Transactions of the Nigerian Association of Mathematical Physics
Volume 1, (November, 2015), **pp 179 – 200**
© Trans. of NAMP

Finite Difference Time Domain Method

Eghuanoye Ikata

Physics Department, Ignatius Ajuru University of Education, Rumuolumeni, P. M. B. 5047, Port Harcourt.

Abstract

This discourse on the finite difference time domain method is pedagogic, and is intended to inculcate the basic knowledge required to use the method. The primary concepts and ideas which make the method work, such as, finite difference, discretizaton, initialization and lattice truncation, are explained. Finite difference time domain algorithms for acoustic wave and electromagnetic wave are derived, showing how the method may be applied to other wave types. This is extended to the alternating direction implicit finite difference time domain method. The conditions for numerical stability of a difference algorithm are obtained using von Neumann method.

1.0 Introduction

Improvements in the accessibility and capability of the digital computer have encouraged widespread use of the finite difference time domain (FDTD) method, introduced by Kane S Yee in 1966 [1], for solving wave interaction problems. The FDTD method is an adaptation of the *leap-frog* scheme [2] for the numerical solution of *hyperbolic* partial differential equation. This procedure is currently well developed [3,4] and quite suited for modelling various wave interaction problems in which the time parameter appears explicitly.
Herein we present the primary concepts and ideas of the FDTD method. The discourse is intended to inculcate the basic knowledge necessary for one to use the method. The discourse is divided into seven major sections, namely, mathematical background, physical background, traditional FDTD method, acoustic wave FDTD algorithm, ADI-FDTD method, numerical stability, and implementation issues.

1.1 Polynomial Approximation

The replacement of a function $y = f(x)$ by a simple expression proves extremely convenient in various problems of mathematical analysis such as evaluating integrals, solving differential equations and the likes. If the exact form of the function $f(x)$ is not known, or is known in a complicated form, then it is necessary to assume that the function $f(x)$ can always be taken as a polynomial in x. The validity of this assumption is guaranteed by Weierstrass' theorem [5], which states that if $f(x)$ is continuous in a finite interval $a \le x \le b$, then a sequence of polynomials $P_1(x), P_2(x), \ldots$ can be constructed which converge uniformly to $f(x)$ in the interval [a, b]. This implies that (i) $f(x)$ is infinitely differentiable in the interval, and (ii) $f(x)$ can be represented as the sum of a power series. The existence of a series expansion is extremely important since it makes it possible to replace the given function, approximately, with a finite polynomial.

1.2 Taylor Series Expansion

If a function $f(x)$ can be expanded into a series in powers of the difference $\Delta x(= x - x_0)$, this series is uniquely determined and is the Taylor series of the function in the neighbourhood of a point x_0; namely,

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(\Delta x)^n , \qquad (1.1)$$

Corresponding author: Eghuanoye Ikata, Tel.: +2348057308476

where $f^{(n)}(x_0) = d^n f / dx^n$ and n! is '*factorial* n'. A partial sum of this series is called a Taylor polynomial approximation of the function *f(x).*
Let $P_n(x)$, the n*th* partial sum of the Taylor series, denote a Taylor polynomial of the n*th* degree:

$$P_n(x) = f(x_0) + f^1(x_0)\Delta x + \ldots + \frac{f^n(x_0)}{n!}(\Delta x)^n . \tag{1.2}$$

Putting $R_n(x) = f(x) – P_n(x)$, equation (1.1) can be written as

$$f(x) = P_n(x) + R_n(x) . \tag{1.3}$$

Comparing equation (1.3) with equation (1.1) shows that replacing the Taylor series with a Taylor polynomial introduces a truncation error $R_n(x)$, the remainder after n terms in a Taylor series.

1.3 Finite Difference Approximation of Derivatives

Through the use of Taylor series we can obtain approximate expressions for derivatives. Suppose the Taylor polynomial with remainder (after the first term),

$$f(x) = f(x_0 + \Delta x) = f(x_0) + \Delta x \frac{df(x_0)}{dx} + \frac{(\Delta x)^2}{2!}\frac{d^2 f(\xi)}{dx^2} , \tag{1.4}$$

often called the extended mean value theorem. From equation (1.4) we can write

$$\frac{df(x_0)}{dx} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{\Delta x}{2}\frac{d^2 f(\xi)}{dx^2} . \tag{1.5}$$

Thus an approximate expression of df/dx, the forward difference approximation, can be written as

$$\frac{df(x)}{dx} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} . \tag{1.6}$$

Since the equations above are valid for any increment, we can replace Δx with $-\Delta x$ to obtain the backward difference approximation of *df/dx*:

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} . \tag{1.7}$$

The numerator on the right-hand-side in equations (1.6) and (1.7) is called the first finite difference, $\Delta f(x)$.
If Δx is small (as is usual), then ξ in equation (1.5) is contained in a small interval $x_0 < \xi < x_0 + \Delta x$, and the truncation error in equation (1.4) is approximately

$$R \approx \frac{(\Delta x)^2}{2}\frac{d^2 f(x_0)}{dx^2} .$$

Similarly, the truncation error in the finite difference approximations (1.6) and (1.7) of the first derivative is about

$$|R| \approx \frac{\Delta x}{2}\frac{d^2 f(x_0)}{dx^2} . \tag{1.8}$$

We say that the truncation error in the forward difference and backward difference approximations is of "order delta-x", $\vartheta(\Delta x)$. Strictly speaking, the error in equation (1.6) is not equal to the error in equation (1.7) since the points at which the error terms are to be evaluated are not identical; $x_0 < \xi_1 < x_0 + \Delta x$ and $x_0 - \Delta x < \xi_2 < x_0$. To obtain a better approximation of the first derivative, we take the average of the forward and backward difference approximations. Suppose the Taylor series:

$$f(x_0 + \Delta x) = f(x_0) + \Delta x \frac{df(x_0)}{dx} + \frac{(\Delta x)^2}{2!}\frac{d^2 f(x_0)}{dx^2} + \frac{(\Delta x)^3}{3!}\frac{d^3 f(x_0)}{dx^3} + \ldots$$

$$f(x_0 - \Delta x) = f(x_0) - \Delta x \frac{df(x_0)}{dx} + \frac{(\Delta x)^2}{2!}\frac{d^2 f(x_0)}{dx^2} - \frac{(\Delta x)^3}{3!}\frac{d^3 f(x_0)}{dx^3} + \ldots$$

Subtracting $f(x_0 - \Delta x)$ from $f(x_0 + \Delta x)$ gives

$$f(x_0 + \Delta x) - f(x_0 - \Delta x) = 2\Delta x \frac{df(x_0)}{dx} + \frac{(\Delta x)^3}{3!}\frac{d^3 f(x_0)}{dx^3} + \ldots$$

Consequently,

$$\frac{df(x_0)}{dx} = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} - \frac{(\Delta x)^2}{6}\frac{d^3 f(\xi)}{dx^3} + \ldots\ldots \tag{1.9}$$

Where $x_0 - \Delta x < \xi < x + \Delta x$. Thus we have the central difference approximation of the first derivative

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}, \tag{1.10}$$

having a truncation error of "order delta-x squared", $\vartheta(\Delta x)^2$, which is approximately

$$|R| \approx \frac{(\Delta x)^2}{6}\frac{d^3 f(x_0)}{dx^3}. \tag{1.11}$$

To avoid using double increments ($2\Delta x$) in the definition of the central difference, a half-index scheme is often used, which allows us to write

$$\frac{df(x_0)}{dx} \approx \frac{f\left(x_0 + \frac{1}{2}\Delta x\right) - f\left(x_0 - \frac{1}{2}\Delta x\right)}{\Delta x}. \tag{1.12}$$

By adding the Taylor series for $f(x_0 + \Delta x)$ to that for $f(x_0 - \Delta x)$ we obtain

$$f(x_0 + \Delta x) + f(x_0 - \Delta x) = 2f(x_0) + (\Delta x)^2 \frac{d^2 f(x_0)}{dx^2} + \frac{2(\Delta x)^4}{4!}\frac{d^4 f(x_0)}{dx^4} + \ldots$$

Consequently,

$$\frac{d^2 f(x_0)}{dx^2} = \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{(\Delta x)^2} - \frac{(\Delta x)^2}{12}\frac{d^4 f(\xi)}{dx^4} + \ldots \tag{1.13}$$

which leads to the central difference approximation of the second derivative

$$\frac{d^2 f(x_0)}{dx^2} \approx \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{(\Delta x)^2}, \tag{1.14}$$

with an error of "order delta-x squared", $\vartheta(\Delta x)^2$. The numerator on the right-hand-side of (1.14) is called a second finite difference $\Delta^2 f(x)$.

In solving partial differential equations, we analyze functions of two or more variables. Finite difference approximations of the partial derivatives involved in a partial differential equation may be obtained using our earlier results for functions of a single variable. All that is necessary is to assume, that while differentiating with respect to one variable all the other variables are fixed, thus allowing us to replace, for instance, $\partial f / \partial x$ with df/dx.

2.0 Physical Background

2.1 Scalar Wave

A small amplitude wave in a homogeneous, isotropic, linear, loss-less medium satisfies an equation (in three-dimensions) of the form

$$\nabla^2\psi - \frac{1}{\upsilon^2}\frac{\partial^2\psi}{\partial t^2} = 0, \tag{2.1}$$

where ψ is a scalar field function (applicable to the specific wave) and υ is the phase velocity. It is possible to describe a wave in terms of field functions, relevant to the specific situation, other than that appearing in equation (2.1). Ordinarily, the other field functions are apparent from the physics of a problem.

In equation (2.1) the word 'field' is used in the ordinary mathematical sense, which implies a function of space (and time), and not in the extended physical sense. In most situations of practical interest the wave fields are often force fields.

2.2 Acoustic Wave

An acoustic wave is a common example of a scalar field, and is completely described by specifying one parameter. It is a small disturbance in pressure and density which propagates in a compressible medium. Since fluids exhibit fewer restrains to deformation, the restoring force responsible for wave propagation is simply due to pressure change. The field functions which describe the wave sustained by a fluid are: the mass density $\rho(\mathbf{r},t)$, the particle velocity $\mathbf{V}(\mathbf{r},t)$, and acoustic pressure $P(\mathbf{r},t)$.

If the disturbance is assumed to be of small amplitude, it is possible to replace the equations which describe an acoustic wave with their linear equivalents [6]. Consequently, the linearized form of the equation of motion (in a lossless medium) is

$$\rho_o \frac{\partial \boldsymbol{V}}{\partial t} = -grad\ P, \tag{2.2a}$$

where ρ_o is a constant equilibrium mass density per unit volume of the fluid and P is the acoustic pressure. For an acoustic wave in a Newtonian fluid, the linearized equation of motion is

$$\rho_o \frac{\partial \mathbf{V}}{\partial t} = -grad\, P + \eta_a \nabla^2 \mathbf{V} \quad , \quad \eta_a = \eta_B + 4\eta/3, \tag{2.2b}$$

where η is the shear viscosity and η_B is the bulk viscosity of the fluid. Similarly, in this limit, the equation of continuity assumes the form

$$\frac{\partial P}{\partial t} = -B\, div \boldsymbol{V} \ , \tag{2.3}$$

where B is the adiabatic bulk modulus of the fluid.
Upon differentiating (2.3) with respect to time and using (2.2a), we obtain a (three-dimensional) wave equation for the acoustic pressure in a loss-less medium:

$$\nabla^2 P - \frac{1}{\upsilon^2}\frac{\partial^2 P}{\partial t^2} = 0 \quad , \quad \upsilon = \sqrt{\frac{B}{\rho_o}} \ . \tag{2.4}$$

2.3 Electromagnetic Wave

An electromagnetic wave is an example of a vector field. In this wave the disturbance transported is an electromagnetic field. Ordinarily, an electromagnetic field is represented using four field functions, which are vector quantities, namely: the electric field strength **E**(**r**,t), the magnetic field strength **H**(**r**,t), the electric displacement **D**(**r**,t), and the magnetic induction **B**(**r**,t). The relations which describe the behaviour of the electromagnetic field, and consequently the electromagnetic wave, are called Maxwell equations.
In a source-free linear medium the symmetric form of Maxwell's, first and second, equations can be written as [4]:

$$curl\, \boldsymbol{E} = -\mu \frac{\partial}{\partial t}\boldsymbol{H} - \sigma_m \boldsymbol{H} \tag{2.5}$$

$$curl\, \boldsymbol{H} = \varepsilon \frac{\partial}{\partial t}\boldsymbol{E} + \sigma \boldsymbol{E} \ , \tag{2.6}$$

where σ is the electric conductivity, σ_m is the equivalent magnetic conductivity, ε is the electric permittivity, and μ is the magnetic permeability of the medium. In the form written above, the constitutive relations for the medium ($\mathbf{D} = \varepsilon\mathbf{E}$, $\mathbf{B} = \mu\mathbf{H}$, $\mathbf{J} = \sigma\mathbf{E}$ and $\mathbf{J}_m = \sigma_m\mathbf{H}$) are assumed. A general, three-dimensional, wave equation can be derived using equations (2.5) and (2.6). However, to simplify the resulting expressions, we assume a non-conductive medium, in which case the conductivity terms are zero, and equations (2.5) and (2.6) reduce to

$$curl\, \boldsymbol{E} = -\mu \frac{\partial \boldsymbol{H}}{\partial t} \ ; \ curl\, \boldsymbol{H} = \varepsilon \frac{\partial \boldsymbol{E}}{\partial t} \ . \tag{2.7}$$

Taking the *curl* of the first equation in (2.7) and using the second to eliminate **H** leads to

$$curl\, curl\, \boldsymbol{E} = -\varepsilon\mu \frac{\partial^2 \boldsymbol{E}}{\partial t^2} \ .$$

But $curl\, curl\, \boldsymbol{E} = -\nabla^2 \boldsymbol{E} + grad\, div\, \boldsymbol{E}$, and *div* **E** = 0 in a source free medium. Thus, writing *curl curl* **E** in terms of the laplacian, we obtain a wave equation in terms of **E:**

$$\nabla^2 \boldsymbol{E} - \frac{1}{\upsilon^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0 \quad , \quad \upsilon = \frac{1}{\sqrt{\varepsilon\mu}} \ . \tag{2.8}$$

Note that since **E** is a vector, in three-dimensions, equation (2.8) represents three scalar equations.

2.4 Initial and Boundary Conditions

It is evident, physically, that the wave in a medium must be induced by some external agent. This agent may be either of the following (or a combination of these):
(a) A non-homogeneous initial condition.
(b) A non-homogeneous boundary condition.
(c) A driving (or impressed) force.
Thus the particular solution of a wave equation is determined by the initial condition and the boundary condition. In the

absence of an impressed force, if the initial and boundary conditions are homogeneous (i.e. are zero) then the solution to the wave equation is identically zero. While dealing with a wave equation, of the *hyperbolic* type (which is second order in both space and time) one may specify two conditions on the space axis and two conditions on the time axis. But, usually, two initial conditions and a boundary condition are prescribed. It is not necessary to prescribe two conditions, each, on both the space and time axes (because one family of *characteristics* connects points on these two axes). Furthermore, on a bounded domain the conditions cannot be prescribed arbitrarily on the entire domain boundary.

3.0 Traditional FDTD Method

Finite difference approximations in the time domain provide a numerical means to solve the partial differential equations which govern a wave interaction problem [7]. Usually called the finite difference time domain (FDTD) method, it is applicable to a *well-posed* initial boundary value problem, given the initial conditions and boundary conditions. Mathematically, a *well-posed* problem guarantees a unique solution. The boundary conditions include the interface conditions at the surface of any object(s) present in the problem domain and the conditions applicable at the outer limits of the domain.
The FDTD method is an attractive computational technique for the prediction of field behaviour in wave interaction problems in which the time parameter appears explicitly. An FDTD algorithm is a system of finite difference equations for updating (the components of) the field functions which describe a wave, and as such provides a time evolution of the wave, with the solution at any time depending on the field values at earlier times. The complete FDTD algorithm is a finite difference approximation of the wave equation, initial condition and boundary condition.
An FDTD algorithm approximates a continuous wave field in space and time by sampled data at some points in a finite space-time lattice. The algorithm is formulated by finite differencing a pair of first-order partial differential equations which represent the wave. An interesting feature of this method is its ability to easily handle inhomogeneities through a local specification of medium parameters.

3.1 Discretization

A basic step in an FDTD algorithm formulation process is to replace the partial differential equations which govern a problem with difference equations. For this purpose the space-time continuum of the problem is replaced with a space-time lattice, and the field functions are defined at certain points on the lattice. This procedure is known as discretization. The domain within which calculations are done for the required field functions is called the computational lattice. Scattering objects (where present) are embedded in this lattice and influence the computation process via the medium parameters.
To discretize the partial differential equations and problem geometry the computation lattice is partitioned into, generally unequal, subintervals. Usually, the time axis is partitioned into equal time increments Δt and we represent any instant of time t with $n\Delta t$ where n is a positive integer. The introduction of a computation lattice automatically implies the choice of a coordinate system. Assuming arbitrary coordinates, in three dimensional space, a space point (η, ξ, ζ) is represented as $(i\Delta\eta, j\Delta\xi, k\Delta\zeta)$ where i, j, k are integers and $\Delta\eta, \Delta\xi$ and $\Delta\zeta$ are the space increments along the η, ξ and ζ axes, respectively.
In rectangular coordinates, the common FDTD symbolism introduced by Yee denotes a space point by

$$(i, j, k) \equiv (i\Delta x, j\Delta y, k\Delta z)\,, \tag{3.1}$$

where Δx, Δy and Δz are space increments along the respective axes, and a function of space-time by

$$\psi^n(i, j, k) \equiv \psi(i\Delta x, j\Delta y, k\Delta z, n\Delta t)\,. \tag{3.2}$$

The finite difference equations are set up by approximating the space and time derivatives contained in the partial differential equation by finite difference formulae at each lattice point where the solution function is to be determined.

3.2 Yee Algorithm

The finite difference time domain method was first introduced by Yee to model time-dependent Maxwell equations. Consider the symmetric form of Maxwell equations in a source free linear medium (2.5), where the medium parameters (ε, μ, σ, σ_m) are assumed to be time-independent. In rectangular coordinates equation (2.5) is re-written as the system of scalar equations (3.3):

$$\begin{aligned}
\frac{\partial H_x}{\partial t} &= \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \sigma_m H_x\right)\\
\frac{\partial H_y}{\partial t} &= \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \sigma_m H_y\right)\\
\frac{\partial H_z}{\partial t} &= \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma_m H_Z\right)
\end{aligned} \tag{3.3a}$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x\right)$$
$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y\right) \qquad (3.3b)$$
$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right)$$

The set of six coupled partial differential equations (3.3) is the basis for the finite difference time domain approximation, in rectangular coordinates, of an electromagnetic wave.
The discretization scheme introduced by Yee for equation (3.3) consists in replacing a first derivative of a field function, say ψ(x), with a second-order accurate central difference approximation:

$$\frac{\partial \psi(x_o)}{\partial x} = \frac{\psi(x_o + 0.5\Delta) - \psi(x_o - 0.5\Delta)}{\Delta} + \vartheta(\Delta^2), \qquad (3.4)$$

where xo is the (space or time) point at which the derivative is evaluated and Δ is an increment in either space or time.

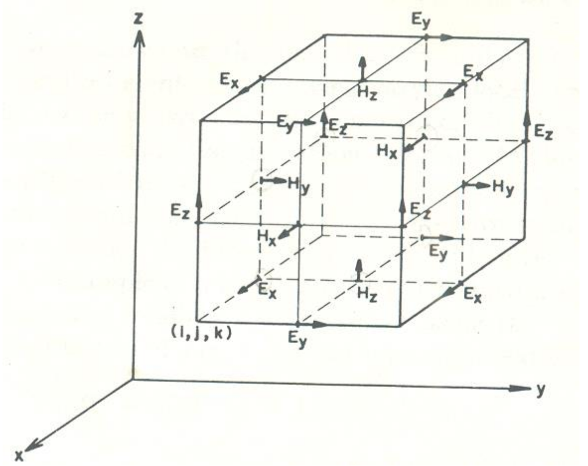

Figure 1: Positions of field components about a unit cell of the Yee lattice

To achieve the accuracy of equation (3.4) for a space derivative and to realize all the space derivatives in equation (3.3), the field components are staggered about a unit cell as shown in Figure 1.
The six rectangular field components of the electromagnetic wave are specified on each cell. The position of the field components in each cell is such as to enable us centre all space derivatives. The electric strength and magnetic strength vector components are interleaved in space, with each magnetic strength component surrounded by four circulating electric strength components, and vice versa. (This field components placement scheme provides a natural geometry for applying the integral forms of Ampere's law and Faraday's law at the unit cell level [8].) To achieve the accuracy of equation (3.4) for a time derivative, the electric and magnetic field components are evaluated at alternate half-time steps. This results in a space-time staggering of the field functions evaluation points on the (space-time) computation lattice.
Upon approximating the derivatives in equation (3.3) using central difference expressions of the form (3.4), and implementing equations (3.3a) and (3.3b) at alternate half-time increments, we obtain the set of finite difference equations (3.5).

$$H_x^{n+1/2}\left(i, j+\tfrac{1}{2}, k+\tfrac{1}{2}\right) = \left\{\frac{2\mu - \sigma_m \Delta t}{2\mu + \sigma_m \Delta t}\right\} H_x^{n-1/2}\left(i, j+\tfrac{1}{2}, k+\tfrac{1}{2}\right)$$
$$+ \frac{2\Delta t}{2\mu + \sigma_m \Delta t}\left\{\frac{1}{\Delta z}\left[E_y^n\left(i, j+\tfrac{1}{2}, k+1\right) - E_y^n\left(i, j+\tfrac{1}{2}, k\right)\right]\right.$$
$$\left. - \frac{1}{\Delta y}\left[E_z^n\left(i, j+1, k+\tfrac{1}{2}\right) - E_z^n\left(i, j, k+\tfrac{1}{2}\right)\right]\right\}$$

$$
\begin{aligned}
H_y^{n+1/2}\left(i+\tfrac{1}{2},j,k+\tfrac{1}{2}\right) &= \left\{\frac{2\mu-\sigma_m\Delta t}{2\mu+\sigma_m\Delta t}\right\} H_y^{n-1/2}\left(i+\tfrac{1}{2},j,k+\tfrac{1}{2}\right) \\
&+ \frac{2\Delta t}{2\mu+\sigma_m\Delta t}\left\{\frac{1}{\Delta x}\left[E_z^n\left(i+1,j,k+\tfrac{1}{2}\right)-E_z^n\left(i,j,k+\tfrac{1}{2}\right)\right]\right. \\
&\left. -\frac{1}{\Delta z}\left[E_x^n\left(i+\tfrac{1}{2},j,k+1\right)-E_x^n\left(i+\tfrac{1}{2},j,k\right)\right]\right\}
\end{aligned}
\tag{3.5}
$$

$$
\begin{aligned}
H_z^{n+1/2}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right) &= \left\{\frac{2\mu-\sigma_m\Delta t}{2\mu+\sigma_m\Delta t}\right\} H_y^{n-1/2}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right) \\
&+ \frac{2\Delta t}{2\mu+\sigma_m\Delta t}\left\{\frac{1}{\Delta y}\left[E_x^n\left(i+\tfrac{1}{2},j+1,k\right)-E_x^n\left(i+\tfrac{1}{2},j,k\right)\right]\right. \\
&\left. -\frac{1}{\Delta x}\left[E_y^n\left(i+1,j+\tfrac{1}{2},k\right)-E_x^n\left(i,j+\tfrac{1}{2},k\right)\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
E_x^{n+1}\left(i+\tfrac{1}{2}.j,k\right) &= \left\{\frac{2\varepsilon-\sigma\Delta t}{2\varepsilon+\sigma\Delta t}\right\} E_x^n\left(i+\tfrac{1}{2},j,k\right)+\frac{2\Delta t}{2\varepsilon+\sigma\Delta t}\left\{\frac{1}{\Delta y}\left[H_z^{n+1/2}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right)\right.\right. \\
&\left.\left. -H_z^{n+1/2}\left(i+\tfrac{1}{2},j-\tfrac{1}{2},k\right)\right]-\frac{1}{\Delta z}\left[H_y^{n+1/2}\left(i+\tfrac{1}{2},j,k+\tfrac{1}{2}\right)-H_y^{n+1/2}\left(i+\tfrac{1}{2},j,k-\tfrac{1}{2}\right)\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
E_y^{n+1}\left(i,j+\tfrac{1}{2},k\right) &= \left\{\frac{2\varepsilon-\sigma\Delta t}{2\varepsilon+\sigma\Delta t}\right\} E_y^n\left(i,j+\tfrac{1}{2},k\right)+\frac{2\Delta t}{2\varepsilon+\sigma\Delta t}\left\{\frac{1}{\Delta z}\left[H_x^{n+1/2}\left(i,j+\tfrac{1}{2},k+\tfrac{1}{2}\right)\right.\right. \\
&\left.\left. -H_x^{n+1/2}\left(i,j+\tfrac{1}{2},k-\tfrac{1}{2}\right)\right]-\frac{1}{\Delta x}\left[H_z^{n+1/2}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right)-H_z^{n+1/2}\left(i-\tfrac{1}{2},j+\tfrac{1}{2},k\right)\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
E_z^{n+1}\left(i,j,k+\tfrac{1}{2}\right) &= \left\{\frac{2\varepsilon-\sigma\Delta t}{2\varepsilon+\sigma\Delta t}\right\} E_z^n\left(i,j,k+\tfrac{1}{2}\right)+\frac{2\Delta t}{2\varepsilon+\sigma\Delta t}\left\{\frac{1}{\Delta x}\left[H_y^{n+1/2}\left(i+\tfrac{1}{2},j,k+\tfrac{1}{2}\right)\right.\right. \\
&\left.\left. -H_y^{n+1/2}\left(i-\tfrac{1}{2},j,k+\tfrac{1}{2}\right)\right]-\frac{1}{\Delta y}\left[H_x^{n+1/2}\left(i,j+\tfrac{1}{2},k+\tfrac{1}{2}\right)-H_x^{n+1/2}\left(i,j-\tfrac{1}{2},k+\tfrac{1}{2}\right)\right]\right\}
\end{aligned}
$$

The set of finite difference equations (3.5) apply to points within the interior of the computation lattice, only. These equations are explicit, that is, computing the value of a field component (at any time) does not require matrix inversion. We observe that the space and time increments may not be chosen arbitrarily.

3.3 Supplementary Parts of an FDTD Algorithm

To implement an FDTD algorithm it is necessary to append to the interior region finite difference equations two supplementary finite difference equations, namely:

Problem initial conditions. Recall that a wave equation is an initial boundary value problem. Hence in a computational scheme the initial condition has to be suitably introduced.

Lattice truncation condition. This is implemented at the outer boundary of the computation lattice, to permit a natural termination of the computation domain.

3.4 FDTD Algorithm Requirements

3.4.1 Consistency

It is important that the finite difference equation approximates the differential equation in some manner, for the finite difference model to be consistent with the differential equation system. Thus, in the limit of small increments, the finite difference equation should reduce to the system of differential equation. Assuming one-space dimension, the consistency requirement may be expressed mathematically as:

$$
\lim_{\substack{\Delta t\to 0\\ \Delta t/\Delta x\to\beta}}\ \lim_{\Delta x\to 0}\frac{\Gamma(\Delta t,\Delta x)}{\beta}=L
\tag{3.6}
$$

where L is the differential operator, Γ is the finite difference time-stepping operator, and β is finite.

3.4.2 Stability

If an FDTD algorithm can produce a solution which grows continuously with time-stepping, then the algorithm is numerically unstable. Basically, the requirement of stability demands that a small error at any stage in the computation produce a smaller cumulative error later. A useful FDTD algorithm must be a stable numerical method.

Suppose that for a single independent ordinary differential equation the error which occurs at time step n is εn, and that at time step n+1 is εn+1 such that

$$\varepsilon^{n+1} = q\varepsilon^{n} \tag{3.7}$$

where q is the growth factor, related to the time-stepping scheme which is employed in the finite difference model. Now, for stability of the algorithm, we require that

$$\left|\varepsilon^{n+1}\right| \le \left|\varepsilon^{n}\right| \quad \text{or} \quad \left|q\varepsilon^{n}\right| \le \left|\varepsilon^{n}\right|. \tag{3.8}$$

Hence numerical stability requires that

$$\left|q\right| \le 1, \tag{3.9}$$

if we assume that the class of problem does not permit a growing solution.

3.4.3 Convergence

A finite difference approximation to a differential equation is useful only if it is convergent and stable. The problem of convergence consists in finding the conditions under which the difference between the analytic solution of the differential equation and the numerical solution from the difference equation at a fixed space-time point, tends to zero uniformly, as the computation lattice is refined in such a way that the space and time increments tend to zero and the number of subintervals tends to infinity. According to Lax and Richtmyer, if a linear difference equation is consistent with a properly posed linear initial value problem then stability is a necessary and sufficient condition for convergence. The relationship between stability and convergence is called Lax equivalence theorem [9]: It states that for consistent finite difference approximations of time-dependent linear partial differential equations which is well posed, the numerical scheme converges if it is stable and it is stable if it converges.

3.4.4 Accuracy

The accuracy of the solution of an FDTD algorithm is impaired by the occurrence of two sources of error. The first is the truncation error, caused by the approximation involved in replacing a differential equation with a difference equation. This error is dependent on the magnitude of the increments in space and time, and on the computation lattice density. In the choice of an FDTD algorithm a basic requirement is the minimization of the truncation error. The other source of error is the round-off error, due to the accuracy with which a particular number is represented in a digital computer (or the finite word-length of the computer). The process of obtaining a numerical result is one which involves finite-precision arithmetic. Often, the accuracy improves with increasing complexity of the algorithm used to approximate a particular differential equation.

3.4.5 Efficiency

A useful FDTD algorithm should be cost effective both in implementation time and memory requirement, since all computers have a finite capacity. The efficiency of an algorithm may be defined as the total number of arithmetic, logical and storage operations performed by a computer to obtain a solution over a unit time-length of the problem. Since efficiency decreases with algorithm complexity, a compromise is needed between efficiency and accuracy.

4.0 Acoustic Wave FDTD Algorithm

4.1 Acoustic Wave Field Equations

Since the FDTD method is an adaptation of the leap-frog scheme for the numerical solution of a hyperbolic partial differential equation, one may apply the FDTD method to any wave type. For a disturbance of small amplitude, in a lossless medium, an acoustic wave is governed by the equations (2.2) and (2.3). Assuming rectangular coordinates, these two coupled first-order equations are equivalent to the following system of equations:

$$\frac{\partial P}{\partial t} = -B\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right)$$

$$\frac{\partial V_x}{\partial t} = -\frac{1}{\rho_o}\frac{\partial P}{\partial x}$$
$$\frac{\partial V_y}{\partial t} = -\frac{1}{\rho_o}\frac{\partial P}{\partial y}$$
$$\frac{\partial V_z}{\partial t} = -\frac{1}{\rho_o}\frac{\partial P}{\partial z} \qquad (4.1)$$

The set of four coupled partial differential equations (4.1) is the basis of an FDTD acoustic wave algorithm [10], in rectangular coordinates. The derivatives in (4.1) are approximated using central difference expressions of the form (3.4) at various space-time points on a computation lattice.

4.2 Acoustic Wave Algorithm

To discretize equation (4.1) by the leap-frog method we use the field placement scheme of Figure 2 [11]. The placement of the field functions is such that the acoustic pressure is at the body-centre of a unit cell while the velocity vector components are each at the face-centre of the cell, such that Vx is normal to the x-plane, Vy is normal to the y-plane and Vz is normal to the z-plane. Observe that the inter-leaving of the points at which the pressure and velocity are evaluated on a computation lattice implies that field data for pressure are available at integer (i) space points while those for the velocity are available at half-integer (i+0.5) space points.
The field placement scheme of Figure 2 is different from that introduced by Yee (Figure 1). This is due to the longitudinal nature of acoustic waves. The use of a Yee cell, in this situation, will lead to inappropriate conclusions (when the time-stepping relations resulting from equation (4.1) are giving the contour-integral interpretation of Taflove and co-workers [8]). This is a fundamental difference which is apparent on applying finite difference time domain to acoustic wave modelling.

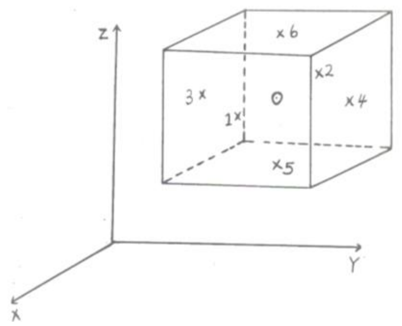

Figure 2: FDTD acoustic wave unit cell. Body-centre point O ≡ i,j,k for P-field.

Face-centre points 1 to 6 for V-fields: 1,2 for Vx; 3,4 for Vy; 5,6 for Vz.
Approximating the time and space derivatives in equation (4.1) using equation (3.4), at alternate-half time steps, we have the finite difference equations:

$$\begin{aligned} P^{n+1}(i,j,k) = P^{n}(i,j,k) &- \frac{B\Delta t}{\Delta x}\left\{V_x^{n+1/2}\left(i+\tfrac{1}{2},j,k\right) - V_x^{n+1/2}\left(i-\tfrac{1}{2},j,k\right)\right\} \\ &- \frac{B\Delta t}{\Delta y}\left\{V_y^{n+1/2}\left(i,j+\tfrac{1}{2},k\right) - V_y^{n+1/2}\left(i,j-\tfrac{1}{2},k\right)\right\} \\ &- \frac{B\Delta t}{\Delta z}\left\{V_z^{n+1/2}\left(i,j,k+\tfrac{1}{2}\right) - V_z^{n+1/2}\left(i,j,k-\tfrac{1}{2}\right)\right\} \end{aligned} \qquad (4.2a)$$

$$V_x^{n+1/2}\left(i+\tfrac{1}{2},j,k\right) = V_x^{n-1/2}\left(i+\tfrac{1}{2},j,k\right) - \frac{\Delta t}{\rho_o \Delta x}\left\{P^n(i+1,j,k) - P^n(i,j,k)\right\}$$
$$V_y^{n+1/2}\left(i,j+\tfrac{1}{2},k\right) = V_y^{n-1/2}\left(i,j+\tfrac{1}{2},k\right) - \frac{\Delta t}{\rho_o \Delta y}\left\{P^n(i,j+1,k) - P^n(i,j,k)\right\}$$
$$V_z^{n+1/2}\left(i,j,k+\tfrac{1}{2}\right) = V_z^{n-1/2}\left(i,j,k+\tfrac{1}{2}\right) - \frac{\Delta t}{\rho_o \Delta z}\left\{P^n(i,j,k+1) - P^n(i,j,k)\right\} \qquad (4.2b)$$

As before, we need the initial conditions at t = 0 together with relations for evaluating the field functions at the computation lattice boundary to implement the set of FDTD time stepping relations (4.2).

5.0 ADI-FDTD Method

5.1 Basic ADI Assumptions

The traditional FDTD method is based on a set of explicit finite difference equations. Consequently, the allowed time increment is bound by the Courant-Friedrichs-Lewy (CLF) stability condition [12]. The CLF condition limits the capability of the traditional FDTD method because, if an object has a small size compared with wavelength, a small time increment introduces a significant increase in the computation time required to obtain an acceptable/accurate solution. To overcome this drawback, the alternating direction implicit - finite difference time domain (ADI-FDTD) method [13, 14] has been introduced to model certain wave problems.

In an ADI-FDTD method the set of finite difference equations for updating the field components from the nth to the (n+1)th time step is broken into a number of sub-sets, and the set of equations is implicit, that is, the solution requires a matrix inversion. These sub-sets represent alternations in the computation directions. The alternations in the computation directions may be made in respect of either:

the spatial coordinate directions, or

the sequence of terms on the right-hand-side of the partial differential equations.

Thus the computation, to advance one time step, is performed using a number of updating sub-procedures equal to the number of sub-sets which constitute the algorithm.

In the ADI-FDTD method all the field components are defined at every time, say n, (n+0.5) and (n+1), contrary to the traditional FDTD method which defines some field components at say (n+0.5) and others at (n+1). Also, a space derivative is replaced with a central difference approximation which is second-order accurate in the increment:

$$\frac{\partial\psi(x_o)}{\partial x}=\frac{\psi(x_o+0.5\Delta x)-\psi(x_o-0.5\Delta x)}{\Delta x}+\vartheta(\Delta x^2) \quad (5.1a)$$

$$\frac{\partial^2\psi(x_o)}{\partial x^2}=\frac{\psi(x_o+\Delta x)-2\psi(x_o)+\psi(x_o-\Delta x)}{\Delta x^2}+\vartheta(\Delta x^2) \quad (5.1b)$$

where xo is the expansion point and Δx is a space increment. However, a time derivative is replaced with a forward difference approximation which is first-order accurate in the increment:

$$\frac{\partial\psi^n}{\partial t}=\frac{\psi^{n+0.5}-\psi^n}{0.5\Delta t}+\vartheta(\Delta t) \quad (5.2)$$

We present the ADI-FDTD method for modelling acoustic wave phenomena in a lossy medium. For pedagogic reasons, the formulation of the ADI-FDTD acoustic wave algorithm begins with a one-dimensional situation.

5.2 ADI-FDTD 1-D Algorithm

To introduce the formulation, let us consider a 1-D situation for which the governing partial differential equations (2.2b) and (2.3) reduce to

$$\rho_o\frac{\partial V_x}{\partial t}=-\frac{\partial P}{\partial x}+\eta_a\frac{\partial^2 V_x}{\partial x^2} \quad (5.3a)$$

$$\frac{\partial P}{\partial t}=-B\frac{\partial V_x}{\partial x}. \quad (5.3b)$$

According to the alternating direction implicit (ADI) principle, the finite difference algorithm for marching from the time instant n to the time instant (n+1) is broken up into a number of sub-steps; namely the half time step from n to (n+0.5), and the half time step from (n+0.5) to (n+1). Considering equation (5.3a) for instance, the two half time steps are:

1) For the half time step from n to (n+0.5), at the time instant (n+0.5), Vx is updated using an explicit finite difference equation arising from

$$\rho_o\left.\frac{\partial V_x}{\partial t}\right|_{i+0.5}^{n}=-\left.\frac{\partial P}{\partial x}\right|_{i+0.5}^{n}+\eta_a\left.\frac{\partial^2 V_x}{\partial x^2}\right|_{i+0.5}^{n} \quad (5.4)$$

2) For the half time step from (n+0.5) to (n+1), at the time instant (n+1), Vx is updated using an implicit finite difference equation arising from

$$\rho_o\left.\frac{\partial V_x}{\partial t}\right|_{i+0.5}^{n+0.5}=-\left.\frac{\partial P}{\partial x}\right|_{i+0.5}^{n+1}+\eta_a\left.\frac{\partial^2 V_x}{\partial x^2}\right|_{i+0.5}^{n+1} \quad (5.5)$$

Thus for the system of equations (5.3) the two sub-procedures which constitute the ADI-FDTD algorithm are:

First sub-procedure (n → n+0.5).

$$V_x^{n+0.5}(i+0.5) = V_x^n(i+0.5) - 2d_x V_x^n(i+0.5) - a_x\{P^n(i+1) - P^n(i)\} + \\ + d_x\{V_x^n(i+1.5) + V_x^n(i-0.5)\} \tag{5.6a}$$

$$P^{n+0.5}(i) = P^n(i) - b_x\{V_x^n(i+0.5) - V_x^n(i-0.5)\} \tag{5.6b}$$

where

$$a_x = \frac{\Delta t}{2\rho_o \Delta x} \quad , \quad b_x = \frac{B\Delta t}{2\Delta x} \quad , \quad d_x = \frac{\eta_a \Delta t}{2\rho_o \Delta x^2} \,. \tag{5.6c}$$

Second sub-procedure (n+0.5 → n+1)

$$V_x^{n+1}(i+0.5) = V_x^{n+0.5}(i+0.5) - 2d_x V_x^{n+1}(i+0.5) - \\ - a_x\{P^{n+1}(i+1) - P^{n+1}(i)\} + d_x\{V_x^{n+1}(i+1.5) + V_x^{n+1}(i-0.5)\} \tag{5.7a}$$

$$P^{n+1}(i) = P^{n+0.5}(i) - b_x\{V_x^{n+1}(i+0.5) - V_x^{n+1}(i-0.5)\} \tag{5.7b}$$

Observe that the ADI-FDTD discretization method applied to a 1-D situation results in one explicit sub-procedure and one implicit sub-procedure, leading to a split implicit algorithm. Note that in a geometrical sense, the notion of 'alternating directions' is not appropriate to one-dimension as there is only a single direction.
The implicit expression (5.7a) cannot be used directly because of the presence of Pn+1. Thus equation (5.7b) is used to eliminate Pn+1 in equation (5.7a) leading to:

$$-t_x V_x^{n+1}(i+1.5) + [1+2t_x]V_x^{n+1}(i+0.5) - t_x V_x^{n+1}(i-0.5) = V_x^{n+0.5}(i+0.5) \\ - a_x\{P^{n+0.5}(i+1) - P^{n+0.5}(i)\} \quad , \quad t_x = a_x b_x + d_x \tag{5.7c}$$

The implicit equation (5.7c) is usually resolved using a tri-diagonal matrix algorithm, after which equation (5.7b) is evaluated.

5.3 ADI-FDTD 2-D Algorithm

In a two-dimensional situation, assuming the problem is independent of z, the governing partial differential equations reduce to

$$\rho_o \frac{\partial V_x}{\partial t} = -\frac{\partial P}{\partial x} + \eta_a \frac{\partial^2 V_x}{\partial x^2} \tag{5.8a}$$

$$\rho_o \frac{\partial V_y}{\partial t} = -\frac{\partial P}{\partial y} + \eta_a \frac{\partial^2 V_y}{\partial y^2} \tag{5.8b}$$

$$\frac{\partial P}{\partial t} = -B\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y}\right) \tag{5.8c}$$

Applying the same approach that we used for the discretization of the one-dimensional case leads to the ADI-FDTD 2-D acoustic wave algorithm. The first sub-procedure is given by equations (5.9) and the second sub-procedure by equations (5.10).

First sub-procedure (n → n+0.5)

$$V_x^{n+0.5}(i+0.5,j) = V_x^n(i+0.5,j) - 2d_x V_x^n(i+0.5,j) - a_x\{P^n(i+1,j) - P^n(i,j)\} + \\ + d_x\{V_x^n(i+1.5,j) + V_x^n(i-0.5,j)\} \tag{5.9a}$$

$$V_y^{n+0.5}(i,j+0.5) = V_y^n(i,j+0.5) - 2d_y V_y^{n+0.5}(i,j+0.5) - a_y\{P^{n+0.5}(i,j+1) - \\ - P^{n+0.5}(i,j)\} + d_y\{V_y^{n+0.5}(i,j+1.5) + V_y^{n+0.5}(i,j-0.5)\} \tag{5.9b}$$

$$P^{n+0.5}(i,j) = P^n(i,j) - b_x\{V_x^n(i+0.5,j) - V_x^n(i-0.5,j)\} - \\ - b_y\{V_y^{n+0.5}(i,j+0.5) - V_y^{n+0.5}(i,j-0.5)\} \tag{5.9c}$$

where, $a_y = \dfrac{\Delta t}{2\rho_o \Delta y} \quad , \quad b_y = \dfrac{B\Delta t}{2\Delta y} \quad , \quad d_y = \dfrac{\eta_a \Delta t}{2\rho_o \Delta y^2}$.

Second sub-procedure (n+0.5→ n+1)

$$V_x^{n+1}(i+0.5,j)=V_x^{n+0.5}(i+0.5,j)-2d_xV_x^{n+1}(i+0.5,j)-a_x\{P^{n+1}(i+1,j)-$$
$$-P^{n+1}(i,j)\}+d_x\{V_x^{n+1}(i+1.5,j)+V_x^{n+1}(i-0.5,j)\} \quad (5.10a)$$

$$V_y^{n+1}(i,j+0.5)=V_y^{n+0.5}(i,j+0.5)-2d_yV_y^{n+0.5}(i,j+0.5)-a_y\{P^{n+0.5}(i,j+1)-$$
$$-P^{n+0.5}(i,j)\}+d_y\{V_y^{n+0.5}(i,j+1.5)+V_y^{n+0.5}(i,j-0.5)\} \quad (5.10b)$$

$$P^{n+1}(i,j)=P^{n+0.5}(i,j)-b_x\{V_x^{n+1}(i+0.5,j)-V_x^{n+1}(i-0.5,j)\}-$$
$$-b_y\{V_y^{n+0.5}(i,j+0.5)-V_y^{n+0.5}(i,j-0.5)\} \quad (5.10c)$$

Observe that equation (5.9b) in the first sub-procedure and equation (5.10a) in the second sub-procedure can not be solved directly. Hence they are re-expressed using the equations (5.9c) for Pn+0.5 and (5.10c) for Pn+1 in the respective sub-procedures, thus:

$$-t_yV_y^{n+0.5}(i,j+1.5)+[1+2t_y]V_y^{n+0.5}(i,j+0.5)-t_yV_y^{n+0.5}(i,j-0.5)=$$
$$=V_y^{n}(i,j+0.5)-a_y\{P^{n}(i,j+1)-P^{n}(i,j)\}+c_{xy}\{V_x^{n}(i+0.5,j+1)-$$
$$-V_x^{n}(i-0.5,j+1)-V_x^{n}(i+0.5,j)+V_x^{n}(i-0.5,j)\} \quad (5.9d)$$

$$-t_xV_x^{n+1}(i+1.5,j)+[1+2t_x]V_x^{n+1}(i+0.5,j)-t_xV_x^{n+1}(i-0.5,j)=$$
$$=V_x^{n+0.5}(i+0.5,j)-a_x\{P^{n+0.5}(i+1,j)-P^{n+0.5}(i,j)\}+c_{xy}\{V_y^{n+0.5}(i+1,j+0.5)-$$
$$-V_y^{n+0.5}(i+1,j-0.5)-V_y^{n+0.5}(i,j+0.5)+V_y^{n+0.5}(i,j-0.5)\} \quad (5.10d)$$

where $c_{xy}=B\Delta t^2/4\rho_o\Delta x\Delta y \quad and \quad t_y=a_yb_y+d_y$.

The equations (5.9d) and (5.10d) are solved using a tri-diagonal matrix algorithm. We can similarly derive the ADI-FDTD three- dimensional algorithm

6.0 Numerical Stability

6.1 Von Neumann Method

An FDTD algorithm may be analyzed for numerical stability using the von Neumann method [2, 9]. In the von Neumann method, instantaneous values of the field functions distributed across the computation lattice are first Fourier transformed into waves in spatial spectral domain, to represent a spectrum of spatial sinusoidal modes. Using the method of separation of variables, assume the function ψn(i, j, k) has a product solution of the form

$$\psi^n(i,j,k)=\psi_oq^{t/\Delta t}\,exp\{J(\alpha_xi\Delta x+\alpha_yj\Delta y+\alpha_zk\Delta z)\} \quad , \quad J=\sqrt{-1}$$
$$=\psi_oq^n\,exp\{J(\alpha_xi\Delta x+\alpha_yj\Delta y+\alpha_zk\Delta z)\} \quad (6.1)$$

where αx, αy, αz are components of the wavenumber, and the substitution t = nΔt has been made. In equation (6.1) it is evident that:

i) if q > 1 the solution grows exponentially in time;
ii) if 0 < q < 1 the solution is an exponential decay in time;
iii) if q = 1 the solution is constant in time;
iv) if -1 < q < 0 the solution is a convergent oscillation in time;
v) if q = -1 the solution is a pure oscillation in time;
vi) if q < -1 the solution is a divergent oscillation in time.

Thus the value of q determines the nature of the solution and the stability of the algorithm. If $|q|\le 1$ for all modes then the numerical algorithm is stable; otherwise, the algorithm is unstable.

6.2 Courant Stability Criterion

Note that each field component for an acoustic wave or electromagnetic wave satisfies equation (2.1). Also, assuming rectangular coordinates, the scalar wave equation (2.1) can be written as

$$\frac{1}{\upsilon^2}\frac{\partial^2\psi}{\partial t^2}=\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}+\frac{\partial^2\psi}{\partial z^2}. \qquad (6.2)$$

Substituting equation (1.14), which is the central difference approximation of a second derivative, into equation (6.2) leads to the finite difference approximation

$$\frac{1}{(\upsilon\Delta t)^2}\left\{\psi^{n+1}(i,j,k)-2\psi^n(i,j,k)+\psi^{n-1}(i,j,k)\right\}=\frac{1}{(\Delta x)^2}\left\{\psi^n(i+1,j,k)\right.$$
$$\left.-2\psi^n(i,j,k)+\psi^n(i-1,j,k)\right\}+\frac{1}{(\Delta y)^2}\left\{\psi^n(i,j+1,k)-2\psi^n(i,j,k)\right.$$
$$\left.+\psi^n(i,j-1,k)\right\}+\frac{1}{(\Delta z)^2}\left\{\psi^n(i,j,k+1)-2\psi^n(i,j,k)+\psi^n(i,j,k-1)\right\}. \qquad (6.3)$$

Equation (6.3) is equivalent to equation (3.5) or (4.2), and we shall analyze (6.3) for numerical stability instead of the FDTD algorithms in Sections 3 and 4. Following the von Neumann method, we substitute equation (6.1) into equation (6.3) and divide through by ψn(i, j, k) to obtain

$$q-2+\frac{1}{q}=\sigma \qquad (6.4a)$$

where

$$\sigma=\left(\frac{\upsilon\Delta t}{\Delta x}\right)^2\left\{\exp(J\alpha_x\Delta x)-2+\exp(-J\alpha_x\Delta x)\right\}$$
$$+\left(\frac{\upsilon\Delta t}{\Delta y}\right)^2\left\{\exp(J\alpha_y\Delta y)-2+\exp(-J\alpha_y\Delta y)\right\}$$
$$+\left(\frac{\upsilon\Delta t}{\Delta z}\right)^2\left\{\exp(J\alpha_z\Delta z)-2+\exp(-J\alpha_z\Delta z)\right\}$$

Note that $J=\sqrt{-1}$ in the above expression. In the alternative, we write

$$\sigma = 2\left(\frac{\upsilon\Delta t}{\Delta x}\right)^2\left\{cos(\alpha_x\Delta x)-1\right\}+2\left(\frac{\upsilon\Delta t}{\Delta y}\right)^2\left\{cos(\alpha_y\Delta y)-1\right\}$$
$$+2\left(\frac{\upsilon\Delta t}{\Delta z}\right)^2\left\{cos(\alpha_z\Delta z)-1\right\}. \qquad (6.4b)$$

Thus the growth factor q is obtained from the quadratic equation (6.4a) which is re-written as

$$q^2-(\sigma+2)q+1=0. \qquad (6.5a)$$

There are two possible values of q, namely,

$$q=\frac{\sigma+2}{2}\pm\sqrt{\left(\frac{\sigma+2}{2}\right)^2-1}. \qquad (6.5b)$$

The two values or roots (q_1, q_2) correspond to two ways in which the numerical solution evolves in time.

If -2 < σ + 2 < 2, the roots are complex conjugates of each other, $q_1=q_2^*$:

$$q_1=\frac{\sigma+2}{2}+J\sqrt{1-\left(\frac{\sigma+2}{2}\right)^2}$$

$$q_2=\frac{\sigma+2}{2}-J\sqrt{1-\left(\frac{\sigma+2}{2}\right)^2}$$

and

$$|q_1|^2=q_1q_2^*=1.$$

Thus for these values of q we have a numerical solution which oscillates with time for a fixed spatial position. This is analogous to the wave equation itself, which permits a time periodic solution when the spatial part is periodic.

If $|\sigma+2|>2,$ the roots are real numbers and their product is unity. For instance with σ+2 > 2,

$$q_1 = \frac{\sigma+2}{2} + \sqrt{\left(\frac{\sigma+2}{2}\right)^2 - 1} \ ,$$

$$q_2 = \frac{\sigma+2}{2} - \sqrt{\left(\frac{\sigma+2}{2}\right)^2 - 1}$$

and $q_1 q_2 = 1$. Thus one of the roots will be greater than unity in absolute value leading to the growth of the numerical solution with time. Such behaviour of the algorithm is called numerical instability. Numerical instability often manifests as a divergent oscillation in time (q < -1) for an oscillatory solution in space. The numerical solution will be stable if -2 < σ + 2 < 2 or -4 < σ < 0. But, -2 ≤ cosθ -1 ≤ 0 for all possible values of θ. Therefore, from equation (6.4b)

$$-4(\upsilon\Delta t)^2 \left\{ \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right\} \le \sigma \le 0. \tag{6.6}$$

But since, for stability of the algorithm, σ cannot be less than -4, we have that

$$\left\{ \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right\} \le \frac{1}{(\upsilon\Delta t)^2} . \tag{6.7}$$

Equation (6.7) is known as the Courant-Friedrichs-Lewy stability criterion (or simply the Courant stability criterion) for finite difference algorithm of the wave equations derived via the leap-frog method. It requires the space and time increments in the FDTD algorithm to be chosen such that the inequality (6.7) is always satisfied, where υ is the maximum phase velocity occurring in the modelled problem. Violating (6.7) leads to numerical instability.

In FDTD models involving only one space dimension equation (6.10) reduces to

$$\frac{\upsilon}{\Delta x / \Delta t} \le 1. \tag{6.8}$$

In a manner analogous to the notion of characteristics, which can be thought of as bounding the region of space-time that is causally influenced by an incident at a given space-time point, Haberman [9] interprets equation (6.8) to mean that for stability the space and time increments must be such that the numerical wave moves at a faster speed than the physical wave. In this way the FDTD algorithm will be able to account for the physical phenomenon. Thus the ratio of space increment to time increment is interpreted as the speed with which the numerical wave travels.

For models limited to two space dimensions equation (6.7) reduces to

$$\upsilon\Delta t \le \frac{\Delta x \Delta y}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} . \tag{6.9}$$

Fusco [15] gives a geometrical interpretation of this stability condition according to which equation (6.9) implies that the normalized time increment, $\upsilon\Delta t$, should not exceed the ratio of area to diagonal distance in any cell of the computation lattice (in rectangular coordinates).

6.3 ADI-FDTD Stability Criterion

The ADI-FDTD algorithm of Section 5 is analyzed for numerical stability using von Neumann method. Assume, in 1-D, a harmonic representation of a field function

$$\psi^n(i) = \psi_o q^n \, exp \, J(\alpha_x i \Delta x) , \tag{6.10}$$

where αx is the wave number.

For the first sub-procedure equation (5.6), we substitute the harmonic representation, equation (6.10), of the pressure and velocity fields into equation (5.6) to obtain

$$\left(q_1^{0.5} - 1 + 2d_x - 2d_x \cos\theta_x\right)V_x^n + J2a_x \sin(\theta_x/2)P^n = 0 \tag{6.11a}$$

$$J2b_x \sin(\theta_x/2)V_x^n + \left(q_1^{0.5} - 1\right)P^n = 0 \quad , \quad \theta_x = \alpha_x \Delta x , \tag{6.11b}$$

where q1 is the growth factor for the first sub-procedure. Eliminating any one of the field functions from the pair (6.11) leads to a quadratic equation for the growth factor

$$\varepsilon_1^2 - (2 - \iota_x)\varepsilon_1 + (\iota_x - \beta_x) = 0 \quad , \quad q_1 \equiv \varepsilon_1^2 , \tag{6.12a}$$

where

$$\beta_x = 1 + 4a_x b_x \sin^2(\theta_x/2) \quad , \quad \iota_x = 2d_x(1 - \cos\theta_x) = 4d_x \sin^2(\theta_x/2) . \tag{6.12b}$$

Thus

$$\varepsilon_1 = \frac{(2-\iota_x) \pm J\sqrt{4(\beta_x - \iota_x) - (2-\iota_x)^2}}{2} \quad , \quad |\varepsilon_1| = \sqrt{\beta_x - \iota_x} \quad \text{or} \tag{6.13a}$$

$$q_1 = |\varepsilon_1||\varepsilon_1| = (\beta_x - \iota_x). \tag{6.13b}$$

Similarly, for the second sub-procedure (5.7) substituting the harmonic representation (6.10) leads to:

$$\{q_2^{0.5}(1 + 2d_x - 2d_x \cos\theta_x) - 1\}V_x^n + Jq_2^{0.5} 2a_x \sin(\theta_x/2)P^n = 0 \tag{6.14a}$$

$$Jq_2^{0.5} 2b_x \sin(\theta_x/2)V_x^n + (q_2^{0.5} - 1)P^n = 0 \tag{6.14b}$$

Eliminating anyone of the field functions from (6.14) gives the quadratic equation

$$(\beta_x + \iota_x)\varepsilon_2^2 - (2 + \iota_x)\varepsilon_2 + 1 = 0 \quad , \quad q_2 \equiv \varepsilon_2^2. \tag{6.15}$$

Thus

$$\varepsilon_2 = \frac{(2+\iota_x) \pm J\sqrt{4(\beta_x + \iota_x) - (2+\iota_x)}}{2(\beta_x + \iota_x)} \quad , \quad |\varepsilon_2| = \frac{1}{\sqrt{\beta_x + \iota_x}}, \tag{6.16a}$$

or

$$q_2 = |\varepsilon_2||\varepsilon_2| = (\beta_x + \iota_x)^{-1}. \tag{6.16b}$$

Since the growth factor for the entire ADI-FDTD algorithm is the product of the growth factors of the sub-procedures, we have that the total growth factor for the 1-D ADI-FDTD algorithm is

$$q = q_1 q_2 = (\beta_x - \iota_x)/(\beta_x + \iota_x) \le 1. \tag{6.17}$$

Given the values of βx and ιx in (6.12b), equation (6.17) is always satisfied and the ADI-FDTD algorithm is unconditionally stable. Observe from the definitions in (6.12b) that ιx is a loss-term arising from the medium viscosity and (6.17) implies that the wave decays with continued time-stepping in a viscous medium. In a non-viscous medium ιx = 0 (since dx = 0 when ηa =0) and the growth factor for the algorithm reduces to unity, signifying that the algorithm is always stable.

The numerical stability analysis for the 2-D ADI-FDTD acoustic wave algorithm is carried out as before, assuming a harmonic wave representation of the field functions

$$\psi^n(i,j) = \psi_o q^n \exp J(\alpha_x i\Delta x + \alpha_y j\Delta y). \tag{6.18}$$

For the first sub-procedure (5.9) we substitute the wave representation (6.18) for the pressure and velocity field functions in (5.9) to obtain:

$$(q_1^{0.5} - 1 + \iota_x)V_x^n + J2a_x \sin(\theta_x/2)P^n = 0 \tag{6.19a}$$

$$[q_1^{0.5}(1 + \iota_y) - 1]V_y^n + Jq_1^{0.5} 2a_y \sin(\theta_y/2)P^n = 0 \tag{6.19b}$$

$$J2b_x \sin(\theta_x/2)V_x^n + Jq_1^{0.5} 2b_y \sin(\theta_y/2)V_y^n + (q_1^{0.5} - 1)P^n = 0 \tag{6.19c}$$

where $\iota_y = 4d_y \sin^2(\theta_y/2)$ *and* $\theta_y = \alpha_y \Delta y$.

We write the set of simultaneous equations in matrix form, calculate the determinant of the system-matrix and equate it to zero, to obtain a cubic equation for the growth factor q1:

$$\begin{aligned} q_1^{1.5}&[1 + \iota_y + 4a_y b_y \sin^2(\theta_y/2)] + q_1[\iota_x\iota_y + \iota_x + 4\iota_x a_y b_y \sin^2(\theta_y/2) - \\ &- 4a_y b_y \sin^2(\theta_y/2) - 2\iota_y - 3] + q_1^{0.5}[3 - 2\iota_x + 4a_x b_x \sin^2(\theta_x/2) + \\ &+ 4\iota_y a_x b_x \sin^2(\theta_x/2) + \iota_y - \iota_x\iota_y] + [\iota_x - 1 - 4a_x b_x \sin^2(\theta_x/2)] = 0. \end{aligned} \tag{6.20}$$

For any cubic equation, if we extract the unity solution, the equation can be written as a product of a linear equation and a quadratic equation thus:

$$(x-1)(ax^2 + bx + c) = ax^3 + (b-a)x^2 + (c-b)x - c = 0. \tag{6.21}$$

Hence, given a, c, (b-a) and (c-a) we have that

$$b = ½\{c + a - (c - b) + (b - a)\}. \tag{6.22}$$

Now, for the cubic equation (6.20), upon extracting the unity solution we have the quadratic equation for the growth factor:

$$(\beta_y + \iota_y)\varepsilon_1^2 - \Gamma_1\varepsilon_1 + (\beta_x - \iota_x) = 0 \quad , \quad q_1 \equiv \varepsilon_1^2, \tag{6.23a}$$

where

$$\beta_y = 1 + 4a_y b_y \sin^2(\theta_y/2),$$

$$\Gamma_1 = \iota_x\iota_y + \iota_x - \iota_y - 2 - 2\iota_y a_x b_x \sin^2(\theta_x/2) + 2\iota_x a_y b_y \sin^2(\theta_y/2) \, . \qquad (6.23b)$$

Solving equation (6.23a) gives the growth factor of the first sub-procedure

$$\varepsilon_1 = \frac{\Gamma_1 \pm J\sqrt{4(\beta_y + \iota_y)(\beta_x - \iota_x) - \Gamma_1^2}}{2(\beta_y + \iota_y)}, \quad |\varepsilon_1| = \sqrt{\frac{\beta_x - \iota_x}{\beta_y + \iota_y}} \qquad (6.24a)$$

or

$$q_1 = \frac{\beta_x - \iota_x}{\beta_y + \iota_y} \, . \qquad (6.24b)$$

For the second sub-procedure (5.10), following the same method, the quadratic equation for the growth factor is

$$(\beta_x + \iota_x)\varepsilon_2^2 - \Gamma_2\varepsilon_2 + (\beta_y - \iota_y) = 0 \quad , \quad q_2 \equiv \varepsilon_2^2 \, , \qquad (6.25a)$$

where

$$\Gamma_2 = \iota_y\iota_x + \iota_y - \iota_x - 2 - 2\iota_x a_y b_y \sin^2(\theta_y/2) + 2\iota_y a_x b_x \sin^2(\theta_x/2) \, . \qquad (6.25b)$$

Thus

$$\varepsilon_2 = \frac{\Gamma_2 \pm J\sqrt{4(\beta_x + \iota_x)(\beta_y - \iota_y) - \Gamma_2^2}}{2(\beta_x + \iota_x)}, \quad |\varepsilon_2| = \sqrt{\frac{\beta_y - \iota_y}{\beta_x + \iota_x}} \qquad (6.26a)$$

and

$$q_2 = \frac{\beta_y - \iota_y}{\beta_x + \iota_x} \, . \qquad (6.26b)$$

Therefore, the growth factor q for the entire 2-D ADI-FDTD algorithm is

$$q = q_1 q_2 = \frac{(\beta_x - \iota_x)(\beta_y - \iota_y)}{(\beta_x + \iota_x)(\beta_y + \iota_y)} \le 1 \, . \qquad (6.27)$$

Since equation (6.27) is always satisfied the ADI-FDTD algorithm is unconditionally stable. By analogy the growth factor for a 3-D ADI-FDTD algorithm is given by

$$q = q_1 q_2 q_3 = \frac{(\beta_x - \iota_x)(\beta_y - \iota_y)(\beta_z - \iota_z)}{(\beta_x + \iota_x)(\beta_y + \iota_y)(\beta_z + \iota_z)} \le 1 \, . \qquad (6.28)$$

7.0 Implementation Issues

7.1 Initialization

To begin computing with an FDTD algorithm, an initialization of the algorithm is necessary [16]. This entails introducing the problem initial condition(s) into the computation lattice. We specify the field function values at zero time, and use these zero-time field values to compute the first set of field values using the finite difference equations for the time stepping procedure.

For instance, initializing the finite difference approximation of the scalar wave equation implies setting n equal to zero in (6.2). This gives the field values at time $t = \Delta t$, ψ1(i,j,k), in terms of the field values at t = 0, ψo(i,j,k), and at the time $t = -\Delta t$, ψ-1(i,j,k). A non-homogeneous initial condition, say ψ(x,y,z,0) = f(x,y,z), where f(x,y,z) is a given space function, provides the values of ψo(i,j,k) at all interior nodal points. Mathematically, to determine the terms ψ-1(i,j,k) a second initial condition $\partial\psi(x,y,z,0)/\partial t$ = g(x,y,z), where g(x,y,z) is a given space function, is required. If this additional initial condition is not given, it is usual to assume that ψ-1(i,j,k) = 0. Physically, this assumption implies that the wave source has no influence in the problem domain prior to the time t = 0.

If the problem is of the radiation type, then initialization entails the introduction of a source into the computation lattice. The source can be modelled by locally specifying a time function, say f(t), in a region within the computation lattice. A monochromatic (time harmonic) point source of wavelength λ that is switched on at time t = 0 may be modelled by adding a term, say

$$f\left(n\Delta t\right) = C\,sin\left(\frac{2\pi\upsilon n\Delta t}{\lambda}\right)H\left(t\right), \tag{7.1}$$

at a chosen lattice point (i,j,k) to the (relevant) time stepping relation, where H(t) denotes the Heaviside unit step function. Such a source generates circular waves which propagate outwards from the selected point.
In a problem involving scattering, the source is an incident wave which approaches a target within the problem domain from 'infinity' and after interacting with the target returns to infinity. Numerically, an incident wave can be modelled using a suitable plane (or surface) of the computation lattice. The numerical source is a time harmonic function in continuous scattering or a Gaussian pulse in transient scattering. The simplest procedure is to apply the selected time function at all nodes along one lattice boundary (which may be viewed as a wavefront). This computation lattice boundary would then 'radiate' (Huygens' like) the desired wave into the computation lattice.

7.2 Lattice Truncation

Problems which involve wave phenomenon may be in either a closed or an open spatial region. Instances of wave scattering and radiation are typical of a problem with an open domain. In scattering a wave approaches from a distant source (usually assumed to be at infinity) and after interacting with a target returns to infinity. However waves in a waveguide which connects an antenna to a receiver or waves in a resonant cavity are representative of a problem with a closed spatial domain. When a problem involves a closed spatial region the size of the computation lattice corresponds to that of the physical problem, and at the lattice boundary the appropriate problem boundary condition is implemented. For example, in solving for the transverse magnetic wave modes of a metallic rectangular waveguide whose cross-section encloses the solution region, Ez = 0 on the walls of the metal which is the lattice boundary. Similarly, while considering waves on a stretched string clamped at both ends, the displacement of the string is equal to zero at the lattice boundary, which is the physical boundary condition at the ends of the string.
In dealing with a problem having an open spatial region, it is necessary to devise a means of truncating the computation lattice. This is because computer capacity is finite. A lattice truncation scheme is implemented at the lattice boundary (essentially) to simulate an extension of the computation lattice to infinity, because the spatial region of the physical problem is infinite. A lattice truncation scheme may act as an absorbing sheet of numerical waves incident on it from the interior of the computation lattice, in which case the lattice truncation scheme is viewed as a lossy medium adjacent to the lattice boundary. The notion of using 'absorbing sheets' as a lattice truncation scheme in models of electromagnetic waves has gained renewed interest, in the form of the perfectly matched layer.
A lattice truncation scheme should reduce non-physical reflection of outgoing waves and permit the numerical solution to remain valid at all times. Though it is the desire that no reflection occurs at an artificial lattice boundary upon implementing a lattice truncation scheme, this is not achieved in practice because a lattice truncation scheme is, in general, an approximation. A very simple lattice truncation scheme imposes the Dirichlet boundary condition [17] at the lattice boundary. For such an arbitrary assumption to yield any degree of accuracy, the lattice boundary must be far removed from any object within the computation lattice. This causes an increase in computer time and memory required for the solution.
Details on implementing a lattice truncation scheme depend on the kind of problem that is modelled. However, while using the FDTD method it is important to note whether: a) the problem domain includes a source of radiation, or b) the wave enters the problem domain from sources located outside. Though, whichever is the case, the lattice truncation condition applies to only outgoing waves. Lattice truncation conditions for differential equation based models have been called absorbing boundary conditions [18], radiation boundary conditions [19], or the perfectly matched layers [20]. We shall consider only those conditions derived from one-way wave equation differential operators.

7.3 Guidelines for Applying Lattice Truncation

Coupling the discrete lattice truncation scheme to the interior region finite difference equations can make the entire algorithm numerically unstable. Finite difference models, even of non-dispersive equations, are necessarily dispersive [21]. Thus for the finite difference approximation, the energy associated with different frequency components will travel at different phase velocities, even when the differential equation is non-dispersive. For a hyperbolic system, the stability question is solved in principle by the theory of Gustafsson, Kreiss and Sundström [22], according to which an initial boundary value problem numerical model is stable if and only if:

a) The Courant stability condition is satisfied everywhere in the computation lattice.
b) The model (including the lattice truncation condition) admits no plane wave solutions that grow from one time step to the next by a growth factor q, with $|q| > 1$.
c) The model admits no wave solutions with frequency components which support active radiation from the lattice boundary to the interior of the computation lattice.

Lattice truncation conditions should be based on the above requirements. Additionally, in other to successfully apply a lattice truncation condition it is necessary that:

a) The distance from any object(s) in the computation lattice to the lattice boundary be large so that only the radiation field dominates in the neighbourhood of the lattice boundary.
b) The origin of the computation lattice be located near the centre of any object.
c) The minimum distance across the computation lattice should be at least twice the maximum dimension of the object of interest within the domain.

7.4 One-Way Wave Equation Differential Operator

A one-way wave equation is a partial differential equation which permits wave propagation in only one direction, and the operator arising from it is called a one-way wave equation differential operator. Usually, a one-way wave equation (or operator) is derived by an algebraic approximation of a radical function that is part of a pseudodifferential equation (or operator). There are three methods by which these pseudodifferential equations may be derived: i) operator factoring of a wave equation, ii) theory of reflection of singularities for solutions of differential equations, iii) using a dispersion relation. The resulting lattice truncation condition in this case is often called an absorbing boundary condition because, in a sense, it absorbs any wave which is incident on it. Such lattice truncation conditions are a natural choice while implementing the FDTD method in rectangular coordinates. Below, pseudodifferential equations will be derived by factoring a wave equation. Re-write the homogeneous scalar wave equation (6.2), in two dimensions using rectangular coordinates, as

$$D_{xx}\psi + D_{yy}\psi - \frac{1}{\upsilon^2} D_{tt}\psi = 0 , \qquad (7.2)$$

where Dxx, Dyy and Dtt denote second partial derivatives with respect to x, y and t, respectively. The wave operator in (7.2) is

$$L \equiv D_{xx} + D_{yy} - \frac{1}{\upsilon^2} D_{tt} . \qquad (7.3)$$

The wave equation may then be written as

$$L\psi = 0 . \qquad (7.4)$$

The wave operator L can be factored allowing us to write (7.4) as

$$L\psi = L^{+}L^{-}\psi = 0 , \qquad (7.5a)$$

where the operators L+ and L- are defined as

$$L^{\pm} \equiv D_x \pm \frac{D_t}{\upsilon}\sqrt{1-s^2} , \qquad (7.5b)$$

$$s = \upsilon D_y / D_t . \qquad (7.5c)$$

Because of the radical in the definitions of L+ and L- in (7.5b), L+ and L- are called pseudo-differential operators. Observe from (7.5a) that

$$L^{-}\psi = 0 \text{, and} \qquad (7.6a)$$

$$L^{+}\psi = 0 . \qquad (7.6b)$$

The presence of the radicals in the pseudodifferential equations (7.6) inhibits a direct numerical implementation of either (7.6a) or (7.6b) because they are non-local both in space and time, except in one-dimensional space. To advance one time step at a single space point requires information from all previous times over the entire boundary.
In three-dimensional rectangular coordinates, the homogeneous scalar wave equation

$$D_{xx}\psi + D_{yy}\psi + D_{zz}\psi - \frac{1}{\upsilon^2} D_{tt}\psi = 0 \qquad (7.7)$$

has the associated wave operator

$$L \equiv D_{xx} + D_{yy} + D_{zz} - \frac{1}{\upsilon^2} D_{tt} . \qquad (7.8)$$

Upon factoring (7.7) we have (7.5b), but with s in this case defined as

$$s = \sqrt{\left(\frac{\upsilon D_y}{D_t}\right)^2 + \left(\frac{\upsilon D_z}{D_t}\right)^2} . \qquad (7.9)$$

Algebraic approximations of the radical in the definition of the pseudo-differential operator(s) lead to local approximate lattice truncation conditions that can be implemented numerically with finite difference schemes. However, care must be taken concerning which algebraic approximation to use as some lead to ill-posed problems that are numerically unstable.

Often it is necessary to check the well-posedness of the resulting lattice truncation condition using general algebraic normal mode analysis specialized for the wave equation. The resulting lattice truncation conditions are not exact in that a small amount of reflection does develop as waves pass through the lattice boundary.

7.5 Taylor Polynomial Approximations of Pseudodifferential Operators

Consider the pseudodifferential operators $L^{\pm}$ in (7.5b) and approximate the radical using a Taylor polynomial expression.

(a) First approximation: Using a one-term Taylor approximation, $\sqrt{1-s^2} \approx 1 + \vartheta\left(s^2\right)$, in (7.6) leads to the lattice truncation conditions

$$\left(D_x - \frac{1}{\upsilon} D_t\right)\psi = 0, \quad x = 0$$

$$\left(D_x + \frac{1}{\upsilon} D_t\right)\psi = 0, \quad x = a. \tag{7.10}$$

The lattice truncation conditions (7.10) are exact for calculations which involve the one-dimensional wave equation.

(b) Second approximation: Using a two-term Taylor approximation, $\sqrt{1-s^2} \approx 1 - ½s2 + \vartheta(s^4)$ in (7.6) with s given by (7.5c) yields the lattice truncation conditions

$$\left(D_{xt} - \frac{1}{\upsilon} D_{tt} + \frac{\upsilon}{2} D_{yy}\right)\psi = 0, \quad x = 0$$

$$\left(D_{xt} + \frac{1}{\upsilon} D_{tt} - \frac{\upsilon}{2} D_{yy}\right)\psi = 0, \quad x = a. \tag{7.11}$$

For the wave equation in three-dimensions, where s is given by (7.9), the corresponding lattice truncation conditions are

$$\left(D_{xt} - \frac{1}{\upsilon} D_{tt} + \frac{\upsilon}{2} D_{yy} + \frac{\upsilon}{2} D_{zz}\right)\psi = 0, \quad x = 0$$

$$\left(D_{xt} + \frac{1}{\upsilon} D_{tt} - \frac{\upsilon}{2} D_{yy} - \frac{\upsilon}{2} D_{zz}\right)\psi = 0, \quad x = a. \tag{7.12}$$

The partial differential equations (7.10 – 7.12) are examples of one-way wave equations. These partial differential equations are replaced with finite difference approximations that are second-order accurate and consistent with the interior region finite difference equations. The lattice truncation conditions arising from them are of first- and second-order; a terminology based on the highest order of the partial derivatives which each contains. Higher-order approximations to the radical have been proposed as a means to derive more accurate lattice truncation conditions

7.6 Corner Boundary Condition

The finite difference approximation of the lattice truncation condition together with the interior region finite difference equations cannot be implemented at corner points in a rectangular lattice, since some of the required field data to be used in the finite difference expressions are outside the computation lattice and not available. Generally, a finite difference scheme for a corner point should: utilize available field data in the computation lattice, yield acceptable low levels of reflection of outgoing numerical wave analogues, and be numerically stable.

The issue of resolving the problem at a corner can be addressed in different ways. Figure 3 illustrates a two-dimensional geometry for a simple and stable corner boundary condition, introduced by Taflove and Umashanker [7]. The corner boundary condition uses a first-order accurate propagation approach wherein the value of a corner field component, say W(0,0), is equated to the time-retarded value of an interior field, $\overline{W}$, located on the radial line connecting the corner point to the centre of the lattice. This approach assumes that each scattered wave analogue is radially outgoing at the corner point.

Suppose the space and time increments are related by $\upsilon\Delta t = \Delta x/m$, where m is an integer greater than unity. Then for $\overline{W}$ at one cell width, Δx , inward along the radial line, the time retardation of the outgoing numerical-wave in propagating from $\overline{W}$ to W(0,0) is exactly m time steps. Hence define a corner boundary condition

$$W^{n+1}(0,0) = d_r \overline{W}^{n-m+1} \tag{7.13}$$

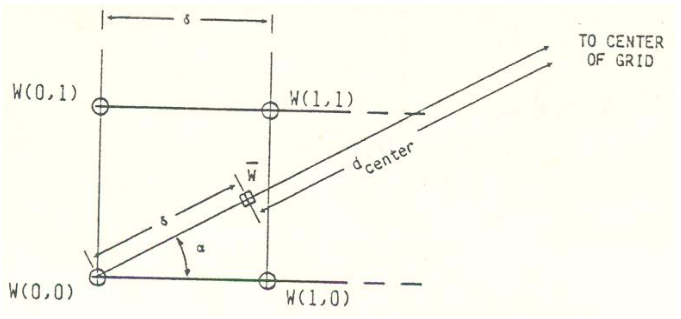

Figure 3: Points near the x = 0, y = 0 grid corner used in the special corner boundary condition.
where dr is the attenuation factor for a radially outgoing numerical-wave given, for 2-D, by

$$d_r = \left(\frac{d_{centre}}{d_{centre}+1}\right)^{1/2} \quad \text{and} \tag{7.14}$$

$$\overline{W}^{n-m+1} = (1-\sin\alpha)(1-\cos\alpha)W^{n-m+1}(0,0) + (1-\sin\alpha)\cos\alpha W^{n-m+1}(1,0)$$
$$+ \sin\alpha(1-\cos\alpha)W^{n-m+1}(0,1) + \sin\alpha\cos\alpha W^{n-m+1}(1,1) \tag{7.15}$$

dcentre is the radial distance, in units of cell width, from $\overline{W}$ to the centre of the computation lattice, and α is the azimuthal angle of the radial line at W(0,0). dcentre is one-half the lattice diagonal minus one. Equation (7.15) results from a simple linear interpolation of the four field values surrounding $\overline{W}$ at the time step (n-m+1).

7.7 Plane Wave Condition

In modelling the exit of a wave from a computation lattice, it should be observed that a lattice truncation condition is implemented only along any surface (or contour) parallel to a wavefront at the lattice boundary. Consequently, for a finite difference model in spherical (or plane-polar) coordinates, a lattice truncation condition is implemented along the entire lattice boundary. However, for a model in rectangular coordinates in which an incident wave enters the computation lattice from sources located outside, the lattice truncation scheme consists of two components: (1) a lattice truncation condition is implemented along those planes (or lines) normal to the propagation direction; and (2) a plane-wave condition is implemented along the remaining planes (or lines) parallel to the propagation direction. The plane-wave condition is a Neumann-type boundary condition [17]:

$$\left.\frac{\partial\psi}{\partial n}\right|_c = 0. \tag{7.16}$$

This requirement represents the fact that an incident plane wave, which is undisturbed by any obstacle, continues propagating as a plane wave. The plane-wave condition is used for computing the field values at the boundary nodes normal to a wavefront.

8.0 Concluding

Validation studies have been omitted in this paper due to space constraint. Results of such computations can be found in some of the cited references, where the FDTD method had been applied to model various types of wave phenomena. It is hoped this discourse provides the basic knowledge necessary for using the method.

9.0 Acknowledgement

I would like to acknowledge the assistance provided by Dr M. A. Morgan (Naval Postgraduate School, Monterey, CA) and Dr Mario A. Fusco (Ministère de la Défense, Délégation Générale pour l' Armement, Paris) while I grappled with the method. Also, I thank Prof J. O. A. Idiodi (Editor-in-Chief) who offered me the privilege to submit an invited paper.

10.0 References

[1] Kane S. Yee (1966) "Numerical simulation of initial boundary value problems involving Maxwell equations in isotropic media" IEEE Trans Antennas Propagat. AP-14, May, 302 – 307.

[2] L. Lapidus and G. F. Pinder (1999) Numerical Solutions of Partial Differential Equations in Science and Engineering. John Wiley: New York.

[3] Allen Taflove (2010) Computaional Electrodynamics: The Finite Difference Time Domain Method, 2nd ed. Artech House: Norwood, MA.

[4] Eghuanoye Ikata (2000) Electromagnetic Waves – Basic Theory and Computational Technique. National Broadcasting Commission: Abuja.

[5] A. F. Bermant and I. G. Aramanovich (1989) Mathematical Analysis – a brief course for engineers. Mir Publishers: Moscow.

[6] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders (1982) Fundamentals of Acoustics, 3rd ed. John Wiley: New York.

[7] M. A. Morgan, Ed. (1989) Finite Element and Finite Difference Methods in Electromagnetic Scattering, PIER vol. 2. Elsevier Science: Amsterdam.

[8] A. Taflove, K. R. Umashankar, B. Beker, F. Harfoush and K. S. Yee (1988) "Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens" IEEE Trans Antennas Propagat AP-36, February, 247 – 257.

[9] Richard Haberman (1987) Elementary Applied Partial Differential Equations – with Fourier series and boundary value problems, 2nd ed. Prentice Hall: New Jersey.

[10] Eghuanoye Ikata and Goza Tay (1998) "Finite difference time domain acoustic wave algorithm" IL NUOVO CIMENTO 20D, N.12, 1779 - 1793.

[11] Eghuanoye Ikata (2005) "FDTD model of acoustic wave interaction with soft targets" J. Nigeria Assoc. Math. Phys. vol. 9, November, 545 – 560.

[12] J. H. Ferziger (1998) Numerical Methods for Engineering Applications, 2nd ed. John Wiley: New York.

[13] Takefumi Namiki (1999) "A new FDTD algorithm based on alternating direction implicit method" IEEE Trans Microw Theory Tech MTT-47, October, 2003 – 2007.

[14] Eghuanoye Ikata (2010) "Alternating direction implicit finite difference time domain acoustic wave algorithm" J. Nigeria Assoc. Math. Phys. vol. 17, November, 197 – 206.

[15] Mario A. Fusco (1990) "FDTD algorithm in curvilinear coordinates" IEEE Trans. Antennas Propagat. AP-38, January, 76 – 89.

[16] C.- Y. Chow (1983) An Introduction to Computational Fluid Mechanics, corrected ed. Seminola Publishing: Boulder, CO.

[17] C. R. Chester (1971) Techniques in Partial Differential Equations. McGraw-Hill: Kogakusha, Japan.

[18] B. Engquist and A. Majda (1977) "Absorbing boundary conditions for numerical simulation of waves" Math. Comp. vol. 31, July. 629 – 651.

[19] A. Bayliss and E. Turkel (1980) "Radiation boundary conditions for wave-like equations" Comm. Pure Appl. Maths. vol. 33, 707 – 725.

[20] J.- P. Berenger (1994) "A perfectly matched layer for the absorption of electromagnetic waves" J. Comp. Phys. vol. 114, October, 185 – 200.

[21] An Ping Zhao (2002) "Analysis of the numerical dispersion of 2-D alternating direction implicit FDTD method" IEEE Trans Microw. Theory Tech. MTT-50, April, 1156 – 1164.

[22] B. Gustafsson, H. O. Kreiss and O. Sundström (1972) "Stability theory of difference approximations for mixed initial boundary value problems II" Math. Comp. vol. 26, 649 – 686