Sci. Technol. B 22 (2004) 1362.

[6] Q.H. Wang, T.D. Corrigan, J.Y. Dai, R.P.H. Chang, Appl. Phys. Lett. 70 (1997) 3308.

[7] R.H. Fowler, L. Nordheim, Proc. Roy. Soc. A (London) 119 (1928) 173.
[8] C.A. Spindt, J. Appl. Phys. 39 (1968) 3504.
[9] W.B. Choi, D.S. Chung, J.H. Kang, et al., Appl. Phys. Lett. 75 (1999) 3129.
[10] G. Pirio, P. Legagneux, D. Pribat, et al., Nanotechnology 13 (2002) 1.
[11] M.A. Hong, J. Vac. Sci. Technol. B 12 (1994) 764.
[12] C. Wang, B. Wang, H. Zhao, et al., J. Vac. Sci. Technol. B 15 (1997) 394.
[13] W. Lei, B. Wang, H. Yin, J. Vac. Sci. Technol. B 16 (1998) 2881.
[14] Y. Hu, C.H. Huang, J. Vac. Sci. Technol. B 21 (2003) 1648.
[15] Y.C. Lan, J.T. Lai, S.H. Chen, et al., J. Vac. Sci. Technol. B 18 (2000) 911.
[16] Y.C. Lan, C.T. Lee, Y. Hu, et al., J. Vac. Sci. Technol. B 22 (2004) 1244.
[17] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulations, Institute of Physics Publishing, Philadelphia, USA, 1991.
[18] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2003.
[19] G. Karypis, K. Schloegel, V. Kumar, ParMetis, University of Minnesota, Department of Computer Science, September 1998.
[20] Y.-Y. Lian, K.-H. Hsu, Y.-L. Shao, et al., Parallel adaptive mesh-refining scheme on three-dimensional unstructured tetrahedral mesh and its applications, Comput. Phys. Comm. (2006), submited for publication.
[21] J.S. Wu, K.C. Tseng, F.Y. Wu, Comput. Phys. Comm. 162 (2005) 166.
[22] J.S. Wu, Y.Y. Lian, Comput. Fluids 32 (2003) 1133.
[23] J.S. Wu, K.H. Hsu, C.T. Hung, L.H. Chen, in: Proc. 32nd Internat. Conf. on Plasma Science (Monterey, USA), 2005, p. 251.
[24] O.C. Zienkiewicz, J.Z. Chu, Internat. J. Numer. Methods Engrg. 24 (1987) 337.