Chapter 2 THEORY AND METHODOLOGY

The finite-difference time-domain method was first presented by K. S. Yee in 1966[12]. Yee's insight was to choose a geometric relation for his spatial sampling of the vector components of the electric and magnetic fields that robustly represents both the differential and integral forms of Maxwell's equations. Therefore, the structure that we want to discuss need to be divide into many small regions, which is called "cell". This can be showed with Figure 2-1.

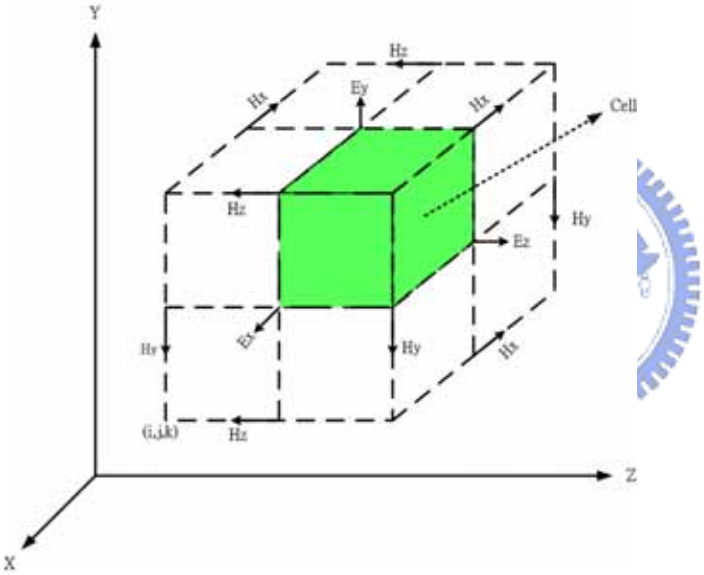

Fig. 2-1 The illustration of Yee's FDTD method.

2-1 One-Dimensional Free Space Formulations

The propagation of light was controlled by the four Maxwell equations. Here, we just need the two curl equations:

$$\frac{\partial D}{\partial t} = \nabla \times H \tag{2.1}$$

$$\frac{\partial H}{\partial t} = -\frac{1}{\mu_0} \nabla \times E \tag{2.2}$$

$$D(\omega) = \varepsilon_0 \cdot \varepsilon_r^*(\omega) \cdot E \tag{2.3}$$

where D is the electric flux density, E is the electric field, H is the magnetic field, ε_0 is the dielectric constant in free space and $\varepsilon_r^*(\omega)$ is the relative dielectric constant. We will assume we are dealing with a lossy dielectric medium of the form

$$\varepsilon_r^*(\omega) = \varepsilon_r + \frac{\sigma}{j\omega\varepsilon_0} \tag{2.4}$$

Here, Eq.(2.4) is written in frequency domain since we may need to simulate frequency-dependent material. For the simplicity in the formulations, we will normalize the Maxwell equations. Let

$$E' = \sqrt{\frac{\varepsilon_0}{\mu_0}} \cdot E \tag{2.5}$$

$$\mathrm{D}' = \sqrt{\frac{1}{\varepsilon_0\mu_0}} \cdot D \tag{2.6}$$

here we can find that the E field and the H field will have the same order of magnitude. This system is called *Gaussian units*. Then, we can rewrite Eqs.(2.1), (2.2) and (2.3):

$$\frac{\partial D'}{\partial t} = \frac{1}{\sqrt{\varepsilon_0\mu_0}} \nabla \times H \tag{2.7}$$

$$\frac{\partial H}{\partial t} = -\frac{1}{\sqrt{\varepsilon_0\mu_0}} E' \tag{2.8}$$

$$D'(\omega) = \varepsilon_r^*(\omega) \cdot E'(\omega) \tag{2.9}$$

Now, Eqs. (2.7) and (2.8) can be derived to the simple finite difference equations. In general, the FDTD method uses the first order difference. If we need the higher order of accuracy, we can use the high order difference approximation. But, the high order difference approximation is more complex and unstable. So, we use the first order difference here. The first order difference has three types: front difference approximation, back difference approximation and central difference approximation. Both of the error range of front and back difference approximation are Δz , and the error range of central difference approximation is $(\Delta z)^2$. So, the FDTD method

usually uses the central difference method for higher accuracy.

If we consider a plane wave travels in the z direction with E field oriented in the x direction and the H field oriented in the y direction, the central difference approximation for temporal and spatial derivatives of Eqs. (2.7) and (2.8) gives

$$\frac{D_x^{n+1/2}(m)-D_x^{n-1/2}(m)}{\Delta t}=-\frac{1}{\sqrt{\varepsilon_0\mu_0}}\frac{H_y^n(m+1/2)-H_y^n(m-1/2)}{\Delta z} \tag{2.10}$$

$$\frac{H_y^{n+1}(m+1/2)-H_y^n(m+1/2)}{\Delta t}=-\frac{1}{\sqrt{\varepsilon_0\mu_0}}\frac{E_x^{n+1/2}(m+1)-E_x^{n+1/2}(m)}{\Delta z} \tag{2.11}$$

(here, we let $E'=E$ and $D'=D$ for simplicity)where the superscripts "n" means a time $t=\Delta t\cdot n$ and the subscripts "m" means the distance $z=\Delta z\cdot m$. i.e, the term "n+1" means one time step later than "n", and the term "m+1" means one distance step next to "m". We should keep in mind that every parameter of formulations was needed to be distributed into the computer.

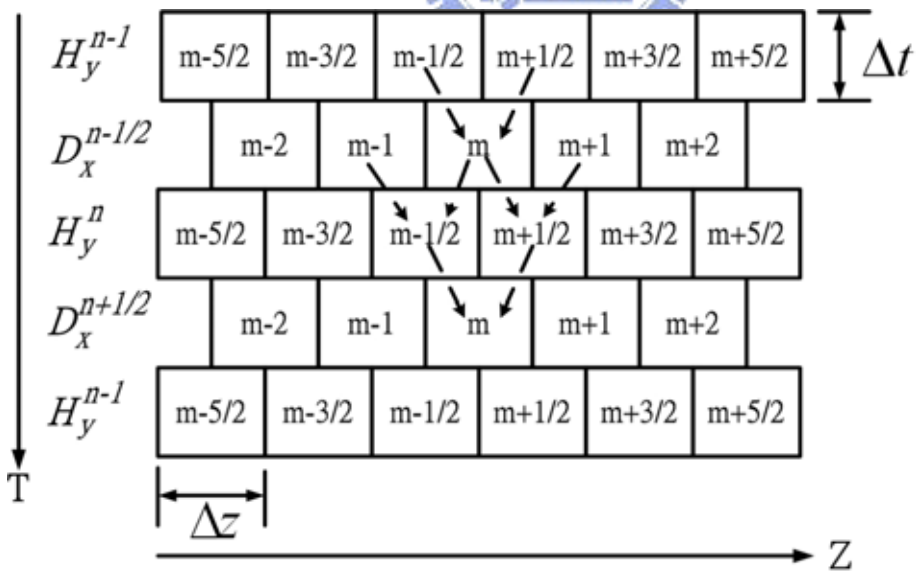

Fig. 2-2 Interleaving of the E and H fields in space and time in the FDTD formulation.

The formulation of Eq. (2.10) and (2.11) assumes that the D and H fields are

interleaved in both space and time. The subscripts "*m+1/2*" and "*m-1/2*" of *H* field mean that the *H* field values are assumed to be located on the both sides of the *D* field value with subscript "m" . Similarly, the superscripts "*n+1/2*" or "*n-1/2*" of *D* field value means that *D* field value is assumed to occur slightly after or before the *H* field value with superscript "n". This is illustrated in Fiq.2-2.

Rearrange Eqs. (2.10) and (2.11) in an iterative algorithm:

$$D_x^{n+1/2}(m) = D_x^{n-1/2}(m) - \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta z} [H_y^n(m+1/2) - H_y^n(m-1/2)] \quad (2.12)$$

$$H_y^{n+1}(m+1/2) = H_y^n(m+1/2) - \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta z} [E_x^{n+1/2}(m+1) - E_x^{n+1/2}(m)]. \quad (2.13)$$

We can find that the calculations are interleaved in both space and time. For example, in Eq. (2.12) the current D_x is calculated from previous value of D_x and the most recent values of H_y.

Now, we should determine the time step- Δt . Since the speed of an electromagnetic wave propagating in free space isn't faster than the speed of light, we should have a relation between Δx and Δt. This relation is the well-know "Courant Condition"[13][14]:

$$\Delta t \le \frac{\Delta z}{\sqrt{n} \cdot c_0} \quad (2.14)$$

where *n* is the dimension of the simulation and c_0 is the speed of light in free space.. From this condition, once the cell size Δz is chosen, the time step Δt is determined by

$$\Delta t = \frac{\Delta z}{u \cdot c_0}, \quad (2.15)$$

where u is a constant dependent on the source that we choose. Therefore,

$$\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta z} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \cdot \frac{1}{\Delta z} \cdot \frac{\Delta z}{u \cdot c_0} = \frac{1}{u} \quad (2.16)$$

and rewriting Eqs. (2.12) and (2.13) :

$$D_x^{n+1/2}(m) = D_x^{n-1/2}(m) - \frac{1}{u}[H_y^n(m+1/2) - H_y^n(m-1/2)] \tag{2.17}$$

$$H_y^{n+1}(m+1/2) = H_y^n(m+1/2) - \frac{1}{u}[E_x^{n+1/2}(m+1) - E_x^{n+1/2}(m)]. \tag{2.18}$$

And then, we need to get Eq.(2.3) into a time domain difference equation for implementation into FDTD method. We substitute Eq.(2.4) into Eq.(2.3)

$$D(\omega) = \varepsilon_r \cdot E(\omega) + \frac{\sigma}{j\omega\varepsilon_0} E(\omega). \tag{2.19}$$

Since the first term on the right hand side is simple multiplication, we take it into the time domain directly. In the second term on the right hand side, we know that *1/jω* in the frequency domain is integration in the time domain from the Fourier theory, so Eq.(2.19) becomes

$$D(t) = \varepsilon_r \cdot E'(t) + \frac{\sigma}{\varepsilon_0}\int_0^t E'(t') \cdot dt'. \tag{2.20}$$

Since we need to go to the sampled time domain, the integral will be approximated as a summation over the time steps Δt:

$$D^n = \varepsilon_r \cdot E^n + \frac{\sigma \cdot \Delta t}{\varepsilon_0}\sum_{i=0}^{n} E^i. \tag{2.21}$$

In Eq.(2.21), the E and D are specified at time $t = n\cdot\ \Delta t$. We can find that we need to solve E^n given the value D^n, but the value E^n is needed in the calculation of the summation. So, we separate the E^n from the rest of the summation:

$$D^n = \varepsilon_r \cdot E^n + \frac{\sigma \cdot \Delta t}{\varepsilon_0} E^n + \frac{\sigma \cdot \Delta t}{\varepsilon_0}\sum_{i=0}^{n-1} E^i. \tag{2.22}$$

And then,

$$E^n = \frac{D^n - \frac{\sigma \cdot \Delta t}{\varepsilon_0}\sum_{i=0}^{n-1} E^i}{\varepsilon_r + \frac{\sigma \cdot \Delta t}{\varepsilon_0}}. \tag{2.23}$$

Now, we can calculate the value E of the current time step from the value D of the current step and the value E of the previous time step. It is convenient to define a new parameter in the calculation

$$I^n = \frac{\sigma \cdot \Delta t}{\varepsilon_0} \sum_{i=0}^{n-1} E^i \tag{2.24}$$

and then, we can reformulate Eq.(2.23) with two equations:

$$E^n = \frac{D^n - I^{n-1}}{\varepsilon_r + \frac{\sigma \cdot \Delta t}{\varepsilon_0}} \tag{2.25}$$

$$I^n = I^{n-1} + \frac{\sigma \cdot \Delta t}{\varepsilon_0} E^n \tag{2.26}$$

So, we can summarize the equations we will use in C++ computer code and arrange them in sequence of C++ computer code:

$$D_x^{n+1/2}(m) = D_x^{n-1/2}(m) - \frac{1}{u}[H_y^n(m+1/2) - H_y^n(m-1/2)] \tag{2.27}$$

$$E^n = \frac{D^n - I^{n-1}}{\varepsilon_r + \frac{\sigma \cdot \Delta t}{\varepsilon_0}} \tag{2.28}$$

$$I^n = I^{n-1} + \frac{\sigma \cdot \Delta t}{\varepsilon_0} E^n \tag{2.29}$$

$$H_y^{n+1}(m+1/2) = H_y^n(m+1/2) - \frac{1}{u}[E_x^{n+1/2}(m+1) - E_x^{n+1/2}(m)]. \tag{2.30}$$

2-2 Absorbing Boundary Condition

When we solve the electromagnetic wave interaction problems with the FDTD method, many geometries of interest are defined in “open” regions. The spatial domain of the computed field is unbounded in one or more coordinate directions, but no computer can store an unlimited amount of data. Therefore, we need to limit the computation region in size. The computation region must enclose the structure of interest, and a suitable boundary condition on the outer perimeter of the domain must be used to simulate its extension to infinity.

For example, we consider that a pulse propagates in the free space in one dimension, and a FDTD computation region has 200 grids. If the pulse is generated at

the grid 100, the pulse will propagate toward both right and left side. When the pulse hits the both ends of the computation region, the pulse will reflect back into the problem space (Figure 2-3). This is wrong! Since we know that there are no sources outside the problem space and the fields at the edge must be propagating outward.

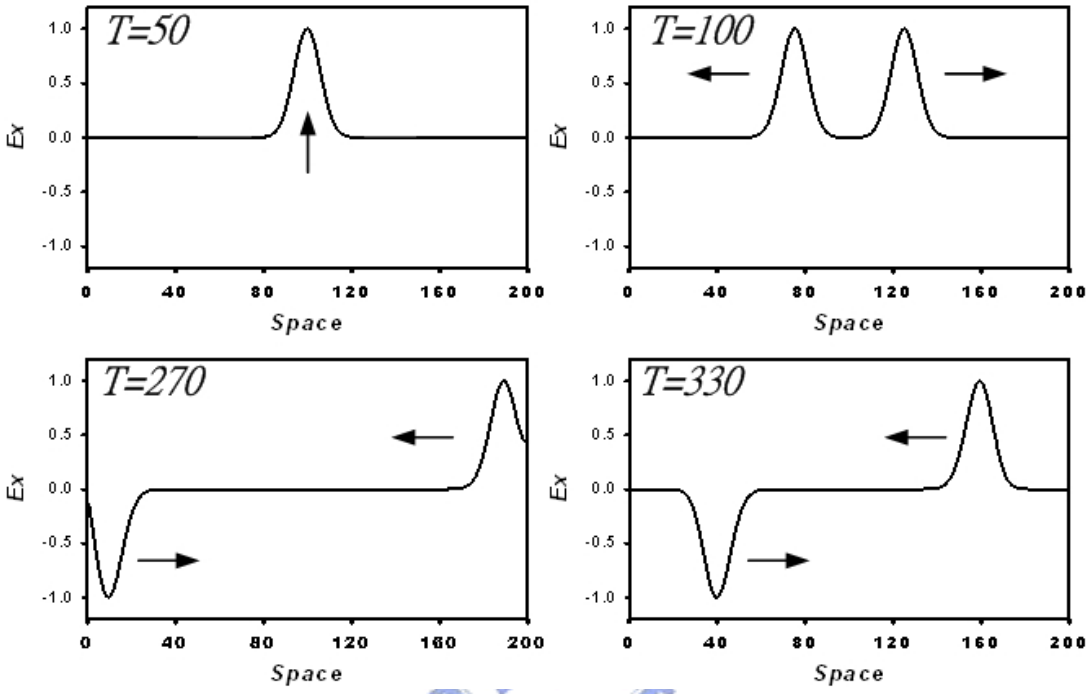

Fig.2-3 1D pulse simulation without ABCs.

Therefore, a boundary condition was needed to permit all outward-propagating waves to exit the both ends of computation region as if the simulation were performed on a computational domain of infinite extent. In the process, the outer-boundary condition must suppress spurious reflections of the outgoing waves to an acceptable level, permitting the FDTD solution to remain valid for all time-steps, especially after the reflected waves return to the vicinity if the modeled structure. The outer-boundary conditions of this type have been called absorbing boundary conditions (ABCs). There are several types of ABCs. Here, we use the Mur's ABCs to be the outer-boundary condition in the 1D-PC simulation.

Suppose that there is a plane wave which propagates in the *z*-direction and

incidents into the absorbing boundary at $z=0$. The E field is in x-direction and the H field is in the y-direction. The propagating speed of the plane wave is v. So, the expression of E_x is

$$E_x = E_x(z+vt) \tag{2.31}$$

and Eq. (2.31) need to satisfy the differential equation

$$\frac{\partial E_x}{\partial z} - \frac{1}{v}\frac{\partial E_x}{\partial t} = 0. \tag{2.32}$$

Since the absorbing boundary make no reflection at $z=0$, the E_x propagates with the shape of Eq. (2.31) and satisfies the Eq. (2.32) at $z<0$. ie. no waves propagate in the $+z$-direction. So, Eq. (2.32) can be treated as the simplest one-way wave equation. We derive the Eq. (2.32) to the finite difference and use the Yee's algorithm:

$$\frac{E_x^n(1/2) - E_x^{n-1}(1/2)}{\Delta t} = v\frac{E_x^{n-1/2}(1) - E_x^{n-1/2}(0)}{\Delta z} \tag{2.33}$$

where n is the index of time and just like the time-index of the equations that we derived before. Since the $E_x^n(1/2)$ on the left-hand side of Eq. (2.33) and the $E_x^{n-1/2}$ on the right-hand side of Eq. (2.33) are not on the point of the grid of FDTD, we can get the approximate value from the average value of the points on the both sides of time and spatial domain:

$$E_x^n(1/2) = \frac{E_x^n(0) + E_x^n(1)}{2} \tag{2.34}$$

$$E_x^{n-1/2}(1) = \frac{E_x^n(1) - E_x^{n-1}(1)}{2} \tag{2.35}$$

Substitute Eqs. (2.34) and (2.35) into Eq. (2.33) and get the solution of $E_x^n(0)$:

$$E_x^n(0) = E_x^{n-1}(1) - \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z}[E_x^{n-1}(0) - E_x^n(1)] \tag{2.36}$$

Similarly, the solution of E_x^n on another end of computation region (ME):

$$E_x^n(ME) = E_x^{n-1}(ME-1) - \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z}[E_x^{n-1}(ME) - E_x^n(ME-1)] \tag{2.37}$$

We can simplify Eqs. (2.36) and (2.37) by substituting Eq. (2.15) into this two

equations

$$\frac{v\Delta t-\Delta z}{v\Delta t+\Delta z}=\frac{v\cdot\dfrac{\Delta z}{u\cdot v}-\Delta z}{v\cdot\dfrac{\Delta z}{u\cdot v}+\Delta z}=\frac{\Delta z-u\cdot\Delta z}{\Delta z+u\cdot\Delta z}=\frac{1-u}{1+u} \tag{2.38}$$

and we can rewriting the Eqs. (2.37) and (2.38):

$$E_x^n(0)=E_x^{n-1}(1)-\frac{1-u}{1+u}[E_x^{n-1}(0)-E_x^n(1)] \tag{2.39}$$

$$E_x^n(ME)=E_x^{n-1}(ME-1)-\frac{1-u}{1+u}[E_x^{n-1}(ME)-E_x^n(ME-1)] \tag{2.40}$$

For 1D-PC, we can apply Eqs. (2.39) and (2.40) on the both ends of computation region, and the reflect waves will vanish (Figure 2-4).

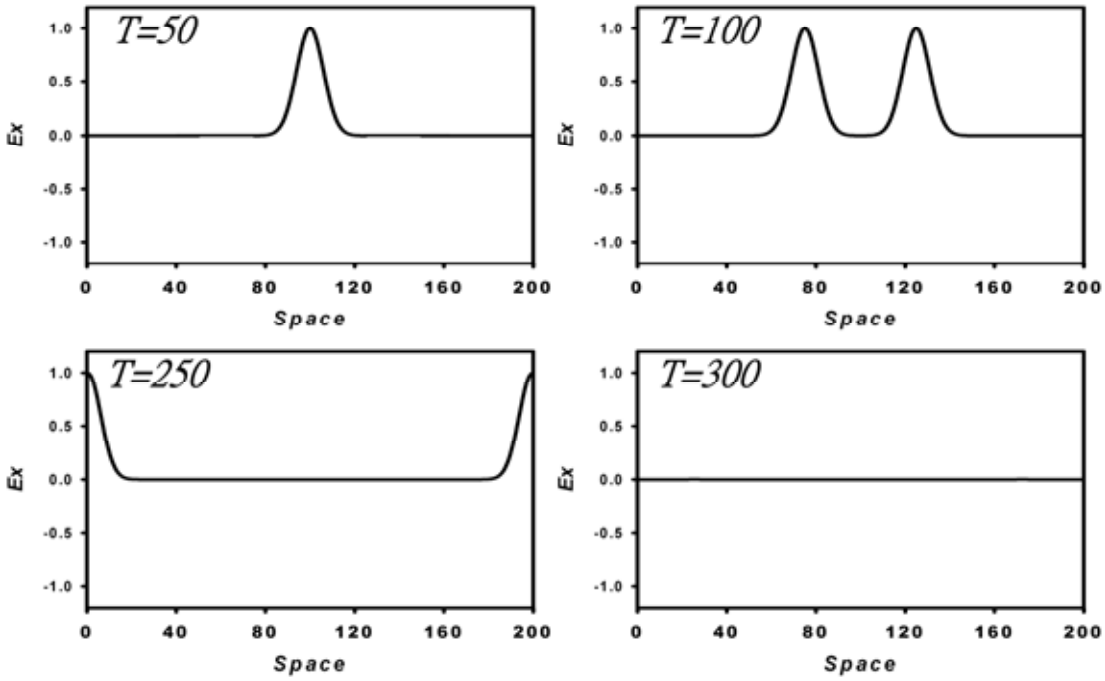

Fig. 2-4 1D Pulse simulation with ABCs.