INTEGRATING FINITE DIFFERENCE SCHEMES FOR SCALAR AND VECTOR WAVE EQUATIONS

Jonathan Botts and Lauri Savioja

Aalto University
Department of Media Technology
P.O. Box 15400, FI-00076 Aalto, Finland

ABSTRACT

Room acoustic simulation is the process of generating approximate solutions to either the linearized Euler equations or the scalar wave equation. As for the continuous equations, the discrete approximations of both are equivalent. The vector formulation is less efficient, but it can inform several unexploited features of the scalar formulation. This paper first demonstrates the equivalence of the two schemes and explores how the vector formulation may be integrated into the more efficient scalar formulation to produce local velocity estimates and velocity sources on the pressure grid.

Index Terms— Finite difference methods, computer simulation, acoustic propagation

1. INTRODUCTION

Linear acoustics can be modeled as a system of first-order equations involving the acoustic pressure and vector particle velocity, or it can be equivalently modeled as a single, second-order equation for the scalar pressure or velocity potential. The numerical methods developed for both have been used in the recent literature [1, 2, 3], but the scalar formulation is simpler to implement and more efficient.

Electric and magnetic fields are both vector fields, so a scalar formulation in electromagnetics requires either updating three field components separately, losing efficiency, or discarding components to maintain efficiency while losing information. In acoustics, pressure is a scalar, so only one field must be updated, and the solution is identical to that of the vector formulation. Furthermore, any apparent effects that stem from utilization of the velocity field may also be reproduced with the pressure-only, scalar formulation. The purpose of this work is to demonstrate the equivalence of the two schemes, show that the vector formulation is unnecessarily expensive for linear acoustics, and show how effective velocity fields may be locally computed to implement sources and boundary conditions.

1.1. Relation to prior work

Previous work on finite difference methods in acoustics has focused primarily on using one formulation without discussing the equivalence or relation of one to the other. Hybridizing the two was suggested in electromagnetics [4], but the acoustic case is distinct because there is no loss of information in the scalar formulation. The contribution of this work includes proof and articulation of the equivalence in the acoustic case, calculation of velocity, and implementation of velocity sources on the pressure grid.

2. BACKGROUND

Finite difference time-domain (FDTD) methods can provide a solution to acoustic wave equations in geometries and with boundary conditions that resist analytical solution. In its first introduction to room acoustics, the Yee algorithm [5] was adapted to the solve the conservation of mass and momentum equations of linear acoustics [6]:

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \nabla p = 0, \tag{1}$$

$$\frac{\partial p}{\partial t} + \rho_0 c^2 \nabla \cdot \mathbf{v} = 0. \tag{2}$$

Although many other methods have been introduced since, the Yee algorithm is still reported in recent literature [2]. The second-order wave equation governs only one scalar variable, most often pressure, but it is also satisfied by the velocity potential, for example. Describing the acoustic pressure, p, the scalar wave equation is

$$\frac{\partial^2 p}{\partial t^2} - c^2 \nabla^2 p = 0. \tag{3}$$

To solve these equations numerically, the solution is typically calculated on a uniformly-discretized grid for simplicity, accuracy, and efficiency, with grid spacing Δx. The pressure grid function will be denoted $p(j\Delta x, k\Delta x, l\Delta x, n\Delta t) \equiv p^n_{j,k,l}$, where Δt is the temporal discretization period. Spatial indices will be denoted by subscripts, and temporal indices will be denoted by superscripts. Difference operators, δ, may be compactly written such that

$$\delta_x^2\, p^n_{j,k,l} = p^n_{j+1,k,l} - p^n_{j,k,l} + p^n_{j-1,k,l}. \tag{4}$$

The superscript on the operator indicates the order of the differencing, and the subscript indicates the variable being numerically differentiated, so the example in Equation (4) is a second difference in the x-dimension. The notation for forward and backward (first) differences will be denoted by an additional $+/-$, respectively. For example, $\delta_{y+}\, p^n_{j,k,l} = p^n_{j,k+1,l} - p^n_{j,k,l}$ and $\delta_{y-}\, p^n_{j,k,l} = p^n_{j,k,l} - p^n_{j,k-1,l}$. For notational clarity, let the Cartesian components of the vector-valued particle velocity, $\mathbf{v}$, be u, v, w. Then, the discrete approximations of Equations (1) and (2) in time-stepping form are [6]

$$u^{n+1/2}_{j+1/2,k,l} = u^{n-1/2}_{j+1/2,k,l} - \frac{\Delta t}{\rho_0 \Delta x}\delta_{x+} p^n_{j,k,l}, \tag{5}$$

$$v^{n+1/2}_{j,k+1/2,l} = v^{n-1/2}_{j,k+1/2,l} - \frac{\Delta t}{\rho_0 \Delta x}\delta_{y+} p^n_{j,k,l}, \tag{6}$$

$$w^{n+1/2}_{j,k,l+1/2} = w^{n-1/2}_{j,k,l+1/2} - \frac{\Delta t}{\rho_0 \Delta x}\delta_{z+} p^n_{j,k,l}, \tag{7}$$

$$p^{n+1}_{j,k,l} = p^n_{j,k,l} - \frac{\rho_0 c^2 \Delta t}{\Delta x}\Big(\delta_{x-} u^{n+1/2}_{j+1/2,k,l} + \delta_{y-} v^{n+1/2}_{j,k+1/2,l} + \delta_{z-} w^{n+1/2}_{j,k,l+1/2}\Big). \tag{8}$$

This is known as the Yee algorithm after the analogous scheme in electromagnetics [5]. The canonical finite difference scheme for the scalar wave equation (3)—below referred to as the *standard rectilinear scheme* (SRS)—uses centered, second differences in space and time on a single pressure grid:

$$p^{n+1}_{j,k,l} = 2p^n_{j,k,l} - p^{n-1}_{j,k,l} + \lambda^2\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) p^n_{j,k,l}\,, \tag{9}$$

where $\lambda = c\Delta t/\Delta x$ defines numerical characteristics which must contain the analytical characteristics for stability [7]. This scheme was proposed long before the advent of computers [7] and much later in room acoustics [8]. The solution in both cases converges to the true solution proportionally to Δx^2 and Δt^2. The Yee scheme maintains its second-order accuracy with only first-order differences by staggering the pressure and velocity grids. The staggered grid also avoids the necessity of inverting a linear system for each update. Although, the Yee algorithm cleverly eludes these potential problems, its solutions are identical to those of the more straightforward scalar wave equation solvers. The following section demonstrates the equivalence of the two schemes.

3. EQUIVALENCE AND COMPARISON

The equivalence of the two schemes could be motivated by the recovery of the acoustic wave equation from the Euler equations. To do so requires that the order of mixed partial derivatives may be exchanged, and this is also true in the discrete case. However, this is a relatively weak assumption because the same condition must hold in order for the discrete approximation to be consistent in the first place [9]. Operating on Equation (8) with δ_{t-} yields

$$\delta_{t-} p^{n+1}_{j,k,l} = \delta_{t-} p^n_{j,k,l} - \frac{\rho_0 c^2 \Delta t}{\Delta x}\Big(\delta_{t-}\delta_{x-} u^{n+1/2}_{j+1/2,k,l} + \delta_{t-}\delta_{y-} v^{n+1/2}_{j,k+1/2,l} + \delta_{t-}\delta_{z-} w^{n+1/2}_{j,k,l+1/2}\Big). \tag{10}$$

Similarly, operating on Equations (5), (6), and (7) with δ_{x-}, δ_{y-}, and δ_{z-} respectively results in

$$\delta_{x-}\delta_{t-} u^{n+1/2}_{j+1/2,k,l} = -\frac{\Delta t}{\rho_0 \Delta x}\delta_{x-}\delta_{x+}\, p^n_{j,k,l}, \tag{11}$$

$$\delta_{y-}\delta_{t-} v^{n+1/2}_{j,k+1/2,l} = -\frac{\Delta t}{\rho_0 \Delta x}\delta_{y-}\delta_{y+}\, p^n_{j,k,l}, \tag{12}$$

$$\delta_{z-}\delta_{t-} w^{n+1/2}_{j,k,l+1/2} = -\frac{\Delta t}{\rho_0 \Delta x}\delta_{z-}\delta_{z+}\, p^n_{j,k,l}. \tag{13}$$

Now, if all grid functions are continuous to second-order, the sequence of mixed difference operators may be reversed. Then, the velocity components may be eliminated, leaving the standard rectilinear scheme:

$$\delta_t^2 p^n_{j,k,l} = \lambda^2\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) p^n_{j,k,l}\,. \tag{14}$$

The analogous continuous result may be derived identically by replacing the difference operators on the Yee update equations with differential operators on the linearized Euler equations. The truncation error can also be carried through the calculations to truly show that the schemes are identical; however, it is left out here for brevity and clarity of presentation. It is also possible to observe that the continuous equations and discrete approximations for both sets of equations are equivalent, so the approximation error must also be equivalent—all again relying on the assumption of sufficient smoothness.

3.1. Computational Requirements

Acoustic fields are inherently simpler than electromagnetic fields, so the same results regarding computational effort [4] do not apply. The computational requirements for the acoustic case are summarized in Table 1. The first two rows indicate the number of multiplications and additions required to advance the pressure at a node one full time step. Since the Yee algorithm only uses first differences, it is possible for values at each node to be overwritten with updated values. For the scalar wave equation second temporal differences are necessary, forcing storage of at least one intermediate value. The memory storage requirements are indicated in Table 1 by *values per node.* The Yee algorithm requires storage of more velocity components as dimension is increased, but the wave equation scheme requires two scalar pressure values per node, regardless of dimension. For this reason, a three-dimensional wave equation simulation should require half as much memory as a Yee simulation.

Table 1. Computational requirements for the Yee and standard rectilinear methods. Values per node are the number of values that must be stored in memory at each grid point in the mesh.

	Yee 1D	SRS 1D	Yee 3D	SRS 3D
Multiplications	2	2	4	2
Additions	4	3	12	7
Values per Node	2	2	4	2

The standard rectilinear scheme not only requires less memory but also fewer operations. In three dimensions the operation count for Yee is nearly doubled. Given that the results are identical, the extra computation and code complexity required by the Yee scheme makes it nearly always less desirable.

These results have only been presented for the basic forms of each scheme. Higher-order [9] or interpolated approximations [10, 1, 11] can be generated for both types of scheme, which by allowing coarser discretization, reduce the overall computational load. However, the properties described above would still apply to analogous interpolated variants. In particular, relative memory requirements will remain the same, even if the overall memory load is reduced.

3.2. Boundary Conditions

The construction of general, frequency-dependent boundary conditions, an important aspect of any room acoustic finite difference simulation, is analogous in both cases as well. The Yee formulation requires a convolution of the impedance with velocity, and the formulation for the standard rectilinear scheme requires a convolution with the pressure gradient [1]. In both cases, the update equation depends on an unknown point, lying outside the domain. The boundary condition is applied so that there are as many equations as unknowns, and the system of equations has a unique solution.

One small difference is that in the Yee scheme, pressure and velocity are not collocated on the boundary, so one variable must be interpolated to apply the boundary condition. In previous work [6], a one-sided difference is used to achieve this, reducing the accuracy of the boundary approximation to first-order. However, one could in principle use centered approximations and a ghost point [1] which is eliminated by the boundary condition, maintaining second-order accuracy. The order of accuracy is not necessarily tied to the scheme, and given comparable implementations, the result should not be substantially different.

4. INTEGRATING VECTOR AND SCALAR UPDATES

This section demonstrates several ways that the equivalence of the two schemes may be exploited in practice. First, to motivate the subsequent applications, a local estimate of velocity on the pressure grid is developed. Then, the two schemes are implemented side-by-side on a single domain to further reinforce equivalence and show that the Yee scheme could be applied locally. Then, the Yee update is used to generate an effective velocity source on the pressure grid. The purpose of this section and this work is to show that anything that can be done with the vector update can be done equivalently and nearly always more efficiently with the scalar update.

The essential conversion is usually a substitution of the history of pressure gradient for velocity. Integrating both sides of (1), the velocity at a point $\hat{x} = \hat{\jmath}\Delta x$ is

$$u(\hat{x}, t) = \int_{-\infty}^{t} \frac{\partial u}{\partial \tau}\, d\tau \bigg|_{\hat{x}} = -\frac{1}{\rho_0} \int_{-\infty}^{t} \frac{\partial p}{\partial x}\, d\tau \bigg|_{\hat{x}}. \qquad (15)$$

Approximating the integral with a sum and the spatial derivative with a centered difference, the velocity at grid point $(\hat{\jmath}, \hat{n})$ is

$$u_{\hat{\jmath}}^{\hat{n}} = -\frac{\Delta t}{\rho_0 2\Delta x} \sum_{n=-\infty}^{\hat{n}} \left(p_{\hat{\jmath}+1}^{n} - p_{\hat{\jmath}-1}^{n}\right). \qquad (16)$$

If the pressure gradient terms at time step $\hat{n}$ are collected, it may be written recursively, identical in form to Equation (5):

$$u_{\hat{\jmath}}^{\hat{n}} = u_{\hat{\jmath}}^{\hat{n}-1} - \frac{\Delta t}{\rho_0 2\Delta x} \left(p_{\hat{\jmath}+1}^{\hat{n}} - p_{\hat{\jmath}-1}^{\hat{n}}\right). \qquad (17)$$

In other words, the velocity, whether calculated on the pressure grid or a staggered grid, is nothing more than a cumulative sum of the pressure gradient. This relationship may be used to update part of the domain with the vector update, compute the velocity locally in a pressure simulation, or realize velocity sources in the scalar pressure simulation.

4.1. Combined scalar-vector simulation

To show that the two simulations may be seamlessly integrated, a simple three-dimensional simulation is constructed where the two updates are applied on either side of an interface. The pressure grid for both is kept constant, and velocity nodes for the Yee portion of the grid are staggered at half-integer steps. Figure 1 shows three snapshots of two-dimensional slices of the simulation. The domain, $x, y, z \in [0, 1]$ is uniformly discretized with 101 points in each dimension, and the interface is located at $x = 0.5$. Homogeneous Neumann conditions are assigned at the boundaries, and a transient source is introduced at $(x, y, z) = (1/3, 3/8, 1/2)$, which lasts 40 time steps. The Courant stability factor set to $1/\sqrt{3}$, and snapshots are taken at time steps 60, 90, and 150. The pulse passes continuously through the interface without reflection, reinforcing that both the scheme and truncation error are equivalent.

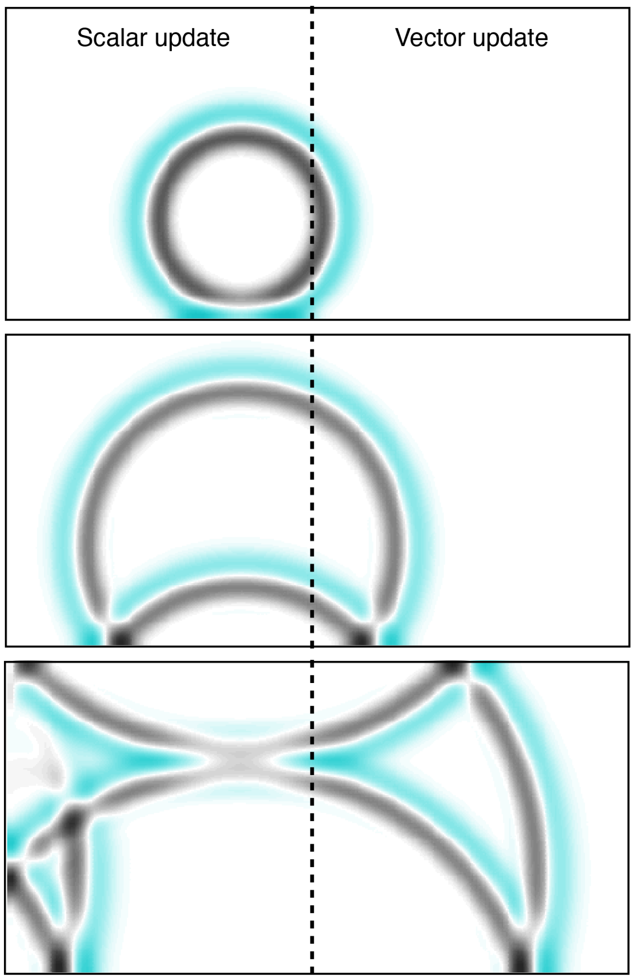

Fig. 1. Snapshots of two-dimensional slices from a three-dimensional wave simulation using both scalar wave equation and Yee updates. The dashed line indicates the interface between the two schemes.

4.2. Velocity Sources

Informed by the Yee update or a local velocity estimate, effective velocity sources may also be realized on the pressure grid. To demonstrate, let the excitation functions at two adjacent pressure nodes be $p(j,k,l) = s(n\Delta t) \equiv s^n$, $p(j+1,k,l) = -s^n$. Using the staggered grid update, (5), this is equivalent to a velocity source

$$u^{\hat{n}+1/2}_{j+1/2,k,l} = -\frac{\Delta t}{\rho_0 \Delta x} \sum_{n=0}^{\hat{n}} s^n. \tag{18}$$

It is also possible to invert this relation to find the pressures associated with the velocity source function. Essentially, the velocity source should be proportional to the integral of the dipole pressure source function, or the pressure source function should be proportional to the derivative of the velocity source function. Since both schemes may be seamlessly integrated and their updates exchanged, specifying adjacent pressure sources induces a velocity source through the implied Yee update.

To demonstrate velocity sources in the scalar wave equation scheme, three-dimensional Yee and SRS simulations are repeated with a velocity source in the center of a cubic room. The pressure dipole source function is $s^n = (t_n - m\Delta t/2)\sin^6(kct_n)$, where $m = 40$ is the length of the pulse in samples and $k = \pi/(mc\Delta t)$. The velocity source is then given by (18). The speed of sound, c, is normalized to 1 m/s, and there are 81 nodes in each of the coordinate directions. Each simulation is run for 70 time steps, and one-dimensional slices, through the source location, of the three-dimensional solutions are shown in Figure 2. The slight deviation is due to quadrature error in (18).

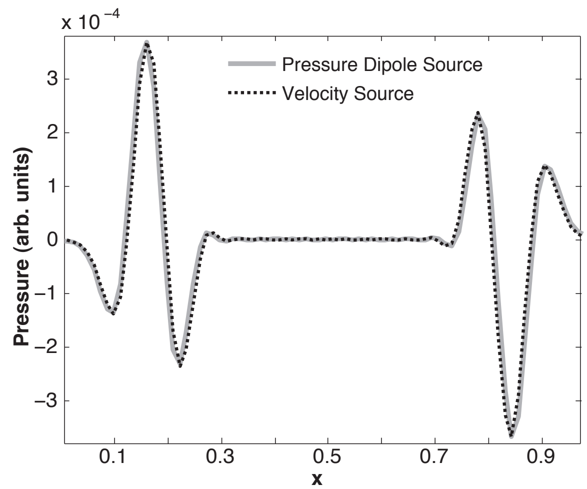

Fig. 2. One-dimensional slices of three-dimensional solutions for a pressure dipole and velocity source to show that velocity sources can be accurately realized on a pressure grid.

5. CONCLUSIONS

Both vector and scalar schemes for solving the linear acoustic wave equation result in identical solutions, but the scalar formulation is in general more efficient. To show that the vector formulation is redundant for room acoustic simulation, a local estimate of velocity and velocity sources on the pressure grid, are developed. Although both schemes are still used [1, 2], the methods based on the scalar wave equation are simpler to implement, require less memory, require fewer operations per update, and most importantly, produce identical solutions.

6. REFERENCES

[1] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-D compact explicit FDTD schemes," *IEEE Trans. Audio Speech Language Process.*, vol. 19, no. 1, pp. 34–46, 2011.

[2] H. Jeong and Y. W. Lam, "Source implementation to eliminate low-frequency artifacts in finite difference time domain room acoustic simulation," *J. Acoust. Soc. Am.*, vol. 131, no. 1, pp. 258–268, 2012.

[3] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Trans. Audio Speech Language Process.*, vol. 20, no. 5, pp. 1421–1448, 2012.

[4] P.H. Aoyagi, J.-F. Lee, and R. Mittra, "A hybrid Yee algorithm/scalar-wave equation approach," *IEEE Trans. Microw. Theory Tech.*, vol. 41, no. 9, pp. 1593 –1600, 1993.

[5] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302 – 307, 1966.

[6] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *J. Acoust. Soc. Am.*, vol. 98, no. 6, pp. 3302 – 3308, 1995.

[7] R Courant, K Friedrichs, and H Lewy, "On the partial difference equations of mathematical physics," *IBM Journal of Research and Development*, vol. 11, no. 2, pp. 215–234, 1967.

[8] L. Savioja, T. Rinne, and T. Takala, "Simulation of room acoustics with a 3-D finite difference mesh," in *Proc. Int. Computer Music Conf.*, Aarhus, Denmark, 1994, pp. 463–466.

[9] B. Gustafsson, *High order difference methods for time dependent PDE*, Springer Verlag, 2008.

[10] L. Savioja and V. Välimäki, "Interpolated rectangular 3-D digital waveguide mesh algorithms with frequency warping," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 783–790, Nov. 2003.

[11] S. Bilbao, "Optimized FDTD schemes for 3-D acoustic wave propagation," *IEEE Trans. Audio Speech Language Process.*, vol. 20, no. 5, pp. 1658–1663, 2012.