Journal of the Nigerian Association of Mathematical Physics
***Volume* 17 (November, 2010), pp 197 - 206**

Alternating Direction Implicit Finite Difference Time Domain Acoustic Wave Algorithm

Eghuanoye Ikata

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

Abstract

A time domain numerical technique is presented for the modelling of acoustic wave phenomena. The technique is an adaptation of the alternating direction implicit finite difference time domain method. The stability condition for the algorithm is given. Simple illustrations of propagation in an infinite homogeneous medium are provided which reveal the influence of the Courant number on a simulation and the basic attributes of the algorithm.

PACS: 02.70.Bf **–** Finite difference methods.

1.0 Introduction

The finite difference time domain (FDTD) method is an attractive technique for the prediction of field behaviour in wave interaction problems in which the time parameter appears explicitly. The basic theory of the FDTD method applied to electromagnetic wave problems can be found in [1]. An adaptation of the FDTD method for the simulation of acoustic wave is given in [2]. An FDTD algorithm is formulated by finite differencing a pair of first-order partial differential equations which represent the wave, leading to recursive time stepping expressions where the field values at the present time is deduced from field values at previous time steps. However, the traditional FDTD method [3] is based on an explicit finite difference algorithm and consequently the allowed time increment is bound by the Courant-Friedrichs-Lewy (CLF) stability condition [4]. 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 acceptable/accurate solution. To overcome this drawback, the alternating direction implicit finite difference time domain (ADI-FDTD) method has been introduced for electromagnetic wave simulations [5].

In an ADI-FDTD method the set of finite difference equations for updating the wave field values 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. These sub-sets represent alternations in the computation directions. The alternations in the computation directions may be made in respect of either (i) the spatial coordinate directions [5] or (ii) the sequence of terms on the right-hand-side of the partial differential equations [6]. 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).

Here the ADI-FDTD method is adapted to model acoustic wave phenomena. For pedagogic reasons, the formulation of the ADI-FDTD acoustic wave algorithm begins with a one-dimensional situation. Numerical stability analysis of the algorithm is done using the Von Neumann method [7].

2.0 Physical Model

An acoustic wave is a variation of pressure and density which propagates through a compressible medium. In a medium that exhibits little restraint to deformation, the restoring force responsible for acoustic wave propagation is due to change in pressure. Using a field approach to describe the wave, the relevant field functions are medium density $\rho(\mathbf{r},t)$, particle velocity $\mathbf{V}(\mathbf{r},t)$ and pressure $p(\mathbf{r},t)$, where $\mathbf{r}$ is the position vector and t is time. The field functions $\rho(\mathbf{r},t)$, $\mathbf{V}(\mathbf{r},t)$ and $p(\mathbf{r},t)$ which represent the wave are related via the equation of continuity, equation of

Corresponding author: Tel. +2348057308476

motion, and equation of state. Mathematically, wave phenomenon is represented by a wave equation. An acoustic wave equation is derived by using the equation of continuity and equation of motion.

Ignoring the temperature dependence of viscosity in a Newtonian fluid, the general equation of motion is the Navier-Stokes (momentum) equation [8] which can be written as,

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t}+\left(\mathbf{V}.\nabla\right)\mathbf{V}\right]=\rho\mathbf{g}-\nabla p+\left(\eta_B+\frac{4}{3}\eta\right)\nabla\left(\nabla.\mathbf{V}\right)-\eta\nabla x\nabla x\mathbf{V}\,, \tag{2.1}$$

where η is the shear viscosity, η_B is the bulk viscosity and **g** is the body force per unit mass. In adapting the Navier-Stokes equation for acoustic wave phenomenon, it is appropriate to admit compressibility and discard circulation; these concepts are represented in (2.1) by the divergence and curl terms (of the velocity) respectively. For an acoustic wave in a source-free medium, the term representing the body force will not apply as it denotes an external force.

Discarding the circulation is equivalent to setting $\nabla x\mathbf{V}=0$. The vector identity

$$\nabla x\nabla x\mathbf{V}=-\nabla^2\mathbf{V}+\nabla\left(\nabla.\mathbf{V}\right) \tag{2.2a}$$

implies that

$$\nabla\left(\nabla.\mathbf{V}\right)=\nabla^2\mathbf{V}\quad,\text{when}\quad\nabla x\mathbf{V}=0. \tag{2.2b}$$

When the wave is of small amplitude, we can ignore the non-linear term in (2.1) and the equation of motion for an acoustic wave in a Newtonian fluid reduces to

$$\rho_o\frac{\partial \mathbf{V}}{\partial t}=-\nabla P+\eta_a\nabla^2\mathbf{V}\quad,\quad\eta_a=\eta_B+4\eta/3\,, \tag{2.3}$$

where ρ_o, the constant equilibrium density of the fluid, is required by the small amplitude approximation [9], P (= p – p_o) is an acoustic pressure and p_o the equilibrium pressure.

For small amplitude waves, assuming an adiabatic equation of state, the equation of continuity can be written as [9]

$$\frac{\partial P}{\partial t}=-B\nabla.\mathbf{V}\,, \tag{2.4}$$

where B is the adiabatic bulk modulus of the fluid.

The two coupled linear partial differential equations (2.3) and (2.4) describe an acoustic wave in a Newtonian fluid, and represent the acoustic wave equation. The pair of coupled equations (2.3) and (2.4) is a system of four scalar partial differential equations governing four unknown scalar field functions: namely, three velocity components and the acoustic pressure. The four coupled partial differential equations in rectangular coordinates (x, y, z) are:

$$\rho_o\frac{\partial V_x}{\partial t}=-\frac{\partial P}{\partial x}+\eta_a\frac{\partial^2 V_x}{\partial x^2} \tag{2.5a}$$

$$\rho_o\frac{\partial V_y}{\partial t}=-\frac{\partial P}{\partial y}+\eta_a\frac{\partial^2 V_y}{\partial y^2} \tag{2.5b}$$

$$\rho_o\frac{\partial V_z}{\partial t}=-\frac{\partial P}{\partial z}+\eta_a\frac{\partial^2 V_z}{\partial z^2} \tag{2.5c}$$

$$\frac{\partial P}{\partial t}=-B\left(\frac{\partial V_x}{\partial x}+\frac{\partial V_y}{\partial y}+\frac{\partial V_z}{\partial z}\right) \tag{2.5d}$$

The system of partial differential equations (2.5) is the basis for a finite difference time domain algorithm describing a general acoustic wave problem.

3.0 Numerical Model

A finite difference time domain model approximates a continuous wave field in space-time by sampled data at points in a finite space-time lattice. The unit cell appropriate for an acoustic wave is given in [2, 10]; the field placement scheme on the cell is slightly different from that on a Yee cell [3]. The partial derivatives in the differential equations which define a problem are approximated using finite differences and, consequently, difference equations replace the differential equations describing the problem. The common FDTD symbolism introduced by Yee denotes a function of space-time as

Corresponding author: Tel. +2348057308476

$$F^n(i, j, k) \equiv F(i\delta x,\ j\delta y, k\delta z, n\delta t) \qquad (3.1)$$

where i, j, k, and n are integers, δx, δy and δz are space increments along the respective axes, and δt is a time increment.

In an ADI-FDTD method, a space derivative is replaced with a centered difference approximation which is second-order accurate in the increment:

$$\frac{\partial F(x_o)}{\partial x} = \frac{F(x_o + 0.5\delta) - F(x_o - 0.5\delta)}{\delta} + \vartheta(\delta^2) \qquad (3.2a)$$

$$\frac{\partial^2 F(x_o)}{\partial x^2} = \frac{F(x_o + \delta) - 2F(x_o) + F(x_o - \delta)}{\delta^2} + \vartheta(\delta^2) \qquad (3.2b)$$

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

$$\frac{\partial F^n}{\partial t} = \frac{F^{n+0.5} - F^n}{0.5\delta t} + \vartheta(\delta t) \qquad (3.3)$$

3.1 ADI-FDTD 1-D Algorithm

To introduce the formulation, let us consider a 1-D situation for which the governing partial differential equations reduce to

$$\rho_o \frac{\partial V_x}{\partial t} = -\frac{\partial P}{\partial x} + \eta_a \frac{\partial^2 V_x}{\partial x^2} \qquad (3.4a)$$

$$\frac{\partial P}{\partial t} = -B\frac{\partial V_x}{\partial x}. \qquad (3.4b)$$

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 (3.4a) 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), V_x is updated using an explicit finite difference equation arising from

$$\rho_o \left.\frac{\partial V_x}{\partial t}\right|_{i+0.5}^{n} = -\left.\frac{\partial P}{\partial x}\right|_{i+0.5}^{n} + \eta_a \left.\frac{\partial^2 V_x}{\partial x^2}\right|_{i+0.5}^{n} \qquad (3.5)$$

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

$$\rho_o \left.\frac{\partial V_x}{\partial t}\right|_{i+0.5}^{n+0.5} = -\left.\frac{\partial P}{\partial x}\right|_{i+0.5}^{n+1} + \eta_a \left.\frac{\partial^2 V_x}{\partial x^2}\right|_{i+0.5}^{n+1} \qquad (3.6)$$

Thus for the system of equations (3.4) the two sub-procedures which constitute the ADI-FDTD algorithm are:

First sub-procedure (n → n+0.5)

$$\begin{aligned} V_x^{n+0.5}(i+0.5) = V_x^n(i+0.5) - 2d_x V_x^n(i+0.5) - a_x\left\{P^n(i+1) - P^n(i)\right\} + \\ + d_x\left\{V_x^n(i+1.5) + V_x^n(i-0.5)\right\} \end{aligned} \qquad (3.7a)$$

$$P^{n+0.5}(i) = P^n(i) - b_x\left\{V_x^n(i+0.5) - V_x^n(i-0.5)\right\} \qquad (3.7b)$$

where $a_x = \dfrac{\delta t}{2\rho_o \delta x}$, $b_x = \dfrac{B\delta t}{2\delta x}$, $d_x = \dfrac{\eta_a \delta t}{2\rho_o \delta x^2}$. (3.7c)

Corresponding author: Tel. +2348057308476

Second sub-procedure (n+0.5 → n+1)

$$V_x^{n+1}(i+0.5)=V_x^{n+0.5}(i+0.5)-2d_xV_x^{n+1}(i+0.5)-a_x\left\{P^{n+1}(i+1)-P^{n+1}(i)\right\}+ \\ +d_x\left\{V_x^{n+1}(i+1.5)+V_x^{n+1}(i-0.5)\right\} \tag{3.8a}$$

$$P^{n+1}(i)=P^{n+0.5}(i)-b_x\left\{V_x^{n+1}(i+0.5)-V_x^{n+1}(i-0.5)\right\} \tag{3.8b}$$

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 (3.8a) can not be used directly because of the presence of P^{n+1}. Thus (3.8b) is used to eliminate P^{n+1} in (3.8a) leading to:

$$-t_xV_x^{n+1}(i+1.5)+[1+2t_x]V_x^{n+1}(i+0.5)-t_xV_x^{n+1}(i-0.5)=V_x^{n+0.5}(i+0.5)- \\ -a_x\{P^{n+0.5}(i+1)-P^{n+0.5}(i)\} \quad , \quad t_x=a_xb_x+d_x \tag{3.8c}$$

The implicit expression (3.8c) is usually resolved using a tri-diagonal matrix algorithm, after which (3.8b) is evaluated. Observer 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.

3.2 ADI-FDTD 2-D Algorithm

In a two-dimensional situation, assuming the problem is independent of z, the governing partial differential equations (2.5) reduce to

$$\rho_o\frac{\partial V_x}{\partial t}=-\frac{\partial P}{\partial x}+\eta_a\frac{\partial^2 V_x}{\partial x^2} \tag{3.9a}$$

$$\rho_o\frac{\partial V_y}{\partial t}=-\frac{\partial P}{\partial y}+\eta_a\frac{\partial^2 V_y}{\partial y^2} \tag{3.9b}$$

$$\frac{\partial P}{\partial t}=-B\left(\frac{\partial V_x}{\partial x}+\frac{\partial V_y}{\partial y}\right) \tag{3.9c}$$

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 (3.10) and the second sub-procedure by equations (3.11).

First sub-procedure (n to n+0.5)

$$V_x^{n+0.5}(i+0.5,j)=V_x^{n}(i+0.5,j)-2d_xV_x^{n}(i+0.5,j)-a_x\left\{P^{n}(i+1,j)-P^{n}(i,j)\right\}+ \\ +d_x\left\{V_x^{n}(i+1.5,j)+V_x^{n}(i-0.5,j)\right\} \tag{3.10a}$$

$$V_y^{n+0.5}(i,j+0.5)=V_y^{n}(i,j+0.5)-2d_yV_y^{n+0.5}(i,j+0.5)-a_y\left\{P^{n+0.5}(i,j+1)- \right. \\ \left. -P^{n+0.5}(i,j)\right\}+d_y\left\{V_y^{n+0.5}(i,j+1.5)+V_y^{n+0.5}(i,j-0.5)\right\} \tag{3.10b}$$

$$P^{n+0.5}(i,j)=P^{n}(i,j)-b_x\left\{V_x^{n}(i+0.5,j)-V_x^{n}(i-0.5,j)\right\}- \\ -b_y\left\{V_y^{n+0.5}(i,j+0.5)-V_y^{n+0.5}(i,j-0.5)\right\} \tag{3.10c}$$

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 to n+1)

Corresponding author: Tel. +2348057308476

$$V_x^{n+1}(i+0.5,j)=V_x^{n+0.5}(i+0.5,j)-2d_xV_x^{n+1}(i+0.5,j)-a_x\{P^{n+1}(i+1,j)- \\ -P^{n+1}(i,j)\}+d_x\{V_x^{n+1}(i+1.5,j)+V_x^{n+1}(i-0.5,j)\} \tag{3.11a}$$

$$V_y^{n+1}(i,j+0.5)=V_y^{n+0.5}(i,j+0.5)-2d_yV_y^{n+0.5}(i,j+0.5)-a_y\{P^{n+0.5}(i,j+1)- \\ -P^{n+0.5}(i,j)\}+d_y\{V_y^{n+0.5}(i,j+1.5)+V_y^{n+0.5}(i,j-0.5)\} \tag{3.11b}$$

$$P^{n+1}(i,j)=P^{n+0.5}(i,j)-b_x\{V_x^{n+1}(i+0.5,j)-V_x^{n+1}(i-0.5,j)\}- \\ -b_y\{V_y^{n+0.5}(i,j+0.5)-V_y^{n+0.5}(i,j-0.5)\} \tag{3.11c}$$

Observe that equation (3.10b) in the first sub-procedure and (3.11a) in the second sub-procedure can not be solved directly. Hence they are re-expressed using the equations for $P^{n+0.5}$ and P^{n+1} in the respective sub-procedures: thus

$$-t_yV_y^{n+0.5}(i,j+1.5)+[1+2t_y]V_y^{n+0.5}(i,j+0.5)-t_yV_y^{n+0.5}(i,j-0.5)= \\ =V_y^n(i,j+0.5)-a_y\{P^n(i,j+1)-P^n(i,j)\}+c_{xy}\{V_x^n(i+0.5,j+1)- \\ -V_x^n(i-0.5,j+1)-V_x^n(i+0.5,j)+V_x^n(i-0.5,j)\} \tag{3.10d}$$

$$-t_xV_x^{n+1}(i+1.5,j)+[1+2t_x]V_x^{n+1}(i+0.5,j)-t_xV_x^{n+1}(i-0.5,j)= \\ =V_x^{n+0.5}(i+0.5,j)-a_x\{P^{n+0.5}(i+1,j)-P^{n+0.5}(i,j)\}+c_{xy}\{V_y^{n+0.5}(i+1,j+0.5)- \\ -V_y^{n+0.5}(i+1,j-0.5)-V_y^{n+0.5}(i,j+0.5)+V_y^{n+0.5}(i,j-0.5)\} \tag{3.11d}$$

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

The equations (3.10d) and (3.11d) are solved using a tri-diagonal matrix algorithm.

4.0 Stability Analysis

The ADI-FDTD algorithm is analyzed for numerical stability using Von Neumann method [7]. In this method, instantaneous values of the field functions distributed in space across the computation space are first Fourier transformed into waves in spatial spectral domain, to represent a spectrum of spatial sinusoidal modes. Assume, in 1-D, a harmonic representation of a field function

$$\psi^n(i)=\psi_o q^n \exp J(k_x i\delta x) \quad , \quad J=\sqrt{-1} \quad , \tag{4.1}$$

where k_x is wave number and q is a growth factor. If the magnitude of the growth factor for an FDTD algorithm is less than or equal to unity, then the algorithm is numerically stable, otherwise the algorithm is numerically unstable.

For the first sub-procedure (3.7), we substitute a harmonic representation (4.1) of the pressure and velocity fields into (3.7) to obtain

$$\left(q_1^{0.5}-1+2d_x-2d_x\cos\theta_x\right)V_x^n+J2a_x\sin(\theta_x/2)P^n=0 \tag{4.2a}$$

$$J2b_x\sin(\theta_x/2)V_x^n+\left(q_1^{0.5}-1\right)P^n=0 \quad , \quad \theta_x=k_x\delta x \quad , \tag{4.2b}$$

where q_1 is the growth factor for the first sub-procedure. Eliminating any one of the field functions from the pair (4.2) leads to a quadratic equation for the growth factor

$$\varepsilon_1^2-(2-\iota_x)\varepsilon_1+(\iota_x-\beta_x)=0 \quad , \quad q_1\equiv\varepsilon_1^2, \tag{4.3a}$$

where $$\beta_x=1+4a_xb_x\sin^2(\theta_x/2) \quad , \quad \iota_x=2d_x(1-\cos\theta_x)=4d_x\sin^2(\theta_x/2). \tag{4.3b}$$

Thus

$$\varepsilon_1=\frac{(2-\iota_x)\pm J\sqrt{4(\beta_x-\iota_x)-(2-\iota_x)^2}}{2} \quad , \quad |\varepsilon_1|=\sqrt{\beta_x-\iota_x} \tag{4.4a}$$

or

Corresponding author: Tel. +2348057308476

$$q_1 = |\varepsilon_1||\varepsilon_1| = (\beta_x - \iota_x)\,. \tag{4.4b}$$

Similarly, for the second sub-procedure (3.8) substituting the harmonic representation (4.1) leads to:

$$\left\{q_2^{0.5}\left(1+2d_x-2d_x\cos\theta_x\right)-1\right\}V_x^n + Jq_2^{0.5}\,2a_x\sin\left(\theta_x/2\right)P^n = 0 \tag{4.5a}$$

$$\mathrm{J}q_2^{0.5}\,2\mathrm{b}_x\sin(\theta_x/2)\mathrm{V}_x^n + \left(q_2^{0.5}-1\right)\mathrm{P}^n = 0 \tag{4.5b}$$

Eliminating anyone of the field functions from (4.5) gives the quadratic equation

$$(\beta_x + \iota_x)\varepsilon_2^2 - (2+\iota_x)\varepsilon_2 + 1 = 0 \quad , \quad q_2 \equiv \varepsilon_2^2 . \tag{4.6}$$

Thus

$$\varepsilon_2 = \frac{(2+\iota_x) \pm J\sqrt{4(\beta_x+\iota_x)-(2+\iota_x)}}{2(\beta_x+\iota_x)} \quad , \quad |\varepsilon_2| = \frac{1}{\sqrt{\beta_x+\iota_x}}, \tag{4.7a}$$

or

$$q_2 = |\varepsilon_2||\varepsilon_2| = (\beta_x + \iota_x)^{-1}. \tag{4.7b}$$

Since the growth factor for the entire 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

$$q = q_1 q_2 = (\beta_x - \iota_x)/(\beta_x + \iota_x) \le 1. \tag{4.8}$$

Given the values of β_x and ι_x in (4.3b), equation (4.8) is always satisfied and the ADI-FDTD algorithm is unconditionally stable. Observe from the definitions in (4.3b) that ι_x is a loss-term arising from the medium viscosity, and (4.8) implies that the wave decays with continued time-stepping in a viscous medium. In a non-viscous medium $\iota_x = 0$ (since $d_x = 0$ when

$\eta_a = 0$) and the growth factor for the algorithm reduces to unity, signifying that the algorithm is always stable.

4.1 Stability Analysis for 2-D Algorithm

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

$$\psi^n(i,j) = \psi_o q^n \exp J\left(k_x i\delta x + k_y j\delta y\right). \tag{4.9}$$

For the first sub-procedure (3.10) we substitute the wave representation (4.9) for the pressure and velocity field functions in (3.10) to obtain:

$$\left(q_1^{0.5} - 1 + \iota_x\right)V_x^n + J2a_x\sin\left(\theta_x/2\right)P^n = 0 \tag{4.10a}$$

$$\left[q_1^{0.5}\left(1+\iota_y\right)-1\right]V_y^n + Jq_1^{0.5}\,2a_y\sin\left(\theta_y/2\right)P^n = 0 \tag{4.10b}$$

$$J2b_x\sin\left(\theta_x/2\right)V_x^n + Jq_1^{0.5}\,2b_y\sin\left(\theta_y/2\right)V_y^n + \left(q_1^{0.5}-1\right)P^n = 0 \tag{4.10c}$$

where $\quad \iota_y = 4d_y\sin^2\left(\theta_y/2\right) \quad and \quad \theta_y = k_y\delta y\,.$

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 q_1:

$$\begin{aligned} &q_1^{1.5}\left[1+\iota_y+4a_yb_y\sin^2\left(\theta_y/2\right)\right]+q_1\left[\iota_x\iota_y+\iota_x+4\iota_xa_yb_y\sin^2\left(\theta_y/2\right)-\right.\\ &\quad\left.-4a_yb_y\sin^2\left(\theta_y/2\right)-2\iota_y-3\right]+q_1^{0.5}\left[3-2\iota_x+4a_xb_x\sin^2\left(\theta_x/2\right)+\right.\\ &\quad\quad\left.+4\iota_ya_xb_x\sin^2\left(\theta_x/2\right)+\iota_y-\iota_x\iota_y\right]+\left[\iota_x-1-4a_xb_x\sin^2\left(\theta_x/2\right)\right]=0. \end{aligned} \tag{4.11}$$

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:

$$(x-1)\left(ax^2+bx+c\right) = ax^3 + (b-a)x^2 + (c-b)x - c = 0. \tag{4.12}$$

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

Corresponding author: Tel. +2348057308476

$$b = ½\{c + a - (c – b) + (b – a)\}. \quad (4.13)$$

Now, for the cubic equation (4.11), upon extracting the unity solution we have the quadratic equation for the growth factor:

$$\left(\beta_y + \iota_y\right)\varepsilon_1^2 - \Gamma_1\varepsilon_1 + \left(\beta_x - \iota_x\right) = 0 \quad , \quad q_1 = \varepsilon_1^2 \ , \quad (4.14a)$$

where

$$\begin{aligned} \beta_y &= 1 + 4a_y b_y \sin^2\left(\theta_y / 2\right), \\ \Gamma_1 &= \iota_x\iota_y + \iota_x - \iota_y - 2 - 2\iota_y a_x b_x \sin^2\left(\theta_x / 2\right) + 2\iota_x a_y b_y \sin^2\left(\theta_y / 2\right) \end{aligned} . \quad (4.14b)$$

Solving equation (4.14) gives the growth factor of the first sub-procedure

$$\varepsilon_1 = \frac{\Gamma_1 \pm J\sqrt{4\left(\beta_y + \iota_y\right)\left(\beta_x - \iota_x\right) - \Gamma_1^2}}{2\left(\beta_y + \iota_y\right)}, \quad \left|\varepsilon_1\right| = \sqrt{\frac{\beta_x - \iota_x}{\beta_y + \iota_y}} \quad (4.15a)$$

or

$$q_1 = \frac{\beta_x - \iota_x}{\beta_y + \iota_y}. \quad (4.15b)$$

For the second sub-procedure (3.11), following the above method, the quadratic equation for the growth factor is

$$\left(\beta_x + \iota_x\right)\varepsilon_2^2 - \Gamma_2\varepsilon_2 + \left(\beta_y - \iota_y\right) = 0 \quad , \quad q_2 = \varepsilon_2^2 \ , \quad (4.16a)$$

where

$$\Gamma_2 = \iota_y\iota_x + \iota_y - \iota_x - 2 - 2\iota_x a_y b_y \sin^2\left(\theta_y / 2\right) + 2\iota_y a_x b_x \sin^2\left(\theta_x / 2\right). \quad (4.16b)$$

Thus

$$\varepsilon_2 = \frac{\Gamma_2 \pm J\sqrt{4\left(\beta_x + \iota_x\right)\left(\beta_y - \iota_y\right) - \Gamma_2^2}}{2\left(\beta_x + \iota_x\right)}, \quad \left|\varepsilon_1\right| = \sqrt{\frac{\beta_y - \iota_y}{\beta_x + \iota_x}} \quad (4.17a)$$

and

$$q_2 = \frac{\beta_y - \iota_y}{\beta_x + \iota_x}. \quad (4.17b)$$

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

$$q = q_1 q_2 = \frac{\left(\beta_x - \iota_x\right)\left(\beta_y - \iota_y\right)}{\left(\beta_x + \iota_x\right)\left(\beta_y + \iota_y\right)} \le 1. \quad (4.18)$$

Since equation (4.18) is always satisfied the ADI-FDTD algorithm is unconditionally stable.

5.0 Lattice Truncation

In an FDTD model, the size of the computation space is limited using a lattice truncation scheme [11]. The details on implementing a lattice truncation scheme depend on the problem. We observe that the finite difference equations presented in Section 3 are for updating the field values at interior lattice points, because those equations were derived by replacing a space derivative with a centered difference approximation, which require knowledge of the field at points outside the computation space.

To update the field values at a lattice boundary point we use a first-order absorbing boundary condition [12], along those planes of the lattice boundary normal to the propagation direction, which is discretized using a Mur differencing scheme [13]. However, because the absorbing boundary condition difference approximation has to be consistent with the interior region difference approximation, a space derivative is replaced with a centered difference while a time derivative is replaced with a forward difference.

At the wall x = 0, say, the absorbing boundary condition for the pressure field is [12]

Corresponding author: Tel. +2348057308476

$$\frac{1}{\upsilon}\frac{\partial P}{\partial t}=\frac{\partial P}{\partial x} \quad , \quad \upsilon=\sqrt{\frac{B}{\rho_o}} \cdot \tag{5.1}$$

We discretize (5.1) by applying a centered difference formula at a fictitious pressure point P(0.5, j) for each space derivative and relate it by a simple average to field values at actual pressure points via:

$$P(0.5, j)=\frac{P(0, j)+P(1, j)}{2} \cdot \tag{5.2}$$

Following the alternating direction implicit method, the absorbing boundary condition is discretized explicitly for the sub-step from n to (n+0.5), and implicitly for the sub-step from (n+0.5) to (n+1). Thus for use with the first sub-procedure we discretize as follows:

$$\left.\frac{1}{\upsilon}\frac{\partial P}{\partial t}\right|_{0.5, j}^{n}=\left.\frac{\partial P}{\partial x}\right|_{0.5, j}^{n} \quad , \tag{5.3a}$$

or

$$P^{n+0.5}(0, j)=-P^{n+0.5}(1, j)+(\gamma+1)P^{n}(1, j)-(\gamma-1)P^{n}(0, j), \quad \gamma=\upsilon\delta t/\delta x , \tag{5.3b}$$

where γ is the Courant number. And, for use with the second sub-procedure time marching from (n + 0.5) to (n + 1) we discretize as follows:

$$\left.\frac{1}{\upsilon}\frac{\partial P}{\partial t}\right|_{0.5, j}^{n+0.5}=\left.\frac{\partial P}{\partial x}\right|_{0.5, j}^{n+1} \quad , \tag{5.4a}$$

or

$$P^{n+1}(0, j)=\frac{\gamma-1}{\gamma+1}P^{n+1}(1, j)+\frac{1}{\gamma+1}\left[P^{n+0.5}(0, j)+P^{n+0.5}(1, j)\right] \cdot \tag{5.4b}$$

The pressure values along the remaining planes parallel to the direction of propagation are obtained using a plane wave (Neumann-type) boundary condition [2]; thus

$$P^{n+0.5}(i,0)=P^{n+0.5}(i,1), \qquad \text{first sub-procedure}, \tag{5.5a}$$

$$P^{n+1}(i,0)=P^{n+1}(i,1) , \qquad \text{second sub-procedure}. \tag{5.5b}$$

Also, because the equation of motion (2.3) includes a second-order partial derivative with respect to space, the finite difference equations in Section 3 are not applicable for updating the velocity values at velocity mesh boundary points (as they use a centered difference approximation). When applied to the column of points $V_x(0.5, j)$, those equations will require information for the non-existent points $V_x(-0.5, j)$. Consequently, at the velocity mesh boundary $V_x(0.5, j)$, say, we implement boundary conditions similar to those described above for the pressure field.

6.0 Source Implementation

Before the calculations implied by an FDTD algorithm begin, an initialization of the algorithm is necessary. This entails specifying the initial condition(s) of a problem in the computation space (i.e. a specification of the field values at zero time) and then using these zero-time field values to determine the first set of computable field values. The zero-time field values are assumed known in any problem, since they represent the initial condition, and must be suitably introduced into the computation space. The zero-time field values represent the field source.

A plane wave source is simulated by making any plane in the computation space "radiate" into the domain. Numerical experiments show that an easier simulation results when this "radiating" plane coincides with a computation boundary plane in the case of time harmonic plane waves. Transient plane waves may be simulated by using a Gaussian pulse

$$g(t)=\exp\left[-\left\{\frac{4(t-t_o)}{T}\right\}^2\right], \tag{6.1}$$

where T is the pulse width and t_o is the time lapse before the pulse enters the domain.

7.0 Numerical Results and Discussion

In order to demonstrate the ADI-FDTD algorithm, we simulate the propagation of an acoustic wave in sea water. The medium is assumed to be homogeneous and infinite. The fluid density and bulk modulus are taken to be 998.0 kgm^{-3} and 2.18 x 10^{9} Nm^{-2}, respectively. The simulations are for time harmonic and transient plane waves, and in both situations our concern is limited to the free propagation of the wave in the absence of any obstacles. The

Corresponding author: Tel. +2348057308476

transient wave is modelled using a Gaussian pulse with pulse width 0.15 ms. The computation space is a 2-D 100 x 100 grid of square cells, each of side 0.01 m. The time increment is dependent on the Courant number used in a particular test.

We show a sample of the results for the case of a transient plane wave, only, as these reveal the salient characteristics of the algorithm. In Figure 1 the Courant number is 1.0, corresponding to a time increment 6.776 μs. Figure 2 is for a Courant number 2.0, implying that for these the time increment is 13.532 μs. The results show the normalized pressure magnitude as a function of space-time.

Generally, we observe a small non-physical decay in the wave amplitude as the numerical method simulates propagation phenomenon. The pulse gradually loses amplitude (or energy) as it propagates through the computation space, even in a non-viscous medium. The amount of this non-physical decay increases slightly with an increase in the Courant number. Thus the algorithm is not energy conservative. This is most likely due to the fact that the discretization uses a forward difference in time. Hence the method belongs to the category called "upwind methods" which are known to be dissipative [4].

For a transient wave, the results show a trailing wiggle or 'ringing' (Figures 2b-d) with increasing computation time, if the Courant number is chosen to be greater than that allowed by the Courant-Friedrichs-Lewy criterion. In FDTD models such wiggles are usually attributed to the shape of the pulse (necessitating the use of a sufficiently 'smooth' pulse). That is why a Gaussian pulse is commonly recommended [14]. Having used a Gaussian pulse, the observed wiggle is likely to be due to an inadequate temporal resolution of the pulse occasioned by using a large time increment (when the Courant number is high). It is known that the temporal sampling rate (of a pulse) is indirectly related to the Courant-Friedrichs-Lewy criterion.

While exploring our ADI-FDTD code, it was observed that using the Courant number 0.9 (or 0.8) caused a *floating-point-overflow*, after some time steps, forcing the program to abort. Incidentally, this was associated with the evaluation of the Gaussian pulse at the particular time instant ($t = 174\delta t = 1.0596$ ms for Courant number = 0.9) and so may be a machine error unconnected with the ADI-FDTD algorithm. These experiments were done on an IBM-clone using an Intel Pentium IV microprocessor. However, the use of other Courant numbers, e.g. 0.5, 0.7 and 1.5, did not cause a floating-point-overflow.

8.0 Conclusion

We have derived the governing equations for applying the alternating direction implicit finite difference time domain method to an acoustic wave, in 2-D, together with the stability analysis for the ensuing algorithm. The fact that an ADI-FDTD algorithm is unconditionally stable is clearly shown by the equation of the growth factor, which also indicates the effect of a lossy medium on the algorithm, in general. It is worthy of note to observe that discretizing the absorbing boundary condition following the alternating direction implicit principle leads to difference equations which are distinct from those used with the conventional FDTD algorithm, where both the space and time derivatives are replaced with centered difference approximations.

The numerical technique has been shown to model propagation of acoustic wave in an infinite homogeneous medium. The liberty which the ADI-FDTD algorithm affords in regard to the choice of Courant number is also demonstrated. Though, care is required in using a Courant number that violates the Courant-Friedrichs-Lewy criterion. While in theory the algorithm is applicable to both lossless and lossy media, because of an inherent numerical dissipation, the alternating direction implicit finite difference time domain algorithm may not be suitable for the study of the effect a lossy medium on wave propagation.

Acknowledgement: Most of the reprints on ADI-FDTD were provided by Mr Nduka W. Aguyi of Obrikom Gas Plant, Nigerian Agip Oil Company, Omoku, Rivers State.

Corresponding author: Tel. +2348057308476

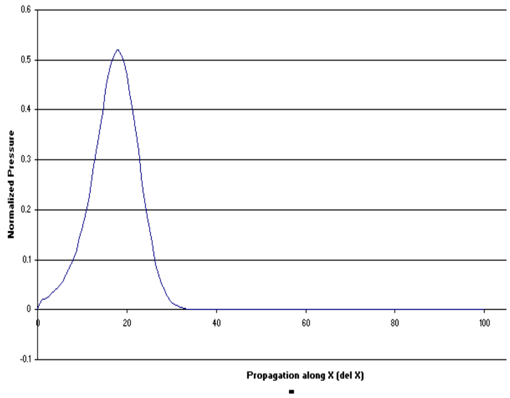

Figure 1a: Pulse after 30 time steps, clf = 1.0

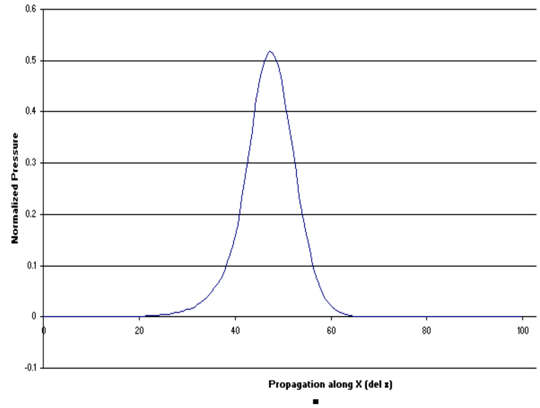

Figure 1b: Pulse after 60 time steps, clf = 1.0

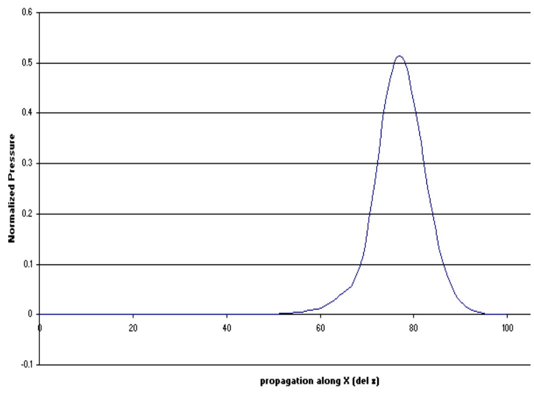

Figure 1c: Pulse after 90 time steps, clf = 1.0

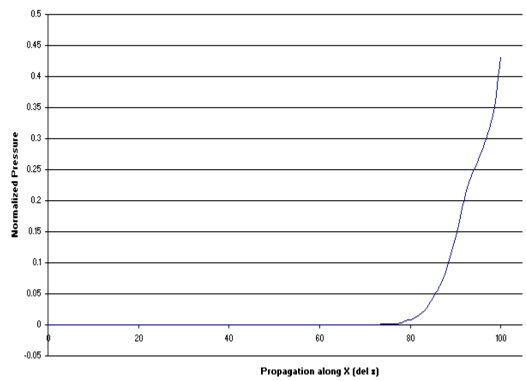

Figure 1d: Pulse after 120 time steps, clf = 1.0

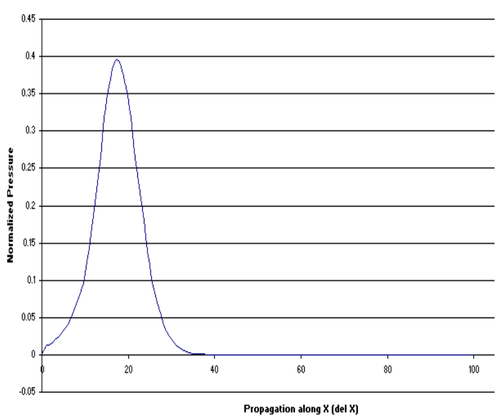

Figure 2a: Pulse after 15 time steps, clf = 2.0

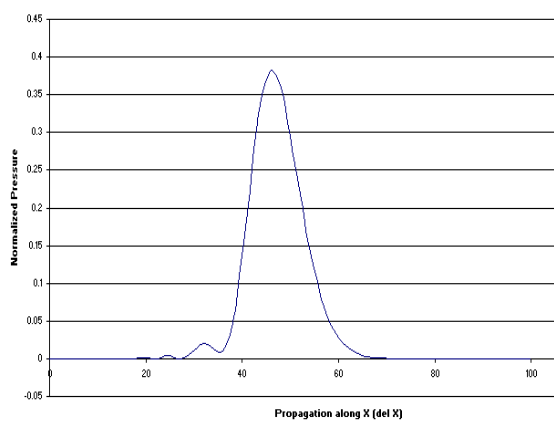

Figure 2b: Pulse after 30 time steps, clf = 2.0

Corresponding author: Tel. +2348057308476

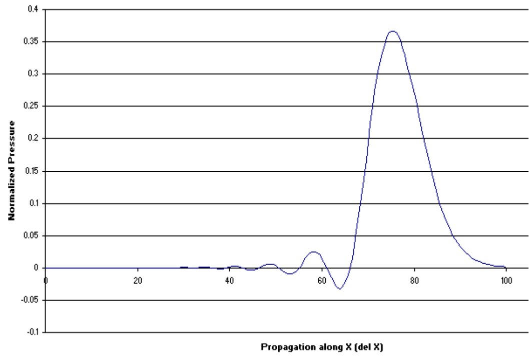

Figure 2c: Pulse after 45 time steps, clf = 2.0

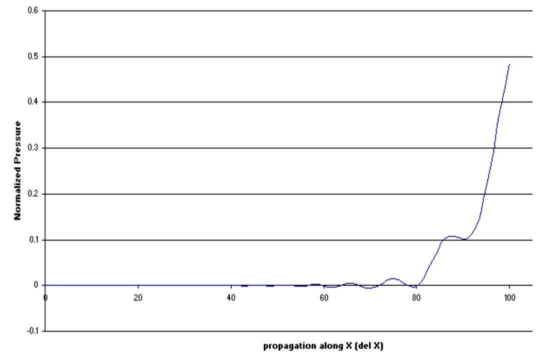

Figure 2d: Pulse after 60 time steps, clf =

References

[1] A. Taflove (1995) Computational Electrodynamics: The finite difference time domain method. Artech House: Norwood, MA.

[2] E. Ikata and G. Tay (1998) "*Finite difference time domain acoustic wave algorithm*" IL Nuovo Cimento **20D**, N.12, 1779 – 1793.

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

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

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

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

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

[8] J. H. Spurk (1997) Fluid Mechanics, English Trans. Springer: Verleg, Berlin.

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

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

[11] M. A. Morgan (1989) "*Principles of finite methods in electromagnetic scattering*" in Progress in Electromagnetic Research, vol. 2, edited by M. A. Morgan. Elsevier: Amsterdam.

[12] B. Enquist and A. Majda (1977) "*Absorbing boundary conditions for numerical simulation of waves*" Math Comput **31**, N.139, 629 -651.

[13] G. Mur (1981) "*Absorbing boundary conditions for the finite difference approximation of the time domain electromagnetic field equations*" IEEE Trans Electromag Compat **EMC-23**, November, 377 – 382.

[14] J. G. Blaschak and G. A. Kriegsmann (1988) "*A comparative study of absorbing boundary conditions*" J Comp. Phys. **77**, July, 109 - 139.

Corresponding author: Tel. +2348057308476