Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid

D. Botteldooren

Citation: The Journal of the Acoustical Society of America **95**, 2313 (1994); doi: 10.1121/1.409866
View online: http://dx.doi.org/10.1121/1.409866
View Table of Contents: http://asa.scitation.org/toc/jas/95/5
Published by the Acoustical Society of America

Articles you may be interested in

Finite-difference time-domain simulation of low-frequency room acoustic problems
The Journal of the Acoustical Society of America **98**, (1998); 10.1121/1.413817

Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid

D. Botteldooren
Laboratory for Electromagnetism and Acoustics, University of Gent, Gent, Belgium

(Received 1 April 1993; revised 15 October 1993; accepted 28 December 1993)

The finite-difference time-domain (FDTD) approximation can be used to solve acoustical field problems numerically. Mainly because it is a time-domain method, it has some specific advantages. The basic formulation of the FDTD method uses an analytical grid for the discretization of an unknown field. This is a major disadvantage. In this paper, FDTD equations that allow us to use a nonuniform grid are derived. With this grid, tilted and curved boundaries can be described more easily. This gives a better accuracy to CPU–resource ratio in a number of circumstances. The paper focuses on the new formulation and its accuracy. The problem of automatically generating the mesh in a general situation is not addressed. Simulations using quasi-Cartesian grids are compared to Cartesian grid results.

PACS numbers: 43.20.Bi, 43.20.Wd, 43.20.Fn

INTRODUCTION

Finite-difference time-domain (FDTD) approximation has been shown to be an interesting modeling technique for different types of wave-propagation problems.[1,2] The FDTD method has a few advantages over more conventional finite-element and boundary-element methods, which can be quite useful in solving acoustical and noise problems. First, the idea and implementation are relatively simple, which ensures a fast transfer to new types of problems. Second, the FDTD method is a direct time-domain method. This allows us to model realistic sound sources (including various types of random noise and impulses), transient phenomena, and nonlinear effects. The FDTD method can also be combined with digital electronics simulation to study transient phenomena in electronically controlled acoustical environments, such as active noise control applications.[3] For sufficiently damped acoustical systems, the FDTD method also allows us to obtain transfer functions in the frequency domain efficiently.

The FDTD method is based on a finite-difference approximation of both the space and time derivatives in the wave equations, which govern the problem. To develop the method, two field quantities are chosen. They will be calculated during alternating time steps. For air acoustics, these quantities are the sound pressure and the three components of the particle velocity. The basic implementation of the FDTD method assumes an analytical, mostly Cartesian grid of constant spacing over the whole modeling region. The introduction of the finite-difference approximation introduces some inaccuracies, both in describing the fields and in describing the boundaries. Lately, however, several authors reported progress in using both local grid refinements[4] and nonuniform grids[5] to reduce this inaccuracy, while keeping CPU requirements moderate.

In this paper an acoustical FDTD algorithm in a nonuniform grid is proposed that allows us to describe more accurately different types of curved and tilted boundaries or particular types of field profiles. However, special care is taken not to complicate the finite-difference equations so that the increase in computational effort required stays limited. If one wants to combine this requirement with accurate calculation results, the allowed grids must be restricted. Here only grids that can be called quasi-Cartesian are used. The examples shown prove that this requirement is not too restrictive on the possible applications. At the same time the FDTD concept remains simple and the implementation sufficiently quick to compete with the more widely used uniform grid implementations. The problem concerned with generating the mesh automatically is not solved here.

I. FDTD IN A QUASI-CARTESIAN GRID

A. FDTD equations

The traditional FDTD equations are derived in a Cartesian grid. As the quasi-Cartesian grid introduced here is a generalization of this grid, the traditional FDTD equations are briefly reviewed. In a Cartesian grid the acoustical pressure is determined at the grid positions $(i\,dx, j\,dy, k\,dz)$ at time $t=l\,dt$, with dx, dy, and dz the spatial discretization steps and dt the time discretization step. The indices i, j, and k mark the spatial points, the index l marks discrete time. The three components of the particle velocity are determined at positions

$$v_x((i\pm\tfrac{1}{2})dx, j\,dy, k\,dz),$$
$$v_y(i\,dx, (j\pm\tfrac{1}{2})dy, k\,dz), \tag{1}$$
$$v_z(i\,dx, j\,dy, (k\pm\tfrac{1}{2})dz),$$

and at intermediate time $t=(l+\frac{1}{2})dt$. In this grid the finite-difference forms of the basic equations of linear acoustics become[3]

$$v_x^{[l+0.5]}(i+\tfrac{1}{2},j,k)=v_x^{[l-0.5]}(i+\tfrac{1}{2},j,k)-\frac{dt}{\rho_0\,dx}\times[p^{[l]}(i+1,j,k)-p^{[l]}(i,j,k)],$$

 0001-4966/94/95(5)/2313/7/$6.00 © 1994 Acoustical Society of America

$$v_y^{[l+0.5]}(i,j+\tfrac{1}{2},k)=v_y^{[l-0.5]}(i,j+\tfrac{1}{2},k)-\frac{dt}{\rho_0\,dy}[p^{[l]}(i,j+1,k)-p^{[l]}(i,j,k)],$$

$$v_z^{[l+0.5]}(i,j,k+\tfrac{1}{2})=v_z^{[l-0.5]}(i,j,k+\tfrac{1}{2})-\frac{dt}{\rho_0\,dz}\times[p^{[l]}(i,j,k+1)-p^{[l]}(i,j,k)], \quad (2)$$

$$p^{[l+1]}(i,j,k)=p^{[l]}(i,j,k)-\frac{\rho_0 c^2 dt}{dx}[v_x^{[l+0.5]}(i+\tfrac{1}{2},j,k)-v_x^{[l+0.5]}(i-\tfrac{1}{2},j,k)]-\frac{\rho_0 c^2\,dt}{dy}[v_y^{[l+0.5]}(i,j+\tfrac{1}{2},k)-v_y^{[l+0.5]}(i,j-\tfrac{1}{2},k)]-\frac{\rho_0 c^2\,dt}{dz}[v_z^{[l+0.5]}(i,j,k+\tfrac{1}{2})-v_z^{[l+0.5]}(i,j,k-\tfrac{1}{2})],$$

where ρ_0 is the local air density and c is the local speed of sound. The finite-difference approximation to the derivatives in both time and space is in this case accurate to within second order in the steps dx, dy, dz, and dt.

To find the FDTD equations in a more general grid, the integral form of the equations of linear acoustics is a good starting point:

$$\iiint_{V_p}\frac{\partial p}{\partial t}\,dV=-\rho_0 c^2\oiint_{S_p}\mathbf{v}\cdot\mathbf{n}\,dS, \quad (3)$$

$$\iiint_{V_v}\frac{\partial \mathbf{v}}{\partial t}\,dV=-\frac{1}{\rho_0}\oiint_{S_v}p\mathbf{n}\,dS, \quad (4)$$

where V_p is an arbitrary volume of air associated with a local pressure p, S_p is the surface of this volume, V_v is a volume of air associated with the particle velocity v, S_v is the surface of this volume, $\mathbf{n}$ is the outer normal vector on these surfaces, c is the local speed of sound, and ρ_0 is the local air density.

Assume that a grid of well-chosen points is given that subdivides the whole simulation region and in which the sound pressure is to be calculated. The constraints on this grid are investigated later in this paper. To be able to solve Eq. (3) a volume $V_p^{[i]}$ has to be associated with each of the pressure points (labeled i). Equation (3) states a conservation of mass. Therefore each air molecule in the simulation region should be precisely in one of the volumes $V_p^{[i]}$ and thus taken into account exactly once. Hence $V_p^{[i]}$ is chosen equal to the volume of air that is closer to point i than to all other points. $V_p^{[i]}$ is sometimes referred to as the Voronoi cell around the point i. This cell is constructed by intersecting the planes orthogonal on the middle of the lines connecting the point i to all its nearest neighbors. The plane surfaces that limit $V_p^{[i]}$ are labeled $S_p^{[j]}$. From (3) and the considerations above, a good choice for the discrete description of the unknown particle velocities can be found: the particle velocity component normal to each of the surfaces $S_p^{[j]}$, averaged over this surface. The first generalized FDTD equation is now directly found by assuming the pressure change during the small time interval dt is constant over the entire volume $V_p^{[i]}$:

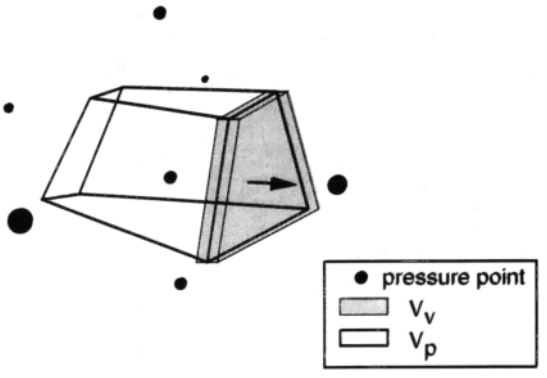


FIG. 1. Voronoi cell around a pressure point and velocity volume at one of its boundaries (gray).

$$p_i^{[l+1]}=p_i^{[l]}-\frac{\rho_0 c^2\,dt}{V_p^{[i]}}\sum_j v_{nj}^{[l+0.5]}S_p^{[j]}, \quad (5)$$

where the summation is over all surfaces that limit the Voronoi cell around point i and $v_{nj}^{[l+0.5]}$ is the average normal particle velocity on surface j at time $t=(l+\frac{1}{2})dt$. The sign of v_{nj} needs special care. The particle velocity v_{nj} is always located in the middle of exactly two pressure points $i-$ and $i+$. Equation (5) assumes v_{nj} has a positive sign if the particle velocity is pointing outside. If this condition is met for point $i-$ the sign in front of v_{nj} in (5) must be changed for the point $i+$. To solve this problem the convention is introduced that $\mathbf{v}$ on the surface $S_p^{[j]}$ points from the pressure point with lower index $i-$ to the pressure point with higher index $i+$. The appropriate sign can then be introduced in (5) depending on the order of the set of pressure discretization points. Remember that the pressures p_i are averaged over the Voronoi cell.

To find the second generalized FDTD equation we start from (4) and apply it to an appropriate volume. The unknown quantity is the average normal particle velocity on the surface $S_p^{[j]}$. Hence the volume $V_v^{[j]}$ to which the conservation of moment equation (4) will be applied is chosen equal to the prism with base $S_p^{[j]}$ and height from $-\delta$ to δ, where δ is arbitrary small. This volume is shown in Fig. 1. To evaluate (4) the pressure on all the surfaces S_v of this volume has to be known. The contributions of the small side surfaces are zero as the normal vector $\mathbf{n}$ on these surfaces is also normal to the unknown component of the particle velocity. Furthermore it is assumed that the pressure on the remaining two larger surfaces is only determined by the pressure in the neighboring volumes V_p.

To obtain the pressure on the large surfaces of V_v, the pressure $p(\mathbf{r})$ is approximated linearly near $S_p^{[j]}$ as

$$p(\mathbf{r})=p(\mathbf{o})+n_j\frac{\partial p}{\partial n_j}(\mathbf{o})+s_j\frac{\partial p}{\partial s_j}(\mathbf{o})+r_j\frac{\partial p}{\partial r_j}(\mathbf{o}), \quad (6)$$

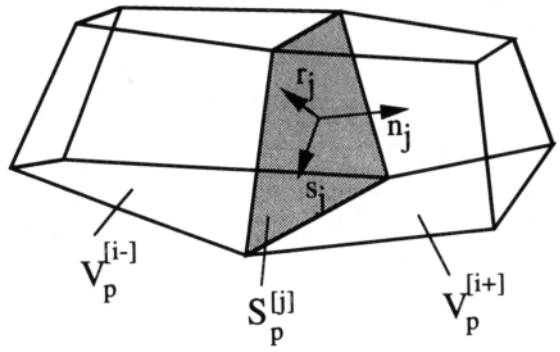


FIG. 2. Surface $S_p^{[j]}$, near volumes $V_p^{[i+]}$ and $V_p^{[i-]}$ and coordinate axes used for expansion of $p(\mathbf{r})$.

where the n_j axis is normal to $S_p^{[j]}$ and s_j and r_j are orthogonal directions in the plane of $S_p^{[j]}$ (see Fig. 2). The origin $\mathbf{o}$ is somewhere in the plane of $S_p^{[j]}$. Substitution of (6) in (4) gives, for small δ,

$$2\delta S_p^{[j]} \frac{\partial}{\partial t} v_{nj} = -\frac{1}{\rho} 2\delta S_p^{[j]} \frac{\partial p}{\partial n_j}(\mathbf{o}), \tag{7}$$

where v_{nj} is the average normal particle velocity on the surface $S_p^{[j]}$. The unknown quantity $\partial p/\partial n_j(\mathbf{o})$ has to be expressed as a function of p_{i+} and p_{i-}, the average pressure in the volumes $V_p^{[i+]}$ and $V_p^{[i-]}$. The linear approximation (6) is used. One obtains

$$\begin{aligned} p_{i+} - p_{i-} &= \frac{\partial p}{\partial n_j}(\mathbf{o}) \left(\frac{1}{V_p^{[i+]}} \iiint_{V_p^{[i+]}} n_j \, dV - \frac{1}{V_p^{[i-]}} \iiint_{V_p^{[i-]}} n_j \, dV \right) \\ &+ \frac{\partial p}{\partial s_j}(\mathbf{o}) \left(\frac{1}{V_p^{[i+]}} \iiint_{V_p^{[i+]}} s_j \, dV - \frac{1}{V_p^{[i-]}} \iiint_{V_p^{[i-]}} s_j \, dV \right) \\ &+ \frac{\partial p}{\partial r_j}(\mathbf{o}) \left(\frac{1}{V_p^{[i+]}} \iiint_{V_p^{[i+]}} r_j \, dV - \frac{1}{V_p^{[i-]}} \iiint_{V_p^{[i-]}} r_j \, dV \right). \end{aligned} \tag{8}$$

The three coefficients between the large parentheses are examined separately. Between the first set of large parentheses the distance between the centers of gravity of the neighboring cells, orthogonal on $S_p^{[j]}$, is recognized. This quantity will be called d_j. The other large parentheses contain projections of the distance between the centers of gravity on the plane of $S_p^{[j]}$. To be able to express $(\partial p/\partial n_j)(\mathbf{o})$ as a function of p_{i+} and p_{i-} the coefficients of $(\partial p/\partial s_j)(\mathbf{o})$ and $(\partial p/\partial r_j)(\mathbf{o})$ should vanish. This introduces a first condition on the mesh. For all surfaces between pressure cells, the projection in the plane of the surface of the distance between the centers of gravity of neighboring cells must be zero.

Finally, from (7) and (8),

$$v_{nj}^{[l+0.5]} = v_{nj}^{[l-0.5]} - \frac{dt}{\rho_0 d_j}(p_{i+}^{[l]} - p_{i-}^{[l]}), \tag{9}$$

where $p_{i+}^{[l]}$ ($p_{i-}^{[l]}$) is the higher (lower) index nearest neighbor pressure at time $t = l\,dt$ and d_j is the orthogonal distance between the nearest neighbor volumes centers of gravity.

Remark that, in the limit of a Cartesian grid with spacing dx, dy, and dz, Eqs. (5) and (9) reduce to Eq. (2).

B. Boundaries

At the boundaries of the simulation region, special forms of the FDTD equations have to be introduced. Since the aim of this paper is primarily to illustrate the advantages of the proposed FDTD method in nonuniform grids for acoustic field simulations, only a limited number of boundary types are introduced. The easiest boundary condition is evidently an acoustically hard surface. It requires that the normal particle velocity on the surface vanishes, which results in the condition $v_{nj} = 0$ on the boundary. The surfaces of the boundary Voronoi cells (where v_{nj} is defined) must thus coincide with the hard surface to describe its presence accurately. This requires a good choice of the pressure grid near a boundary. For this reason a layer of dummy pressure points is introduced at positions obtained by reflecting the nearest layer of pressure points around this boundary.

The boundary condition $v_{nj} = v(t)$, where $v(t)$ is a given function of time, can be used as a first-order approximation to a vibrating surface or a loudspeaker. Slightly more complicated is the description of a boundary with given normal impedance $Z = Z_0 + j\omega Z_1$. In the time domain this leads to the relation

$$p(t) = \left(Z_0 + Z_1 \frac{\partial}{\partial t} \right) v_n(t) \tag{10}$$

between pressure and normal particle velocity, both at the surface. Equation (10) can be used to find v_{nj} on an absorbing boundary. In a particular surface cell, v_{nj} is always pointing outward. It will not be possible to use (9) to determine v_{nj} since the pressure p_{i+} is not defined outside the boundary. From (10) the average pressure over the surface can, however, be determined as a function of v_{nj} and its time derivative:

$$p_{Sj}^{[l]} = Z_0 v_{nj}^{[l]} + \frac{Z_1}{dt}(v_{nj}^{[l+0.5]} - v_{nj}^{[l-0.5]}), \tag{11}$$

where the time derivative was approximated by a finite-difference form. The normal surface particle velocity v_{nj} can be obtained as a function of surrounding pressure in the same way Eq. (9) was derived. The only difference is that p_{i+} is now replaced by p_{Sj} on the surface $S_p^{[j]}$ and that the volume average over $V_p^{[i+]}$ is replaced by an average over $S_p^{[j]}$. This results in d_j being replaced by $d_j/2$, where d_j is now twice the distance from the center of gravity of $V_p^{[i-]}$ to the plane of $S_p^{[j]}$. The local mesh condition now reads: The projection in the plane of $S_p^{[j]}$ of the distance between the center of gravity of the volume $V_p^{[i-]}$ and the

center of gravity of $S_p^{[j]}$ must be zero. Equation (11) can be used to obtain p_{Sj}. One problem remains to be solved. In (11) the normal particle velocity at time $t=l\,dt$ is needed, while v_{nj} is only determined at time $t=(l\pm 0.5)dt$. This problem is simply solved by linear interpolation in time. Finally, after some straightforward equation manipulation, the boundary impedance FDTD equation is found:

$$\left(1+\frac{Z_0}{Z_{\mathrm{FDTD}}}+\frac{2}{dt}\frac{Z_1}{Z_{\mathrm{FDTD}}}\right)v_{nj}^{[l+0.5]} =\left(1-\frac{Z_0}{Z_{\mathrm{FDTD}}}+\frac{2}{dt}\frac{Z_1}{Z_{\mathrm{FDTD}}}\right)v_{nj}^{[l-0.5]}+\frac{2\,dt}{\rho_0 d_j}p_{i-}^{[l]}, \qquad (12)$$

$$Z_{\mathrm{FDTD}}=\frac{\rho_0 d_j}{dt}.$$

Remark that the subset (10) of boundary impedance leads to a boundary FDTD equation that only takes into account nearest neighbor pressures. Including higher-order time derivatives or requiring higher-order accuracy in d_j leads to FDTD equations that include next-nearest neighbors. Such formulas therefore require a substantial increase in numerical effort and storage and are for this reason excluded from this paper.

C. Stability

As for FDTD simulation in a Cartesian grid there is a maximum size of the simulation time step dt that can be allowed to get stable simulation results. To find this maximum allowed time step dt for the generalized grid introduced here, different techniques can be adapted. The method used here is based on the equivalence between FDTD cells and digital filter components. Indeed, the relations (5) and (9) can be looked at as describing infinite impulse response digital filters, which give an output based on older inputs. To find the stability condition for a single grid cell, the dependence of a discrete pressure p_0 on the pressure at the same point a time step dt ago is expressed by combining (5) and (9):

$$p_0^{[l+1]}=\left(1-\frac{c^2\,dt^2}{V_p^{[0]}}\sum_j\frac{S_p^{[j]}}{d_j}\right)p_0^{[l]} -\frac{\rho_0 c^2\,dt}{V_p^{[0]}}\sum_j S_p^{[j]}v_{nj}^{[l-0.5]} +\frac{c^2\,dt^2}{V_p^{[0]}}\sum_j\frac{S_p^{[j]}}{d_j}p_{j,i+}^{[l]}, \qquad (13)$$

where $p_{j,i+}$ refers to the pressure near p_0, at the other side of surface $S_p^{[j]}$. In deriving (13), the identity $p_{j,i-}=p_0$ for all velocities v_{nj} surrounding p_0 was used in (9). The last two terms on the right-hand side of this equation can only depend on p_0 at least a time 2 dt ago. To have local stability of this digital system the coefficient of $p_0^{[l]}$ must be in the interval $[-1,1]$ (poles of transfer function must be inside unit circle). Taking into account that the summation part of this coefficient is always positive, this yields

$$c\,dt<\left(\frac{1}{2V_p^{[0]}}\sum_j\frac{S_p^{[j]}}{d_j}\right)^{-1/2}. \qquad (14)$$

The upper limit for dt expressed in (14) must hold for all cells within the modeling region.

Analog, but rather boring, calculations can be performed to study the dependence of p_0 on the pressure at the same point two or more time steps ago. From these it can be shown that the resulting filter is also stable if (14) is satisfied. On the other hand, one cannot rely on a less strict stability condition that could result from these relations since the cell under study might be near a border of some sort.

For the particular cells limited by a boundary impedance, some care is needed to ensure stability for all possible values of the boundary impedance.

Here again in the limit of a Cartesian grid with spacing dx, dy, and dz, the known stability condition $c\,dt<(1/dx^2+1/dy^2+1/dz^2)^{-1/2}$ is found.

D. Accuracy

Equations (5) and (9) are obtained in such a manner that stability can always be guaranteed. Furthermore, they are accurate, by construction, to within the first order in the spatial discretization step and to within the second order in the time discretization step. It is known that in a Cartesian grid the spatial finite-difference approximation is also accurate to within second order in the spatial discretization step. In this section the limitations on the grid of pressure points, which make (5) and (9) accurate to within second order in the spatial grid step, are derived.

If the unknown quantities p_i and v_{nj} are chosen to be averages over the volume $V_p^{[i]}$ and the surface $S_p^{[j]}$, then Eq. (5) is an exact discretization of the corresponding equation of linear acoustics. However, in deriving Eq. (9) assumptions on $p(\mathbf{r})$ are made that introduce an error. This error needs to be studied in more detail.

The accuracy of the FDTD equation (9) will depend on typical size of the mesh cells. If coefficients of second-order terms in the approximation (6) disappear in the final equation (9), the discretization error will depend on the size of mesh cells to power 3 or higher. One can easily show that second-order terms disappear in Eq. (7) if the origin $\mathbf{o}$ for the expansion (6) is chosen in the center of gravity of $S_p^{[j]}$. In (8), all second-order derivatives of p in $\mathbf{o}$ are present. They have coefficients of the form

$$\left(\frac{1}{V_p^{[i+]}}\iiint_{V_p^{[i+]}}\chi_j\xi_j\,dV-\frac{1}{V_p^{[i-]}}\iiint_{V_p^{[i-]}}\chi_j\xi_j\,dV\right), \qquad (15)$$

where χ and ξ are either of n, r, or s. Second-order accuracy is obtained if all of these coefficients are zero. This means that all the second-order moments around the center of gravity of $S_p^{[j]}$ must be the same for $V_p^{[i+]}$ and $V_p^{[i-]}$. This requirement is quite strict since it must be met for all surfaces $S_p^{[j]}$ in the simulation region.

Only a limited set of analytical meshes with sufficient symmetry meets the requirement for second-order accu-

racy. In this paper we get around the difficult task of generating a mesh that meets the requirement for second-order accuracy by using only quasi-Cartesian grids. This means that a grid is used that does not differ a lot from a Cartesian grid or differs from it only on a limited number of places. If a cell is not exactly Cartesian, (15) will not be zero, but it can be expected to be small. Second-order accuracy is lost to some extent. Numerical accuracy tests on global results showed that as long as only a limited number of cells in the whole simulation region is involved, a substantial relaxation of the second-order accuracy requirement can be allowed.

Remark that it is generally not allowed to replace p_i by the pressure in point i. Depending on the form of cell i, accuracy can be lost. However, for output purposes p_i can be interpreted as the pressure at point i without large error. This interpretation corresponds to a shift of the detection point by less than a grid cell diameter.

E. Approximate solution to the problem of finding nearest neighbors and obtaining $V_p^{[i]}$

The simplicity of the generalized FDTD equations (5) and (9) is remarkable. In fact, the major problem is now one of computational geometry. First, a suitable grid to determine the discrete pressure has to be found, then the Voronoi cell around each pressure point has to be determined. Both problems are far from trivial. At this stage no general applicable gridder is used. Each problem studied is treated separately using standard computational tools such as Mathematica. To find the nearest neighbors, many techniques have been tried and are reported in literature.[6] As most of these techniques have the disadvantage to be relatively slow and as only quasi-Cartesian grids are used, a specialized computational technique was developed. Although relatively limited in applicability, it has the advantage of speed.

In the sets of points that will be used in this FDTD application it is known that the nearest neighbors of point i will be found close to this point, so the search is limited to a close-by subset of points. Moreover, in all the quasi-Cartesian grids considered here, there will be six dominating nearest neighbors. They are located in three nearly orthogonal directions around point i. The procedure followed to find these six nearest points is easy.

(1) For a given point i_0, find the closest point from the set of grid point.

(2) Look for a nearby point close to the opposite direction. The quantity to minimize can, for example, be $d(1+\alpha \sin\theta)$, where d is the distance, θ is the angle between the direction from point i_0 to the point considered and the search direction, and α is a parameter that states how important the direction is. It introduces some fuzziness in the search.

(3) Find a nearby point close to the plane orthogonal on the direction between the two first points that were found. "Close to" can be formulated as in step (2).

(4) Look for a nearby point in the opposite direction.

(5) Find a nearby point close to the direction orthogonal on the plane of the first four points found.

(6) Look for a nearby point in the opposite direction.

The fuzziness allowed on the search directions is a first parameter that needs to be tuned for specific grids. Once the six nearest neighbors are found, an almost rectangular beam that contains the Voronoi cell can be determined. The next step is to find out if any of the other near neighbors are near enough for the plane constructed orthogonal on the middle of the line segment from the point i_0 to this near neighbor to cut some pieces of this beam. As cutting an extremely small piece from the Voronoi cell will not influence the FDTD simulation in any significant way (but will still slow down calculations) a parameter is introduced that allows us to decide not to take some of these next-nearest neighbors into account. Tuning this parameter will also allow us to wipe out the effect of numerical errors on the grid. Once all important nearest neighbors of a point i_0 are found, the Voronoi cell can be constructed by intersecting the planes constructed orthogonal on the middle of the line segments connecting point i_0 to all these neighbors. $V_p^{[i]}$ and $S_p^{[j]}$ can be calculated using standard geometrical procedures.

F. Numerical considerations

The purpose of introducing more general grids in the FDTD approximation is to minimize the error on simulation results, while keeping the computational effort low. Using the generalized FDTD formulation is only useful if this goal is achieved, so a critical look at the use of computer resources is necessary. As far as storage is concerned, a nonuniform grid requires storage of local grid information in some way. The formulation derived above requires the storage of one additional real number per discrete particle velocity if one is not interested in the actual velocity and two real numbers if the velocity is needed during the simulation. If more than six nearest neighbors are taken into account, the same pressure grid results in the storage of more velocities. The exact amount of memory needed for this depends on the type of implementation (memory friendly or CPU-time friendly).

The CPU time per grid cell and per time step dt is nearly the same for both Cartesian and quasi-Cartesian FDTD implementations as long as only six nearest pressure neighbors are taken into account. As the grid becomes less Cartesian and more neighbors are important, the amount of CPU time required will increase and can easily double. During preprocessing, there is, however, a large CPU-time penalty in using the generalized FDTD method. The larger consumption is in finding the nearest neighbors and calculating $S_p^{[j]}$ and $V_p^{[i]}$.

In conclusion, the quasi-Cartesian FDTD method requires more memory (almost twice as much) and is rather CPU-time inefficient for short simulated times (a few hundred time steps). It is, however, not noticeably slower for long simulated times (a few thousand time steps). In acoustics these long simulated times are often used.

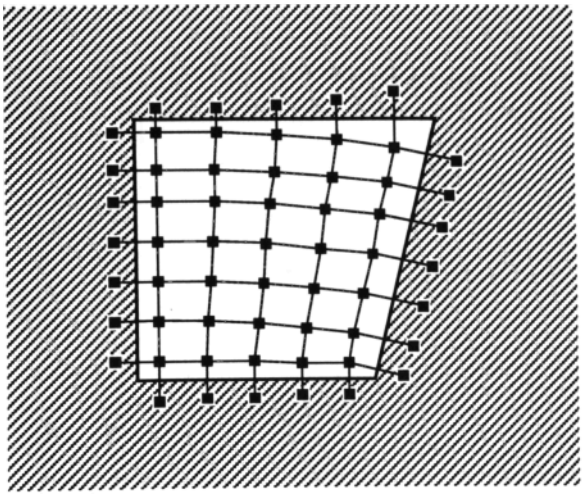


FIG. 3. Side view of the quasi-Cartesian grid used to describe a tilted boundary more accurately.

II. COMPARISON WITH CARTESIAN GRID RESULTS

A. Rectangular box with tilted boundary

The first example illustrates how the quasi-Cartesian coordinates FDTD method can be useful for calculating the acoustical fields in a cavity with plane walls that do not coincide with Cartesian coordinate planes. The cavity is 60 cm wide and 65 cm high. It has three vertical sidewalls and one tilted wall. The tilted wall makes the length of the cavity change from 50 cm, near the ground, to 70 cm, near the roof. All walls are perfectly hard except for the roof, which is covered with absorbing material. The cavity is excited by a velocity impulse at a boundary and the frequency response is obtained by Fourrier transformation of the pressure fluctuation in one of the corners. The Cartesian grids that are used as a reference have grid steps of 10, 5, and 2.5 cm, respectively. A side view of the quasi-Cartesian grid, which will be evaluated, is shown in Fig. 3. This grid is uniform with step size 10 or 5 cm in the direction orthogonal to the plane of the figure.

The three lowest-order resonance frequencies of the box obtained by Cartesian and quasi-Cartesian FDTD simulations are given as a function of CPU time needed per time step on a HP9000/705 workstation in Fig. 4. The lines in the figure show the change in resonance frequencies obtained in a Cartesian grid as the grid step is reduced and the required CPU time for a single time step increases. It is expected that accuracy increases as the grid step is refined. Turning to the quasi-Cartesian grid results (dots), it can be seen that almost the same accuracy can be obtained as with the finest Cartesian grid but that the required CPU time per time step is approximately two orders of magnitude lower.

B. Transfer function of a duct bend

It is not an easy task to find examples that allow us to study the accuracy obtained with the quasi-Cartesian FDTD simulation. Analytical solutions can only be found for structures that can easily be described by known ana-

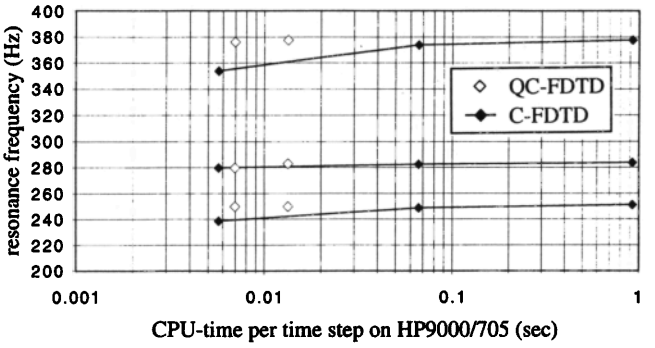
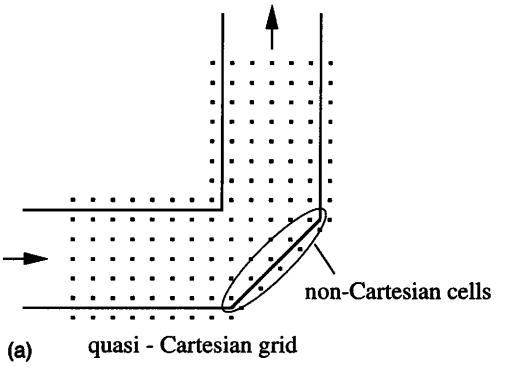


FIG. 4. Resonance frequencies of the box, calculated with different grids, as a function of CPU time per time step required in calculating them.

lytical grids. For this example the symmetry of the problem is used to reduce it to an easier problem. The duct bend studied is shown in Fig. 5(a). A 45° plate is put in a 90° corner of a square duct. It is rather difficult to describe the boundary created by this plate using a Cartesian grid. However, in this particular situation the response is the same when the structure of Fig. 5(b) is excited symmetri-



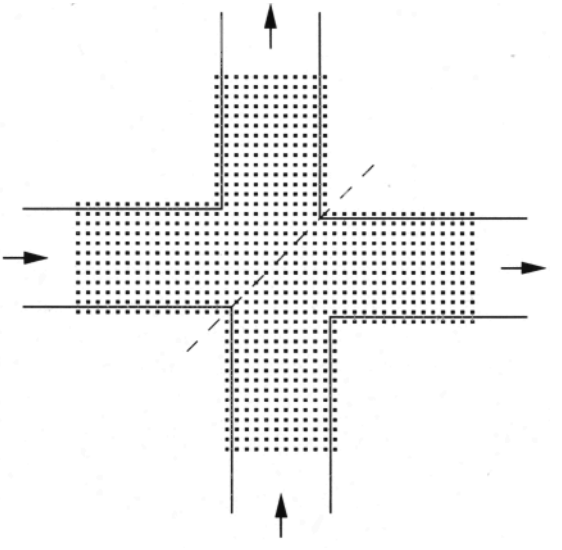


FIG. 5. (a) Quasicubic grid used to describe the duct bend; (b) reference situation based on symmetry, showing cubic grid.

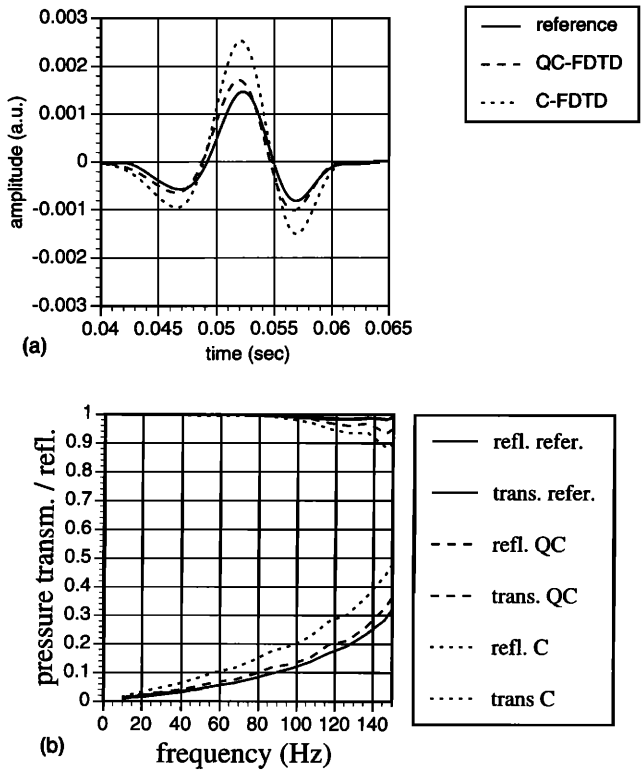


FIG. 6. Comparison of three simulations of (a) the pulse reflected at the bend; (b) spectral reflection and transmission coefficient of the bend of Fig. 5.

cally at both input ducts, because the 45° plate is a symmetry plane of the new structure. The impulse response of this structure will be used as a reference. It can be obtained quite accurately using a conventional FDTD method in a cubic grid. The grid step size is in this case 10 cm.

Two simulations are compared. First, a cubic grid is used to discretize the duct bend in Fig. 5(a) as accurate as possible. The grid step is 20 cm. Second, noncubic cells are introduced near the boundary only. They allow us to describe the 45° plate accurately and at the same time the relative low number of noncubic cells does not increase CPU requirements a lot. At the input duct a signal of the form

$$v(t)=(t-t_0)\exp[-(t-t_0)^2/\sigma] \qquad (16)$$

is injected. The pulse width σ is such that the main frequency of the signal is at 75 Hz. Results are given in Fig. 6. Figure 6(a) shows the reflected signal for the reference calculation, the simulation using a cubic grid, and the simulation with the quasi-Cartesian grid described above. The quasi-Cartesian results are much closer to the reference while the CPU requirements are almost the same as for a simulation with a cubic grid. The same conclusion can be drawn from the spectral transmission and reflection coefficients shown in Fig. 6(b).

III. CONCLUSION

In this paper a general formulation of the finite-difference time-domain equations of linear acoustics was introduced. This formulation was used to implement FDTD simulation software that allows us to describe boundaries and particular fields more accurately than conventional uniform-grid FDTD methods. As a result the numerical performance of the simulator is increased considerably in certain situations. This was illustrated with a few examples. The FDTD formulation in this paper is a step toward using FDTD in general grids that accurately describe complicated acoustical environments. The major problem that remains to be solved is a computational geometry one, namely, finding pressure point grids that give high accuracy.

ACKNOWLEDGMENT

Dr. Ir. D. Botteldooren is a senior research assistant of the Belgian National Fund for Scientific Research.

[1] For an example in electromagnetics, see A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time dependent Maxwell's equations," IEEE Trans. Microwave Theory Tech. **23**, 623–630 (1975).

[2] For an example in optics, see W. P. Huang, S. T. Chu, A. Goss, and S. K. Chaudhuri, "A scalar finite-difference time-domain approach to guided-wave optics," IEEE Photon Tech. Lett. **3**, 524–526 (1991).

[3] D. Botteldooren, "Feasibility study of active control of the noise perceived by the operator of large agricultural machines," Noise Control Eng. J. **40**, 221–229 (1993).

[4] D. T. Prescott and N. V. Shuley, "A method for incorporating different sized cells into the finite-difference time-domain analysis technique," IEEE Microwave Guided Wave Lett. **2**, 434–436 (1992).

[5] J-F. Lee, R. Palandech, and R. Mittra, "Modeling three-dimensional discontinuities in waveguides using nonorthogonal FDTD algorithm," IEEE Trans. Microwave Theory Tech. **MTT-40**, 346–352 (1992).

[6] G. T. Toussaint, ed., "Special Issue on Computational Geometry," Proc. IEEE **80** (9) (1992).