Parallel Beam Propagation Methods

Husain M. Masoudi and John M. Arnold

Abstract—**We have implemented two explicit finite-difference beam propagation methods on a transputer array for analyzing three-dimensional optical semiconductor devices. Both methods, in their parallel form, can execute, per propagational step, a large problem that contains 10^6 discretization points in a few seconds. We compare the speed of the transputer implementations to the speed of Connection Machine implementation of the same methods.**

I. Introduction

The Beam Propagation Method (BPM) has been used, in the last decade or so, to analyze various two- and three-dimensional optical devices. It has been demonstrated by many authors that the finite-difference BPMs are more efficient than the classical FFT-BPM [1]–[4]. The application of the FD-BPM to 2-D structures proves to be very efficient because it involves computation only one-dimensional arrays. On the other hand, practical optical devices may contain multiple linear and nonlinear rectangular waveguides (3-D) where simulating such devices, using the BPM, on a serial computer accurately is very time consuming. In addition to the large problem posed by rectangular waveguides, the existence of large contrast media, which forces the BPM to use small transverse mesh sizes and small longitudinal step size in order to converge or to be stable, will multiply the computational effort many-fold. The best way to speed up the execution of a large problem, such as the BPM, is to use parallel computers. These computers are designed specifically to execute large numerical algorithms rapidly using the idea of breaking the problem into small pieces and arranging to solve for all pieces simultaneously. In this letter, we show that the implementation of two finite difference BPMs on a transputer array will speed up the execution of these methods with a very high gain in the efficiency. Then we compare these results to the implementation of the same methods on the Connection Machine. In this work we will focus on the efficiency issue rather than a comparison between the two methods, which requires an accuracy assessment. Full details of these results will be published later.

II. Numerical Methods

The parabolic wave equation in three dimensions can be written as

$$2jkn_0\frac{\partial E}{\partial z} = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + k^2(n^2 - n_0^2)E. \qquad (1)$$

Manuscript received December 27, 1993; revised March 25, 1994. This work was supported in part by the Science and Engineering Council under Grant GR/H82471.

The authors are with the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, U.K.

IEEE Log Number 9402536.

There are several methods to simulate the propagation in (1) based on the finite difference approximation. The Explicit Finite Difference (EFD-BPM) in [2] uses a direct application of the finite difference approximation to the parabolic equation. This leads to an algorithm that involves multiplication of the initial optical field with a very sparse matrix with only five nonzero elements in each row of the matrix. The EFD-BPM, which is second-order accurate with respect to z, is very efficient but only conditionally stable. The stability of this method depends on the transverse mesh sizes and the difference between the refractive index of reference and the transverse media. As the transverse mesh sizes decrease, the longitudinal step size must also decrease. On the other hand, other FD-BPM's use the operator in (2) to approximate (1),

$$E(x, y, z+\Delta z) = e^{\left(-j\frac{\Delta z}{2a}\nabla_x^2\right)} e^{\left(-j\frac{\Delta z}{2a}\nabla_y^2\right)} e^{\left(-j\frac{\Delta z}{a}d\right)} e^{\left(-j\frac{\Delta z}{2a}\nabla_x^2\right)} \cdot e^{\left(-j\frac{\Delta z}{2a}\nabla_y^2\right)} E(x, y, z) + O((\Delta z)^3) \qquad (2)$$

where

$$a = 2kn_0$$

$$d(x, y, z) = k^2[n^2(x, y, z) - n_0^2]$$

$$\nabla_\rho^2 = \frac{\partial^2}{\partial \rho^2}, \qquad (\rho = x, y).$$

The most popular BPM is based on the implicit approach which is the Alternating Direction Implicit approximation (ADI-BPM) [1], [3], [4]. This algorithm uses the operator in (2) combined with the following relation:

$$e^{\left(-j\frac{\Delta z}{2a}\nabla_\rho^2\right)} = \frac{\left[1 - j\frac{\Delta z}{4a}\frac{\partial^2}{\partial \rho^2}\right]}{\left[1 + j\frac{\Delta z}{4a}\frac{\partial^2}{\partial \rho^2}\right]}. \qquad (3)$$

The ADI-BPM is unconditionally stable, but requires solution of a large system of tridiagonal equations for each propagational step. On the other hand, the Real Space method (RS), which uses the finite difference matrix splitting operators, can be written as [3], [5]

$$\begin{aligned} &E(x, y, z + \Delta z) \\ &\quad = e^{(\alpha_y S_y^0)} e^{(\beta_y S_y^e)} e^{(\alpha_y S_y^0)} e^{(\alpha_x S_x^0)} e^{(\beta_x S_x^e)} e^{(\alpha_x S_x^0)} \\ &\quad \cdot e^{\left(-j\frac{\Delta z}{a}U\right)} e^{(\alpha_y S_y^0)} e^{(\beta_y S_y^e)} e^{(\alpha_y S_y^0)} e^{(\alpha_x S_x^0)} \\ &\quad \cdot e^{(\beta_x S_x^e)} e^{(\alpha_x S_x^0)} E(x, y, z) + O((\Delta z)^3) \end{aligned} \qquad (4)$$

where

$$U(x, y, z) = d(x, y, z) - \frac{2}{\Delta x^2} - \frac{2}{\Delta y^2}$$

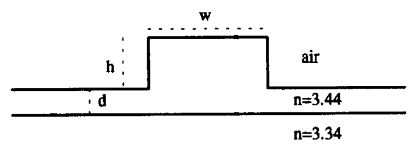

Fig. 1. The rib waveguide.

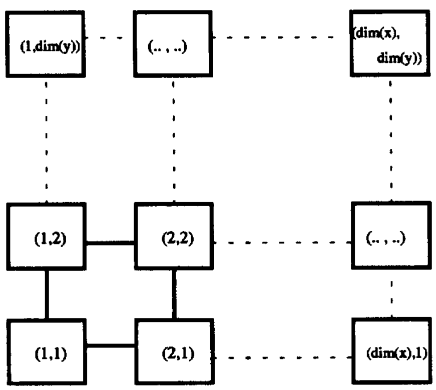

Fig. 2. The 2-D grid topology used for the implementation of both the EFD-BPM and the RS-BPM. The number shown indicates the position of each processor in terms of the 2-D grid.

$$\beta_\rho = -j\frac{\Delta z}{2a\Delta\rho^2} = 2\alpha_\rho.$$

The splitting is chosen such that each matrix S_ρ^o and S_ρ^e in (4) is block-diagonal where each block contains a small submatrix (e.g., 2×2) which may be easily exponentiated analytically. This method is unconditionally stable but, as the FFT-BPM, requires small propagational steps to converge when applied to large contrast media. However, it proves to be much more efficient per propagational step than the ADI-BPM because it does not involve solving a system of equations but multiplication of independent small matrices. Full details of the RS method can be found in [5].

We can summarize that both the EFD-BPM and the RS-BPM are explicit and highly parallel due to the locality of their spatial points, which reduces the movement of data between processors. In contrast, the ADI-BPM requires information from all parts of the problem in order to perform the propagation on any given spatial point. Hence, both of the explicit methods are more efficient than the implicit method because they are much more efficient per propagational step and they gain a larger speedup when run on supercomputers. For these reasons, we have not fully implemented the ADI-BPM on parallel machines.

We have applied and tested both the EFD-BPM and the RS-BPM using the scalar parabolic equation in (1) to analyze a rib waveguide shown in Fig. 1 using a wavelength of $\lambda = 1.55$, $w = 2$, $d = 0.2$, and $h = 1.1$ (all dimensions are in μm). We have used the 2-D grid topology shown in Fig. 2 for the implementation of both the EFD-BPM and the RS-BPM on the transputer array. The transputer is a PARSYTEC super-cluster consisting of 64 IMS-T800 processors (MIMD machine) each with 4 Mbytes. We believe that the topology shown in the figure is the best arrangement to parallelize both of these methods in terms of efficiency and transputer memory distribution. Each processor, except for those at the borders as shown in the figure, has four links connected to its neighbors, where each link is a bidirectional communication channel for exchanging data (local mesh points at the borders). The transverse mesh points of the explicit methods have been divided into 2-D identical blocks where the size of each block is equal to $(M_x/\dim(x), M_y/\dim(y))$. M_x and M_y are the number of mesh points in the transverse direction x and y, respectively, and dim(x) and dim(y) are the dimensions of processors in the x and y direction, respectively, of the 2-D grid topology shown in Fig. 2. Every processor in the topology has been assigned one of the blocks of the transverse mesh points. This arrangement will ensure that all processors have an equal amount of computation, without the need to load-balance the system, and it also gives the freedom to change both the number of processors and the number of mesh points without

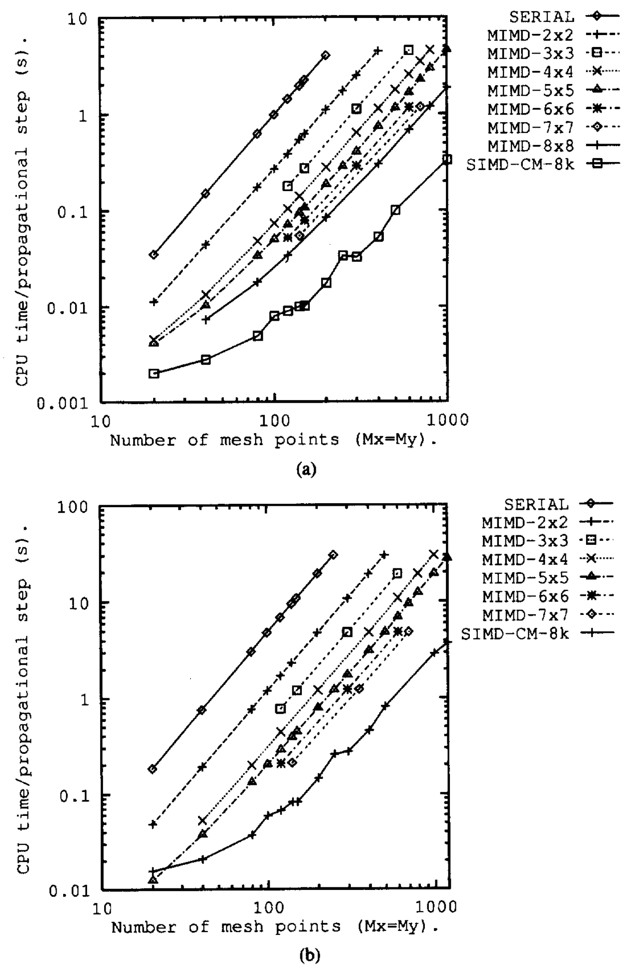

Fig. 3. Comparison between the speed of the serial and the parallel implementations of the explicit methods using the transputer array (MIMD) and the Connection Machine (SIMD-CM-8k); (a) The EFD-BPM; (b) the RS-BPM.

altering the parallel computer code. For every propagational step, all processors exchange data only from the border points of the local blocks. In order to study the gain in efficiency by using the transputer array, we have implemented both of the explicit methods on a single processor (serial) of the transputer in addition to the parallel implementations. All the computer codes were written in FORTRAN with a double precision accuracy. Fig. 3 (in log-log scale) shows the total CPU time of both of the second-order EFD-BPM and the second-order RS-BPM per propagational step versus the number of mesh points in one of the transverse directions x. We have set the number of mesh points in both directions to be equal ($M_x = M_y$) with uniform grid spacing. The figure shows both the serial and the transputer results (MIMD) in addition to the Connection Machine (CM-8k) results for comparison. It can be seen from Fig. 3 that for a fixed M_x, the speed of both methods increases as the number of processors increases. For the full transputer size (8×8), the speedup of the EFD-BPM at $M_x = 200$ is around 47.3 times the serial speed while for the transputer size of 7×7 (8×8 could not be used due to a hardware fault in one of the processors) the speedup of the RS-BPM at $M_x = 210$ is around 45.4 times the serial speed. On the other hand, generally the Connection Machine speed, using 8k processors, is always a few times faster than the best performance of the transputer. The serial computation could not be continued after $M_x = 200$ for the EFD-BPM and $M_x = 250$ for the RS-BPM due to the computer memory limit; therefore, Fig. 3 does not contain results for M_x exceeding these values. Also from the figure we can calculate the percentage efficiency gain by using the transputer for both methods. We define the efficiency as the speedup value divided by the number of processors used by the parallel methods. We can see from the figure that for a fixed number of processors, the efficiency gain of both methods increases as the number of mesh points increases. For $M_x = 200$, the percentage efficiency of the EFD-BPM and the RS-BPM is around 90 and 100%, respectively, when using four processors. On the other hand, at $M_x = 200$ for the EFD-BPM when using 64 processors, and at $M_x = 210$ for the RS-BPM when using 49 processors, the percentage efficiency is around 74 and 93%, respectively.

In summary, we have demonstrated that the implementation of the finite-difference explicit methods of the BPM on a transputer array results in a large speedup of the execution of these methods compared to the serial ones. Comparison between the execution speed of implementing these methods on a transputer array and the Connection Machine showed that the Connection Machine execution speed is a few times faster than the transputer execution speed. Although these methods have limitations on their longitudinal step length when applied to large contrast media, both of these methods are very efficient finite-difference BPM's because of their efficiency per propagational step, as normal serial algorithms, and their large speedup when implemented on parallel computers.

ACKNOWLEDGMENT

Both authors would like to thank the University of Glasgow for providing the facility. H. M. Masoudi is very grateful for the support of King Fahd University of Petroleum and Minerals (Saudi Arabia) while he is at the University of Glasgow.

REFERENCES

[1] Y. Chung and N. Dagli, "An assessment of finite difference beam propagation method," *IEEE J. Quantum Electron.*, vol. 26, pp. 1335–1339, 1990.
[2] ——, "Analysis of Z-invariant and Z-variant semiconductor rib waveguides by explicit finite difference beam propagation method with nonuniform mesh configuration," *IEEE J. Quantum Electron.*, vol. 27, pp. 2296–2305, 1991.
[3] D. Yevick and B. Hermansson, "Efficient beam propagation techniques," *IEEE J. Quantum Electron.*, vol. 29, pp. 109–112, 1990.
[4] W. Huang, C. Xu, S. Chu, and S. Chaudhari, "The finite difference vector beam propagation method: Analysis and assessment," *J. Lightwave Technol.* , vol. 10, pp. 209–305, 1992.
[5] H. DeRaedt, "Product formula algorithms for solving the time dependent Schrodinger equation," *Comput. Phys. Rep.*, vol. 7, pp. 1–72, 1987.
[6] J. Richardson, "Visualizing quantum scattering on the CM-2 supercomputer," *Comput. Phys. Comm.*, vol. 63, pp. 84–94, 1991.