

Progressive Analysis of Electromagnetic Field by Parallel FDTD Algorithm for Big Urban Area 2-D Simulations

Glen Rodriguez* Non-member
Yasumitsu Miyazaki* Member

This paper explores some improvements in the parallel FDTD algorithm using system of linear equations. The objectives are to be able to simulate realistic size models of urban wireless communications using FDTD. The obstacles are the need of large memory and processing time. The previous research has dealt with decreasing the use of memory. From now, the research deals with the improvement of the speed of the algorithm. This is done by using the properties of the impulse response in the system of equations and using it to store pre calculated results that are the most common calculations in FDTD equations.

Keywords: mobile communication, FDTD method, parallel processing, system of linear equations

1. Introduction

Wireless communications is already an important segment of telecommunication traffic and it is one of the most fast expanding technologies. Between the fundamentals of this technology, there is the propagation of electromagnetic waves. The wave propagation and scattering happens in big areas with many kind of geometries. Because this arbitrary geometry, FDTD method is useful (3), but it demands a great amount of memory and computer time. Most research has been done to improve the speed by using FDTD parallel algorithms, but no effort has been put to being able to solve problems needing more memory than available. This limits the size of problems FDTD can deal with. In previous papers we have presented an initial research done to decrease the need of memory. A big size FDTD solver could allow simulating many interesting behaviors in realistic size models. But this saving in memory came with a cost in yet longer computing time (8). First, we will present a brief summary of the method, and then we will discuss some steps done in the improvement of the speed of our parallel FDTD algorithm.

2. Basis of the algorithm

This is a summary of the basis of the algorithm, which is discussed in detail in (1), (2), (7). The Yee's formulation for the FDTD method in the 2-D case, TM mode is used (Fig. 1). If there is only linear, isotropic nondispersive materials, the Yee's FDTD formulation is:

$$\begin{aligned}
 E_z^{n+1}(i, j) &= C_a[E_z^n(i, j)] \\
 &+ C_x[H_y^n(i+1, j) - H_y^n(i, j)] \\
 &- C_y[H_x^n(i, j+1) - H_x^n(i, j)] \\
 &+ f^n(i, j) \dots \dots \dots (1)
 \end{aligned}$$

$$\begin{aligned}
 H_x^{n+1}(i, j) &= H_x^n(i, j) - D_y[E_z^{n+1}(i, j) - E_z^{n+1}(i, j-1)] \\
 &\dots \dots \dots (2)
 \end{aligned}$$

$$\begin{aligned}
 H_y^{n+1}(i, j) &= H_y^n(i, j) + D_x[E_z^{n+1}(i, j) - E_z^{n+1}(i-1, j)] \\
 &\dots \dots \dots (3)
 \end{aligned}$$

Where C_a, C_x, C_y, D_x and D_y are constants with respect to cell (i, j) with the m -th material, and F is related to current J (in this case only J_z component) of the source antenna:

$$f(t) = f(n\Delta t) = \frac{\Delta t}{\epsilon} J_{src}(n\Delta t) \dots \dots \dots (4)$$

In these equations, the time variable is expressed as integer, that is, the time n and $n + 1/2$ are both expressed as n as a convention in this paper. Function f is the signal expressed in the same units as E (V/m), operating in this special point (i, j) , that is, zero everywhere except at antenna point. In this paper, standard FDTD means the original formulation of Yee's FDTD in Cartesian coordinates and 2nd order accuracy.

$$\left. \begin{aligned}
 C_a(m) &= \frac{1 - (\sigma(m)\Delta t/2\epsilon(m))}{1 + (\sigma(m)\Delta t/2\epsilon(m))} \\
 C_x(m) &= \frac{\Delta t/\epsilon(m)}{1 + (\sigma(m)\Delta t/2\epsilon(m))} \cdot \frac{1}{\Delta x} \\
 C_y(m) &= \frac{\Delta t/\epsilon(m)}{1 + (\sigma(m)\Delta t/2\epsilon(m))} \cdot \frac{1}{\Delta y} \\
 D_x(m) &= \frac{\Delta t}{\mu(m)\Delta x} \\
 D_y(m) &= \frac{\Delta t}{\mu(m)\Delta y}
 \end{aligned} \right\} \dots \dots (5)$$

The Yee's formulation can be expressed as a coupled system of matricial equations as in Ref. (6). The system is shown in eq.(14),(15), X_i is a vector with all the variables E_z, H_x, H_y in time step i , and in most simulations $X_0 = \vec{0}$. Matrix A_1 in eq. (15) represents the

*Toyohashi University of Technology
Hibarigaoka 1-1, Tempaku-cho, Toyohashi 441-8580

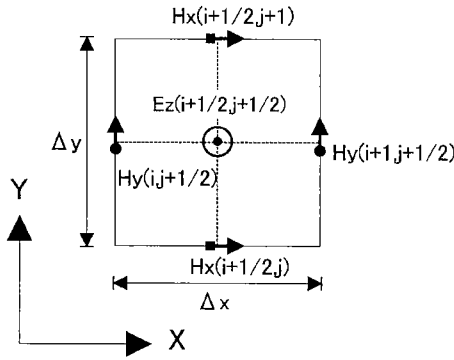


Fig. 1. Fields in the 2-D, TM problem

linear relations which transforms variables of time step n into variables of time step $n+1$, and it is a matrix of size $3pq \times 3pq$. The coefficients for the rows (equations) corresponding to E_z are the 5 coefficients of the right side of eq. (1), one coefficient for each independent variable (C_a once, C_x twice, C_y twice). The coefficients for H_x and H_y are derived by substitution of the value of E_z in eq. (1) into eq. (2), (3) as follows:

$$\begin{aligned}
 H_x^{n+1}(i, j) &= H_x^n(i, j) - D_y(m_1)[C_a(m_1)E_z^n(i, j) \\
 &+ C_x(m_1)(H_y^n(i+1, j) - H_y^n(i, j)) \\
 &- C_y(m_1)(H_x^n(i, j+1) - H_x^n(i, j)) \\
 &- C_a(m_2)E_z^n(i, j-1) \\
 &- C_x(m_2)(H_y^n(i+1, j-1) - H_y^n(i, j-1)) \\
 &+ C_y(m_2)(H_x^n(i, j) - H_x^n(i, j-1))] \dots \dots (6)
 \end{aligned}$$

$$\begin{aligned}
 H_y^{n+1}(i, j) &= H_y^n(i, j) - D_x(m_1)[C_a(m_1)E_z^n(i, j) \\
 &+ C_x(m_1)(H_y^n(i+1, j) - H_y^n(i, j)) \\
 &- C_y(m_1)(H_x^n(i, j+1) - H_x^n(i, j)) \\
 &- C_a(m_3)E_z^n(i-1, j) \\
 &- C_x(m_3)(H_y^n(i, j) - H_y^n(i-1, j)) \\
 &+ C_y(m_3)(H_x^n(i-1, j+1) - H_x^n(i-1, j))] \dots \dots (7)
 \end{aligned}$$

The equations for variables near the boundaries are different, because it is required to use some absorbing boundary condition (ABC). This algorithm uses the 1st order Mur ABC for its simplicity, its tolerable reflection and its dependence on variables for time step n and $n+1$ only. The equations are:

$$H_x^{n+1}(i, 1) = H_x^n(i, 2) + M_y(H_x^{n+1}(i, 2) - H_x^n(i, 1)) \dots \dots (8)$$

$$H_y^{n+1}(1, j) = H_y^n(2, j) + M_x(H_y^{n+1}(2, j) - H_y^n(1, j)) \dots \dots (9)$$

$$E_z^{n+1}(p, j) = E_z^n(p-1, j) + M_x(E_z^{n+1}(p-1, j) - E_z^n(p, j)) \dots \dots (10)$$

$$E_z^{n+1}(i, q) = E_z^n(i, q-1) + M_y(E_z^{n+1}(i, q-1) - E_z^n(i, q)) \dots \dots (11)$$

$$M_x = \frac{c\Delta t - \Delta x}{c\Delta t + \Delta x} \dots \dots (12)$$

$$M_y = \frac{c\Delta t - \Delta y}{c\Delta t + \Delta y} \dots \dots (13)$$

Again, the values in the right side must be replaced by its equivalents using eq. (1)~(3) in order to express the left side variables at time $n+1$ as dependent on variables at time n . While it is true that 1st order Mur ABC has more undesired reflections than others ABCs, if the simulation can be big, more space between the scatterers and the boundaries can be arranged and this lessens the impact of undesired reflections.

The coefficient matrix A of linear equation (16) is formed by the same components as matrix A_1 . Matrix A can be expressed as a block banded matrix with two block components, matrix I (diagonal has only '1' components) and the sparse matrix A_1 , see eq.(16), and the constant matrix b consist of small matrices ϕ_n where $\phi_n = [0, 0, \dots, f_k(n\Delta t), \dots, 0]'$, $0 < n \leq T$, $f_k(n\Delta t)$ is the value of E_z component in the k -th antenna at time $n\Delta t$.

$$X_n = [H_{x1,1,n}; H_{y1,1,n}; E_{z1,1,n}; \dots; E_{z p,q,n}]' \dots (14)$$

$$\text{Let be } A_1 \text{ fixed matrix} \Rightarrow X_{n+1} = A_1 X_n \dots (15)$$

$$\begin{bmatrix} I & 0 & \dots & 0 & 0 \\ -A_1 & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \\ 0 & 0 & \dots & -A_1 & I \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{T-1} \\ X_T \end{bmatrix} = \begin{bmatrix} A_1 X_0 \\ \phi_1 \\ \vdots \\ \phi_{T-1} \\ \phi_T \end{bmatrix} \dots \dots (16)$$

In eq.(14), p is the number of cells in the x direction, and q is the number of cells in the y direction. In eq.(16), T is the number of time steps through the simulation, and $\phi_0 = A_1 X_0$. The matrix A_1 is an operator that express the relation between the variables at time step n and time step $n+1$, its elements are the coefficients of eq.(1),(6),(7). And matrix A is the coefficient matrix of a linear system of equations (big matrix at the left side of eq.(16))that is equivalent to solving the FDTD simulation.

From this point, the unknown components of \mathbf{E} and \mathbf{H} will be represented as x variables, and the following notation is defined: $x_{t,i,j,c}$ represents $H_x^t(i, j)$ if $c=0$, $H_y^t(i, j)$ if $c=1$ and $E_z^t(i, j)$ if $c=2$. Other notation will be $x_{\mathcal{L}}$ which is equivalent to some $x_{t,i,j,c}$, where $\mathcal{L} = (t-1)(3pq) + 3qx + 3y + c$; this formula works in both ways. Matrix A is a sparse matrix and its rows are stored in memory using sparse matrix techniques.

Gaussian back-substitution is generally used for solving upper triangular systems(10). Here, it is used in a lower triangular system. Partial Gaussian back-substitution is carried from the equation for $x_{\mathcal{L}} = x_{T,i_0,j_0,c_0}$ (greatest time step T), many times, changing

the original equation of the following type, where each x_b^a are the equivalent to some variable $x_{T-a,i,j,c}$ and c_b^a are constant coefficients:

$$x_{\mathcal{L}} + c_1^1 x_1^1 + c_2^1 x_2^1 + \dots + c_{z_1}^1 x_{z_1}^1 = b_{\mathcal{L}}^1 \dots \dots \dots (17)$$

Into an equation with independent variables x_i^2 , by substitution of the variables x_i^1 using the same eq.(17) but for a different $x'_{\mathcal{L}} = x_i^1$:

$$x_{\mathcal{L}} + c_1^2 x_1^2 + c_2^2 x_2^2 + \dots + c_{z_2}^2 x_{z_2}^2 = b_{\mathcal{L}}^2 \dots \dots \dots (18)$$

This process is carried out n times until getting:

$$x_{\mathcal{L}} + c_1^N x_1^N + c_2^N x_2^N + \dots + c_{z_N}^N x_{z_N}^N = b_{\mathcal{L}}^N \dots \dots (19)$$

For a simulation of T time steps, if we split the calculations along the time using eq. (19) in equations advancing N time steps. The size of the array ($z_N = 6N^2 + 4N - 1$ for H_x or H_y and $6N^2 - 2N - 1$ for E_z) storing the data of eq. (19) multiplied by the number of equations (T/N) yields the necessary memory, and for some value of "N", the necessary memory becomes larger than the available memory. Then we take the variables $x_{T-N,i,j,c}$ one by one, create a new array and process this variable in the same way as we processed $x_{\mathcal{L}}$. Each stage of processing is called a level, where the stage processing $x_{\mathcal{L}}$ until getting all the $x_{T-N,i,j,c}$ is called level 1; the stage beginning with the processing of any x_b^N is level 2, and so on. Each level uses a distinct array to represent the respective equations. The algorithm is recursive in nature, that is, a process that calls itself. The flowchart of the algorithm called "solution" is as shown in Fig. 2. The initial call would use

the parameters (T,i,j,c).

The parallelization is done by distributing the storage and calculation of variables in the sparse arrays to different processors. Each processor stores one subset of variables and solves an different sub-summation which is later gathered into the final answer:

$$x_{\mathcal{L}} = - \sum_{u=1}^p \left(\sum_{v=1}^{z_N^p} c_v^N x_v^N \right) + constant \dots \dots \dots (20)$$

Where p is the number of processors and z_N^p is the load of each processor, which should be proportional to the speed of the each processor.

3. Improvement using Impulse Response

Analyzing the algorithm, we can see that the savings in memory comes because the standard FDTD method stores all variables of time step N to calculate all the variables of time step N+1. We using back-substitution, we can solve some variables at time step t_s using only a small set of dependent variables of time step $t_s - N$. The number of variables is proportional to $6N^2$, the number of step is T/N, then we need only stored variables in the order of 6TN. By choosing an appropriate n, this value 6TN becomes smaller than the 3pq variables of any time step. But this process also causes the repeated calculation of many variables. For example, for $N > 3$, to calculate the variable $E_z^t(i, j)$, we need to know $E_z^{t-N}(i-1, j)$, $E_z^{t-N}(i, j)$ and $E_z^{t-N}(i+1, j)$, and all this variables need the value of $E_z^{t-2N}(i, j)$. But in calculating those three variables at time t-N, the algorithm does not keep the values of variables at time t-2N (to save memory) and it must calculate $E_z^{t-N}(i+1, j)$ three times. This increases the processing time.

A second problem is that Gaussian back-substitution using sparse matrix techniques in A matrix is inefficient when the substitution reach many equations, because the matrix becomes less sparse and the sparse arrays must constantly fill in new variables. That is, the condition:

$$1 \leq isp(1) < isp(2) < \dots < isp(z) \leq 3pqT \dots (21)$$

must be hold for the array *isp* storing the columns of the non-zero coefficients of A. The requisite to keeping the arrays values ordered by column means that there is constantly insertion of new values when the matrix becomes less sparse.

We have researched an improvement that can deal mostly with the second problem and comply with the condition in eq. (21). It is based in the properties of the impulse response being used in FDTD linear equations. A Green's function is an integrating kernel that can be used to solve an inhomogeneous differential equation with boundary conditions. For any partial differential equation in a \mathcal{N} dimensional domain:

$$Lu(x) = f(x) \dots \dots \dots (22)$$

With operator L, we can think of an integral which solves that equation:

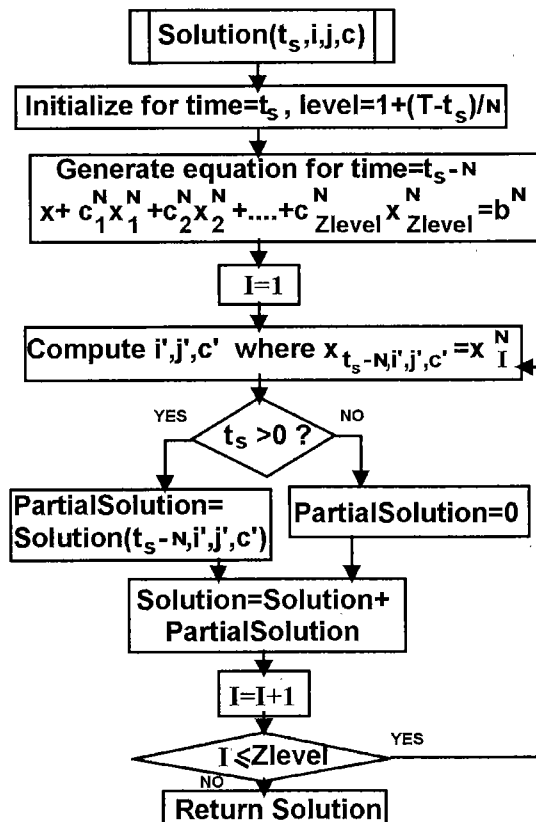


Fig. 2. Flowchart of the recursive algorithm

$$u(\mathbf{x}) = \int G(\mathbf{x}; \mathbf{x}') f(\mathbf{x}') d^N \mathbf{x}' \dots\dots\dots (23)$$

Here, $G(\mathbf{x}; \mathbf{x}')$ is the solution of:

$$LG(\mathbf{x}; \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}') \dots\dots\dots (24)$$

The physical meaning of the Green's function in wave and Maxwell equation is the dependence of some component of the electric or magnetic field at some time t and position \mathbf{r} ($\mathbf{x}=(\mathbf{r},t)$) on the values of the fields at any time t' ($t > t'$) and any position; in our case of study, this dependence is linear. Because those differential equations are hyperbolic, not all fields of time t' affect the value of some variable at time t , but it is limited by the speed of the wave propagation. And G it can be calculated solving the differential equation for the delta function.

The discrete approximation of the impulse response, $g(\mathbf{r},t,\mathbf{r}',t')$, can be calculated by using the numerical delta function: a unitary impulse at $t=1$. That is, in eq.(16), $\phi_1 = [0,0,\dots,0,1,0,\dots,0]'$ and $\phi_2 = \phi_3 = \dots = \phi_T = 0$. The kernel or Green's function $g(\mathbf{r},t,\mathbf{r}',t')=g(i,j,t)$, with $\mathbf{r}=(i,j)$, \mathbf{r}' fixed (antenna point) and $t'=1$ fixed, is the result of the FDTD simulation, and for any general current source $J(i,j)=f(t) = f(n\Delta t)$ at some point (i,j) we have:

$$E_z(i\Delta x, j\Delta y, n\Delta t) = \sum_{v=1}^n g(i, j, v) f(n+1-v) \dots\dots\dots (25)$$

It is true that the approximation is not the correct Green's function, but if we calculate this approximation using a unitary pulse and FDTD, the computation of any input function $f(n\Delta t)$ using eq. (25) is equivalent to the FDTD simulation using directly $f(n\Delta t)$, for the case of study (linear, isotropic nondispersive materials). Because the FDTD equations are linear, we can use the next decomposition formula:

$$\left. \begin{aligned} AX &= b \Leftrightarrow AX_I = b_I, AX_{II} = b_{II}, \\ X &= X_I + X_{II}, b = b_I + b_{II} \end{aligned} \right\} \dots\dots\dots (26)$$

And by expressing the vector X as summatory of vectors $X^{(i)}$, the eq. (16) can be expressed as:

$$AX^{(i)} = (0, \dots, e, \dots, 0)' \Leftrightarrow AX^{(1)} = (e, 0, \dots)' \dots\dots\dots (27)$$

$$X = \sum_{v=1}^T \phi(v-1) X^{(v)} \dots\dots\dots (28)$$

Where $e=(0,0,\dots,1,0,\dots,0)$, and the only value "1" corresponding to the position of the antenna. This relation is independent of the signal $f(n\Delta t)$, with $\phi_i = [0,0,\dots, f(n\Delta t), \dots, 0]'$. In other words, there are two errors that are present in the approximations: the error in approximating Green's function and the error in using a summatory as eq. (25) instead of an integral. But these two errors cancels mutually as shown in eq. (26)~(28).

There is also some research which validates the use of FDTD-computed impulse responses and summatory to approximate the complete numerical solution using FDTD (11) using less accurate approximations. They use the Green's functions in 54 points as approximated incident field instead of the complete incident field, but the final result is very close to FDTD results for frequencies around 835 MHz. Our approximation is better because we use the Green's function over all the cells at free space, and exact substitution in other points.

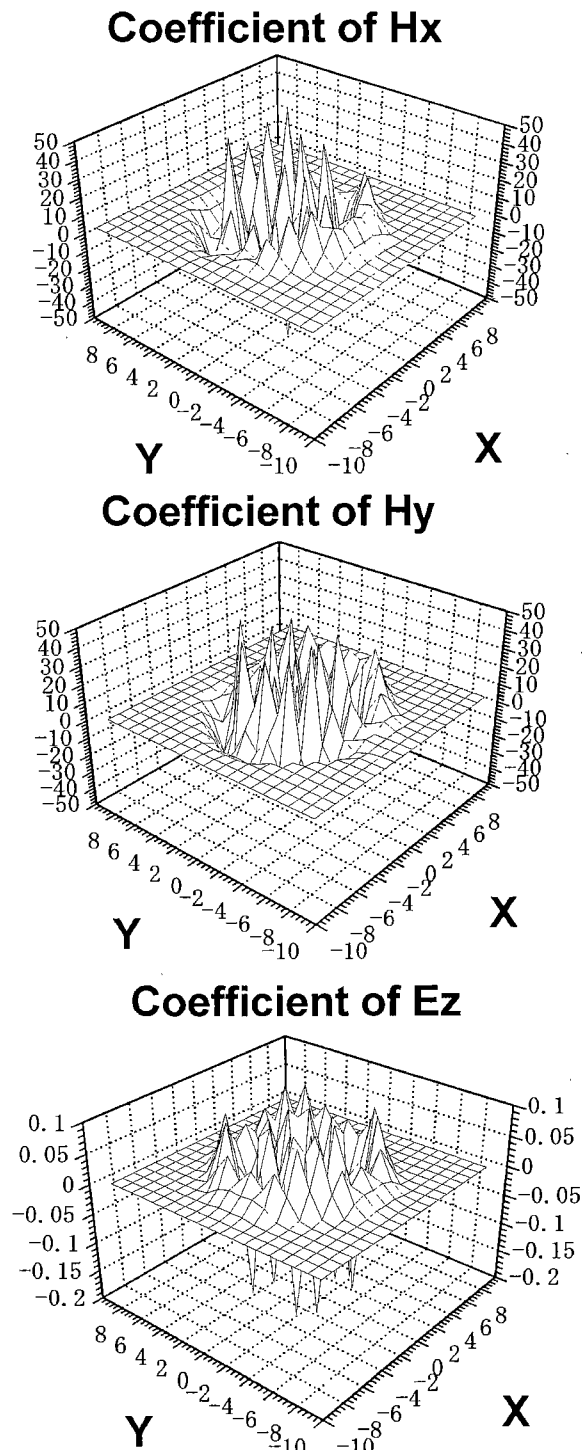


Fig. 3. Kernel functions $g_1, g_2, g_3(\mathbf{r}' - \mathbf{r})$ for E_z when $t'=t-10$

The advantage of using the impulse response is that in eq.(16), the constant matrix b have one value equal to 1, and 0 elsewhere. The linear equation becomes simpler. The physical meaning of $g(i,j,t)$ is that they are the influence in some variable $X(x=a,y=b,time=c)$ of the field components at time $c-t$ and position $(x-i,y-j)$. Mathematically, they are the coefficients c_i^N of the system of linear equations shown in eq.(19). In this equation, $b_c^N = 0$ if there is no antenna in the vicinity of X . This is a realistic assumption for urban communication models. There is 3 different kinds of g functions, g_1 , g_2 and g_3 , for the influence of each component H_x , H_y and E_z respectively, as seen in Fig. 3.

By the physical meaning of this functions, we can see that the values of $g(i,j,t)$ depends only on the materials and sources around variable X , in a vicinity of "t" cells, because in FDTD, the values of the fields propagates no faster than 1 cell each time step. The main material in an urban communication channel is free space. Most of the propagation is in free space, except the walls of the buildings. Then, most of the values of $g(i,j,t)$ and c_i^N will be the same.

At the start, we calculate g_1 , g_2 and g_3 for a free space environment with no sources and we store it in memory. This process is done only once, using the standard FDTD method. The improvement into the algorithm consist in checking the environments of each variable, and if there is only free space and no sources, then instead of doing Gaussian back-substitution, we used the stored values of g_1 , g_2 and g_3 as coefficients and $b_c^N = 0$. Otherwise, we use the Gaussian substitution process. This modification is shown in Fig. 4. The functions g_1 , g_2 and g_3 are also different for each type of dependent variable x_L . To obeying eq. (21), it is only required that the stored coefficients are in ascendent order.

The arrays for sparse matrix structure here are sp and isp . Array sp stores the values of the coefficients of matrix A , and isp stores the columns. The pre-calculated coefficients are stored in array $Value$ and the relative column displacements are stored in $relat$. Variable pos is the column of the variable x . Variable typ is the previous defined $c=0$ for H_x , $c=1$ for H_y , $c=2$ for E_z . The condition regarding the vicinity excludes also points near the

boundary of the simulation.

An important consequence of eq. (15) for impulse response case is the following equation:

$$X_{t+1} = A_1^t \phi_1 \dots \dots \dots (29)$$

From this equation, the impulse response (stored coefficients) after t time steps is A_1^t . The coefficients associated with $g(i,j,t)$ are the elements of the P -th row of matrix A_1^t , where P is the index associated with the variable H_x , H_y or E_z at position (i,j) and time t , that is $P=(t-1)(3pq)+3qi+3j+c$. In Fig. 5 is shown the sparsity of matrix A_1^{10} for a model of size 28×28 in free space. The power of matrix A_1 becomes less sparse, and the rows are the coefficients we should store.

The evaluation of the first condition in Fig. 4 is simple to evaluate for scatterers with simple geometries. In the case of buildings, the shapes are squares or rectangles. The evaluation of proximity to boundary or antennas are also quite simple. It adds a negligible step to the algorithm. But the savings of process for a urban communication channel simulation are very big. In a model of size $10,000 \times 10,000$, with 9 buildings, for $N=100$, form a

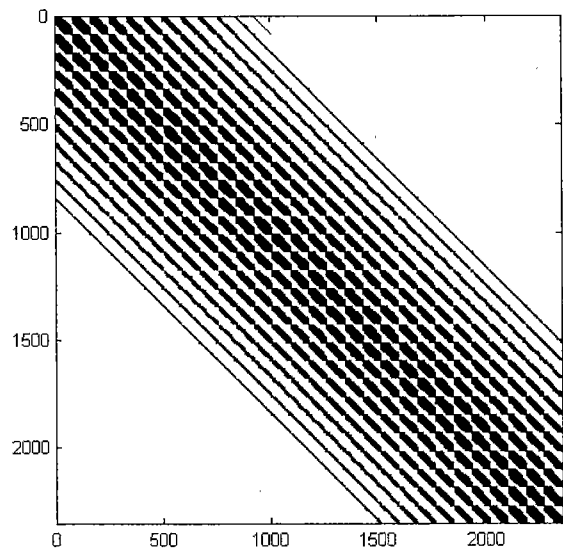


Fig. 5. Sparsity of matrix A_1^{10}

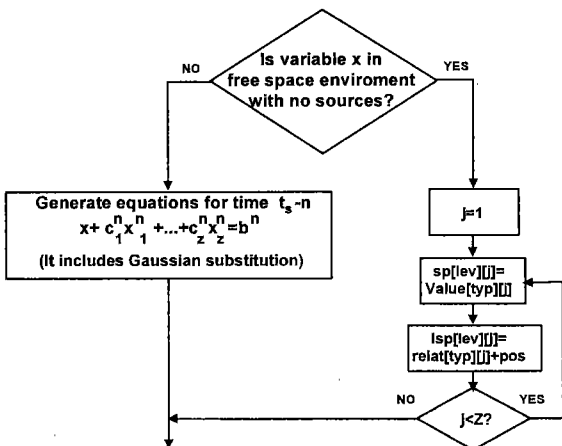


Fig. 4. Improvement in the algorithm

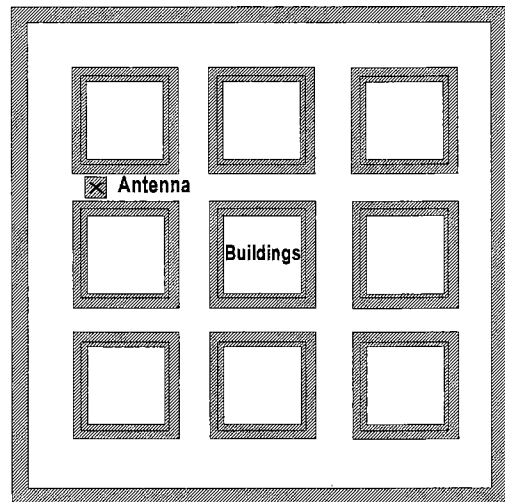


Fig. 6. White areas use faster calculation

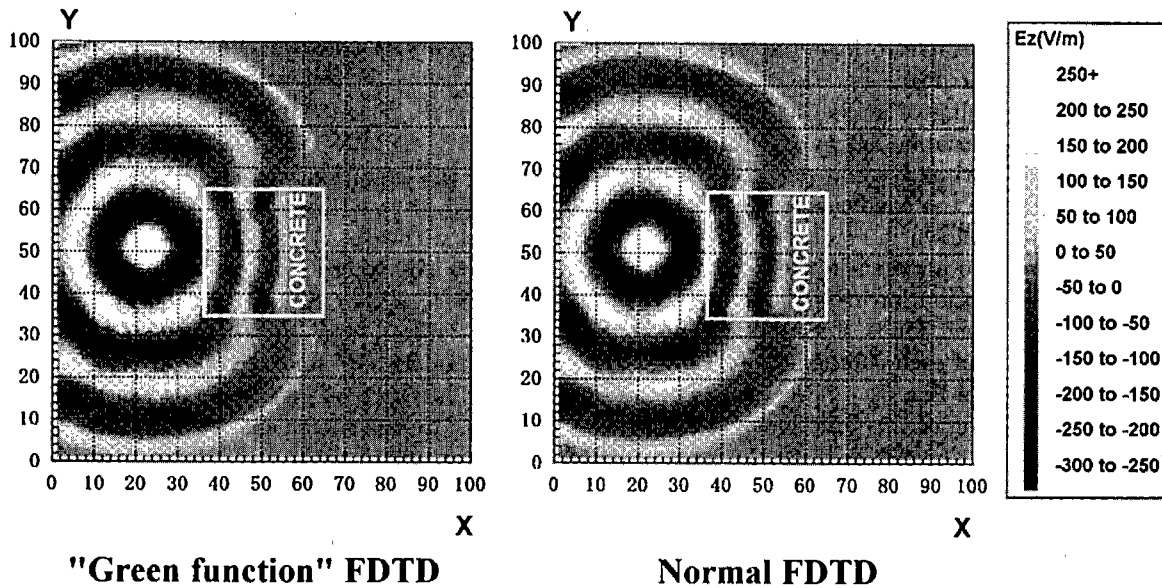


Fig. 7. Comparison between parallel FDTD using Green's function and non parallel FDTD without Green's function, $t=90\Delta t$

total of 10^8 cells, only around 12 million cells=12% need to run the Gaussian back-substitution, the other 88% can run the faster direct substitution using stored values. For a model of size $100,000 \times 100,000$, $n=100$, with 36 buildings, the number of cells which needs Gaussian substitution is around 203 millions=2.03% of the cells. The value $N=100$ is a good value to run the simulation using 4.5 Gb of memory. A sketch of this is shown in Fig. 6, where shaded areas use Gaussian substitution and white areas use faster, stored values.

In a practical point of view, the use of Green's function allow us to advance the simulation not in 1 time step, but instead in many time steps in many regions of the simulation.

4. Results

To test the accuracy of the convolution equation (25) as a way to calculated electromagnetic fields using only impulse response in FDTD and then convolution. The model is the same small model used in (1), with propagation from the antenna at point (20,50) emitting a signal like eq. (30), $f=950$ MHz, $J_{max}^i = 1000$, $t_{max}=30\Delta t=1.131$ ns, and a block of concrete in the center with parameters are:

- Frequency of source : 950 MHz
- Cell size, δ : 0.02 m
- Time increment Δt : 37.7 ps
- Relative permittivity(concrete): 3.0
- Conductivity of concrete, σ : 0.005 S/m
- Current amplitude: 1000.0 A/m

$$J_z^{i,n} = J_{max}^i \sin 2\pi ft; \text{ for } t < t_{max} \dots \dots \dots (30)$$

Fig.7 shows the calculated E_z at $t=90\Delta t$ using the normal FDTD method and the Convolution after impulse response FDTD. The convolution of a discrete Green's function after 90 time steps=3.393 ns is quite accurate to reproduce the resultant field when convoluting with the signal of eq. (30). For our purpose, it proves that we can use the impulse response stored values for $N=90$. As our planned N for very big models could be

Table 1. Comparison table for parallel FDTD algorithms

Category	Other parallel algorithms	This algorithm
Space size	Small, medium size: possible Large size: Not possible	Large size : possible
Processing time	Small, medium size: fast Large size: Not possible	Small size: fast Medium size: less fast Large size: slow

$N=50$ or $N=100$, the results should be accurate.

We have run the improved algorithm in a test case for a 1000×1000 model in free space. There is only one antenna is located in position (100,100). A simulation process that takes 70.1 minutes running in 10 processors with the original algorithm now takes only 5.2 minutes with the new one = 7.42% of previous time. The value of $N=10$, and the evaluated variable was $E_z(110,110)$. The reduction is significant because this variable is near the antenna source, and many variables in the calculation will not qualify for the faster process. For other locations, the process was a bit faster but the balance was very bad, because some processors got cells far from the antenna and other processors got cells near the antenna, so the result is not significant to evaluate the improvement. The stored coefficients were pre-calculated, stored in a file and then read for the algorithm. The next table shows the actual state of this research compared to other parallel FDTD algorithms.

The source for the information in the first column comes from ref. (4), (5), (9). In the standard FDTD parallel methods mentioned there, the space size of the simulation is severely limited for the total available memory.

The second example is a medium size model, more realistic, of size $80m \times 80m$, with the antenna in position (11m,55m) after $5000\Delta t=188.5$ ns. The urban area has 9 shallow buildings with concrete walls of width 30 cm.

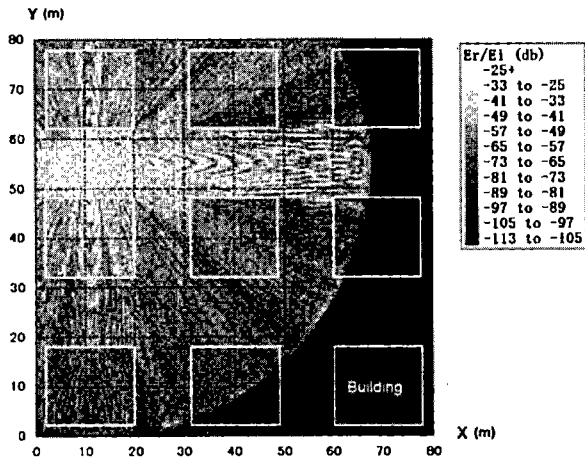


Fig. 8. $|E_r/E_i|$ in decibels. Medium size model (80m \times 80m), $t=5000\Delta t=188.5$ ns

The size of each building is 18 m \times 16 m. Others parameters are the same as at the first example. Fig. 8 shows the ratio $|E_r/E_i|$ in decibels, where E_r is the value of E_z at any point and E_i is the maximum value of E_z at the antenna point. It can be appreciated that the behavior of the waves is similar to the behavior obtained by Uniform theory of diffraction (ray tracing).

5. Conclusion

Research on the use of impulse response in this algorithm is being done. The use of impulse response has a physical equivalent but also a mathematical one in our system of linear equations. They are the coefficients of the partial results of Gaussian substitution. In the case of big urban environments, most of the calculation is done around free space, therefore, more of the time the coefficients are the same. The parallel processing is done in both cases: linear equations or impulse response coefficients, without further change in the load balance of the processors. By calculating the coefficients only once and then using it when possible, we are increasing the speed of a medium size simulation using parallel processing by a factor of 13 for small problems, and now we can obtain the same speed than non parallel algorithms for medium size problems (80m \times 80m). We expect that this process will be fast enough in biggest problems. The merits of the algorithm researched until now are the need of less memory and a processing speed comparable with the speed of non parallel FDTD for medium size problems. Our next research will be the solution of the growing complexity of the algorithm for very large problems with long simulation times. Only then, combined with these improvements, we should be able to carry out the huge simulations of wave propagation in urban areas.

(Manuscript received February 19, 2002, revised July 3, 2002)

References

- (1) G.Rodriguez and Y.Miyazaki : "Electromagnetic Field Analysis using a Parallel FDTD Algorithm in Extremely Large Areas for Mobile Communication", *T.IEE Japan*, **121-C**, No.12,

- pp.1826-1833 (2001-12)
- (2) G.Rodriguez and Y.Miyazaki : "Analysis of Propagation Characteristics on Big Urban Areas using a Parallel FDTD Algorithm", Papers of Tech. Meeting on EMT, IEE Japan, EMT-01-62-79, pp.13-18 (2001)
- (3) P. Selormey and Y. Miyazaki : "Electromagnetic Compatibility Characteristics of Buildings in Mobile Radio Waves Propagation Channel", *T.IEE Japan*, **119-C**, No.1, pp.97-104 (1999-1)
- (4) U.Anderson : "Parallelization of a 3D FD-TD code", Proc. 4th International Workshop of Applied and Parallel Comp. '98, LNCS, pp.12-19 (1998)
- (5) A. Tafflove : Computational Electrodynamics, 1st edition, pp.545-581, Artech House, Boston (1995)
- (6) A.Fijany, M.Jensen, et. al. : "A massive Parallel Computation Strategy for FDTD", *IEEE Trans. Ant. & Prop.*, **43**, No.12, pp.1441-1449 (1995-12)
- (7) G.Rodriguez and Y.Miyazaki : "Analysis of Electromagnetic Scattering in Large Areas using a Parallel FDTD method", Proc. OFSET 2000, pp.163-166 (2000-12)
- (8) E.Miller : "Solving bigger problems- By Decreasing the operation count and Increasing the Computational Bandwidth", *Proc. IEEE*, **79**, No.10, pp.1493-1504 (1991-10)
- (9) Y.Lu and C.Shen : "A Domain Decomposition Finite-Difference Method for Parallel Numerical Implementation of Time-Dependent Maxwell's Equations", *IEEE Trans. Ant. & Prop.*, **45**, No.3, pp.556-562 (1997-3)
- (10) M.Osano, K.Nakajima, and M.Tanimoto : "A Partially Solving Method (PSM): A new efficient method for a large system of linear equations", *Bulletin Electrotechnical Laboratory*, **60**, No.2, pp.93-101 (1996-2)
- (11) G.Lazzi and O.Gandhi : "A Mixed FDTD-Integral Equation Approach for On-site Safety Assesment in Complex Electromagnetic Environments", *IEEE Trans. Ant. & Prop.*, **48**, No.12, pp.1830-1836 (2000-12)

Glen Rodriguez (Non-member) Was born in Lima, Peru, on March 27, 1974. He received the B.Sc. degree in Systems Engineering from National University of Engineering, Peru, in 1994, and the M.E. degree in Information and Computer Science Engineering from Toyohashi University of Technology in 2001. He is currently working towards the D.E. degree in Electronic and Information Engineering. His research interest are Parallel Processing, Mobile Communication and Electromagnetic Waves Propagation and Scattering.



Yasumitsu Miyazaki (Member) Was born in Nagoya, Japan, on February 4, 1941. He received the B.E. M.E. and D.E. degrees in Electronic Engineering from Nagoya University, Nagoya, in 1963, 1965 and 1969, respectively. He has engaged in research in the field of electromagnetic waves including waves in millimeter waveguides, optical fibers, integrated optics, electromagnetic scattering and diffraction. He also studies biological phenomena of electromagnetic fields and optical neural computing. He is presently Professor of the Department of Information Engineering, Toyohashi University of Technology, since 1981. From January 1973 to January 1975, he stayed in Institute of High Frequency-Technics, Technical University of Braunschweig, West Germany, and engaged in research on electromagnetic fields for optical communications. In 1996, he was a guest Professor in IHFT of Technical University of Berlin, Germany. He received Yonezawa Memorial Paper award in 1970. He is a member of IEEE, IEE of Japan, Japanese Society of Medical Electronics and Biological Engineering and the Japan Society of Applied Physics.