46th International Scientific Colloquium
Ilmenau Technical University
September 24–27, 2001

S. Spors and R. Rabenstein

Virtual Acoustics by Numerical Simulation

1. INTRODUCTION

New emerging multimedia standards, like the MPEG-4 standard allow the creation of virtual or synthetic acoustical environments. To auralize an environment from its geometric description, simulation algorithms for acoustic wave propagation are required. The different methods used for computational modeling of room acoustics can be divided into three groups [1]: *statistical models*, *ray-based models* and *wave-based models*. *Statistical models* try to model the statistical properties of the sound intensity and are therefore not useful in our context. *Ray-based models* suppose that the sound behaves like optical rays. As a result the effects caused by the wave nature of sound, like diffraction, cannot be handled by these methods. *Wave-based* methods try to find numerical simulations for the wave equation. Because they use distributed parameter models, they are able to handle all relevant physical effects, namely wave propagation, reflection, transmission and diffraction.

To meet the high simulation quality requirements of the proposed application, *wave-based* methods are the only ones suitable here. Among various other methods developed in the last two decades, we present here a direct method to computational acoustics, which leads from the partial differential equations to a state space description of the simulation algorithm.

2. SIMULATION ALGORITHM

The propagation of sound waves in air is governed by the equation of motion and the equation of continuity for the acoustic pressure $p(\mathbf{x},t)$ and the acoustic fluid velocity vector $\mathbf{v}(\mathbf{x},t)$ [1],

$$\rho_0 \frac{\partial}{\partial t}\mathbf{v}(\mathbf{x},t) + \text{grad}\, p(\mathbf{x},t) = \mathbf{e}_s(\mathbf{x},t) \tag{1a}$$

$$\frac{1}{\rho_0 c^2}\, \frac{\partial}{\partial t} p(\mathbf{x},t) + \text{div}\, \mathbf{v}(\mathbf{x},t) = j_s(\mathbf{x},t) \tag{1b}$$

where t denotes time and $\mathbf{x}$ the vector of space coordinates x, y, z. ρ_0 is the static density of the air and c is the speed of the sound. $\mathbf{e}_s$ and j_s are appropriate source terms. These two physical principles form a set of two partial differential equations (PDEs) describing the propagation of sound waves. For our purposes a symmetric form of these equations is advantageous. This is archived by introduction of the normalization constant $r_0 = \sqrt{3}\,\rho_0 c$ and combining (1) into one matrix equation

$$\underbrace{\begin{bmatrix} \rho_0 D_t & 0 & 0 & r_0 D_x \\ 0 & \rho_0 D_t & 0 & r_0 D_y \\ 0 & 0 & \rho_0 D_t & r_0 D_z \\ r_0 D_x & r_0 D_y & r_0 D_z & 3\rho_0 D_t \end{bmatrix}}_{\mathbf{Z}} \underbrace{\begin{bmatrix} i_1(\mathbf{x},t) \\ i_2(\mathbf{x},t) \\ i_3(\mathbf{x},t) \\ i_4(\mathbf{x},t) \end{bmatrix}}_{\mathbf{i}(\mathbf{x},t)} = \underbrace{\begin{bmatrix} e_1(\mathbf{x},t) \\ e_2(\mathbf{x},t) \\ e_3(\mathbf{x},t) \\ e_4(\mathbf{x},t) \end{bmatrix}}_{\mathbf{e}(\mathbf{x},t)} \tag{2}$$

where the operators D_t, D_x, D_y, D_z denote partial derivation with respect to time and to the components x, y, z of $\mathbf{x}$. The components of $\mathbf{v}$ are denoted by i_κ, $\kappa = 1 \ldots 3$ and $i_4 = p/r_0$. Similarly,

the components of $\mathbf{e}_s$ are denoted by e_κ, $\kappa = 1 \ldots 3$ and $e_4 = r_0 j_s$. This vector PDE is the starting point for the derivation of our simulation algorithm. It is essentially based on the multidimensional wave digital principle. However, a more direct access is given in [2], based on a four-dimensional discrete-time and discrete-space state space description.
The derivation of this discrete system according to the state space approach starts from the normalized vector PDE (2). After a series of intermediate steps, the state space representation of a discrete-time and discrete-space algorithm is obtained. These steps are

- *Separation into spatial components*
  The PDE description (2) is separated into three different spatial components. Each component contains derivatives with respect to time and only one of the spatial directions x, y, or z.

- *Numerical solution for each component*
  A numerical integration is carried out for each of the spatial components. The discretization is performed by the trapezoidal rule in two dimensions (one time and one space dimension).

- *Combination of the spatial components*
  The discrete-time, discrete-space approximations for each of the three spatial components are combined into a full four-dimensional representation (one time and three space dimensions) of the PDE description (2).

- *State space formulation*
  A suitable choice of internal states allows to formulate the discrete model in the state space context.

The following subsections cover these steps in detail.

2.1. Separation into spatial components

Inspection of (2) shows, that this PDE for three spatial components can be broken down into three PDEs with only one spatial component each. For example, the PDE for the x-component has the form

$$\bar{\mathbf{Z}}_1 \, \bar{\mathbf{i}}_1 = \bar{\mathbf{u}}_1 \, . \tag{3}$$

with

$$\bar{\mathbf{Z}}_1 = \begin{bmatrix} \rho_0 D_t & r_0 D_x \\ r_0 D_x & \rho_0 D_t \end{bmatrix}, \quad \bar{\mathbf{i}}_1 = \begin{bmatrix} i_1 \\ i_4 \end{bmatrix}, \quad \bar{\mathbf{u}}_1 = \begin{bmatrix} u_{11} \\ u_{14} \end{bmatrix}. \tag{4}$$

The matrix $\bar{\mathbf{Z}}_1$ is obtained from $\mathbf{Z}$ in (2) by elimination of the second and the third row and column, which contain only y and z components. Furthermore, the element $3\rho_0 D_t$ in $\mathbf{Z}$ contributes equally to all spatial components and is represented in $\bar{\mathbf{Z}}_1$ by one third of its value. Similarly, the first element in $\bar{\mathbf{u}}_1$ is equal to the first element in $\mathbf{e}$. The sum of u_{14} and the corresponding elements for the other spatial components is equal to the fourth element in $\mathbf{e}$.

2.2. Numerical solution for each component

The numerical solution of the separate spatial components follows through a series of steps, which are shown in Fig. 1. The procedure is explained for the spatial direction x. The starting point is

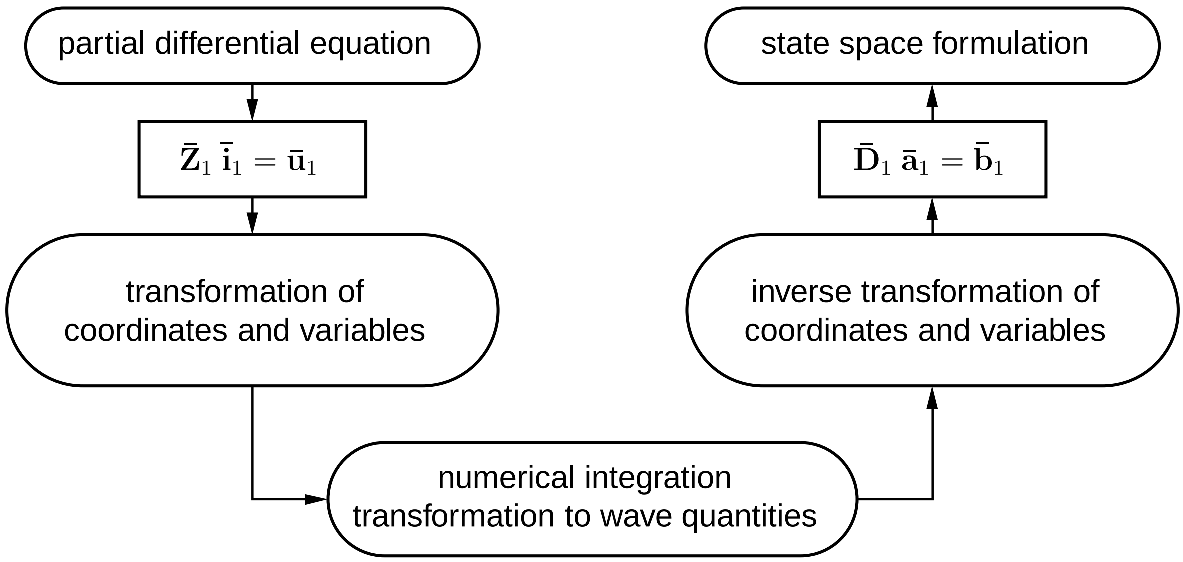


Figure 1: Transformation of the partial differential operator for one spatial component into a state space formulation

the partial differential operator $\bar{\mathbf{Z}}_1$ from (4). Since it involves both time and space differentiation, it is not very suitable for direct numerical integration. A decoupled form with simple differentiation operators would be more desirable. This is achieved by two measures: First, a variable transformation, which decouples the differentiation operators, or in other words, a diagonalization of the operator matrix $\bar{\mathbf{Z}}_1$. Second, a coordinate transformation, such that each entry in the diagonal operator matrix contains only differentiation with respect to a single coordinate. The resulting decoupled form allows a numerical integration by the trapezoidal rule. Care has to be taken to avoid delay free loops, which would call for an iterative solution. This problem is circumvented by another transformation of the variables. It leads to the so called wave quantities $\bar{\mathbf{a}}_1$ and $\bar{\mathbf{b}}_1$ [3]. Now, the spatial components can be integrated numerically, though in transformed coordinates and variables. Therefore, the decoupling steps have to be reversed, to arrive at a discrete-time, discrete space wave quantity formulation of the original problem. A more detailed description of the coordinate and variable transformations shown in Fig. 1 is given in [2]. The matrix $\bar{\mathbf{D}}_1$ in Fig. 1 contains shift operators in x-direction and delay operators in time direction. As a difference operator matrix it corresponds to the differential operator matrix $\bar{\mathbf{Z}}_1$.

2.3. State space formulation

The final form of the algorithm can be formulated in terms of a state space description. The state $\mathbf{z}$ is associated with one of the wave quantities (see [2] for details). The state space model consists of the state equation and the output equation

$$\mathbf{z} = \mathcal{D}\left[\mathcal{A}\mathbf{z} + \mathcal{B}\mathbf{e}\right], \qquad (5a)$$
$$\mathbf{i} = \mathcal{C}\mathbf{z} + \mathcal{F}\mathbf{e}, \qquad (5b)$$

The state equation (5a) follows from condensing the procedure outlined in Fig. 1 into one matrix equation. The output equation (5b) represents the conversion of the wave quantities back to acoustic variables pressure and velocity. The concise matrix formulation of the discrete model as a state space description allows the direct implementation of (5) in a software algorithm.

2.4. Boundary conditions

The operator matrix $\mathcal{D}$ in the state equation (5a) contain shifts in both directions of each spatial dimension. This requires the knowledge of the previous states in all adjacent points. However, if a point lies at the boundary of the spatial domain, e.g. at the wall of an enclosure, then one or more of the adjacent points are beyond the boundary, where the PDE is no more valid. In this case, the state of these points has to be determined from boundary conditions rather than from the PDEs (1). (See also [4].) A detailed presentation of the incorporation of various types of boundary conditions is beyond the scope of this paper. Only a short outline of the general approach is given here.

The idea is to split the state vector $\mathbf{z}$ into two components: the interior states $\mathbf{z}_i$ and the boundary states $\mathbf{z}_b$. The interior states follow from a state equation similar to (5a). The boundary states follow from the interior states and the boundary conditions. The state space representation has to consider both types of states appropriately. Its general form is given by

$$\begin{aligned} \mathbf{z}_i &= \left(\mathbf{T}_i^T \mathcal{D}\right) \left[\mathcal{A}\mathbf{z} + \mathcal{B}\mathbf{e}\right], && \text{(6a)} \\ \mathbf{z}_b &= \mathcal{A}_b \mathbf{z}_i + \mathcal{B}_b \mathbf{e}, && \text{(6b)} \\ \mathbf{z} &= \mathbf{T}_i \mathbf{z}_i + \mathbf{T}_b \mathbf{z}_b, && \text{(6c)} \\ \mathbf{i} &= \mathcal{C}\mathbf{z} + \mathcal{F}\mathbf{e}. && \text{(6d)} \end{aligned}$$

The matrices $\mathbf{T}_i$ and $\mathbf{T}_b$ contain only ones and zeros. They depend on the geometry and describe whether a state is an interior state $\mathbf{z}_i$ or a boundary state $\mathbf{z}_b$. Equation (6a) is very similar to the state equation (5a), except that it delivers only the interior states. The boundary states are computed in (6b) from the interior states and the boundary conditions, which determine $\mathcal{A}_b$ and $\mathcal{B}_b$. Both interior and boundary states are merged into the complete state vector $\mathbf{z}$ in (6c). It is used to deliver the output quantities in (6d) and to update the interior states in (6a).

3. IMPLEMENTATION

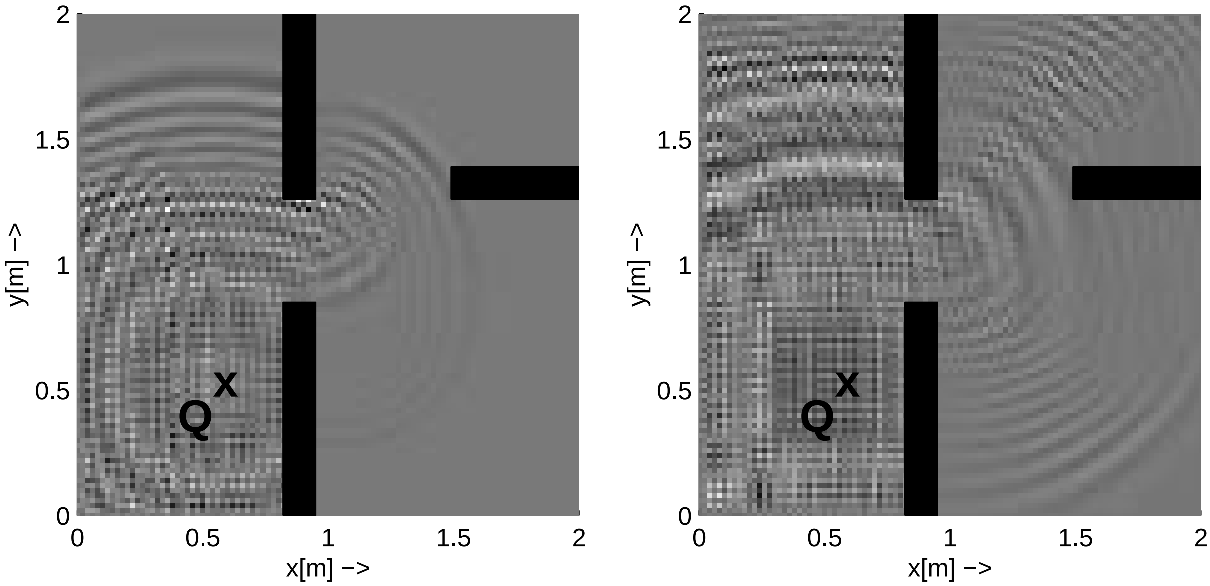
The algorithm described above has been implemented in `C++` in an object oriented fashion. This implementation is based on a multidimensional systems library which was developed at our laboratory. In the current version, objects with rectangular shape and analytical objects of 2nd degree (e.g. ellipsoids) with given surface reflexion factors can be modeled. Available sources include point sources, loudspeaker arrays and horn loudspeakers. The respective sound pressure of the wavefield can be captured at any point within the spatial grid.

4. RESULTS

This section shows some results of simulations carried out by the algorithm described in the previous sections. The first example shows an application of the algorithm to room acoustics, the second example shows the simulation of loudspeaker arrays. Animated simulations for other acoustical environments are available at [5].

4.1. Room acoustics

The experimental setup consists of a room with the size $2.0 \times 2.0 \times 1.8$ m (w $\times$ d $\times$ h), with three walls inserted inside. The ceiling and floor have a surface reflection factor of $r = 0$ and are therefore absorbing, the remaining walls have a surface reflection factor of $r = 0.7$. Figure 2 shows a snapshot of the wavefield at $t = 4.1$ ms and $t = 6.7$ ms in 90 cm height after excitation



(a) $t = 4.1$ ms

(b) $t = 6.7$ ms

Figure 2: Snapshots of the wavefield after excitation with an Gaussian impulse at the position Q.

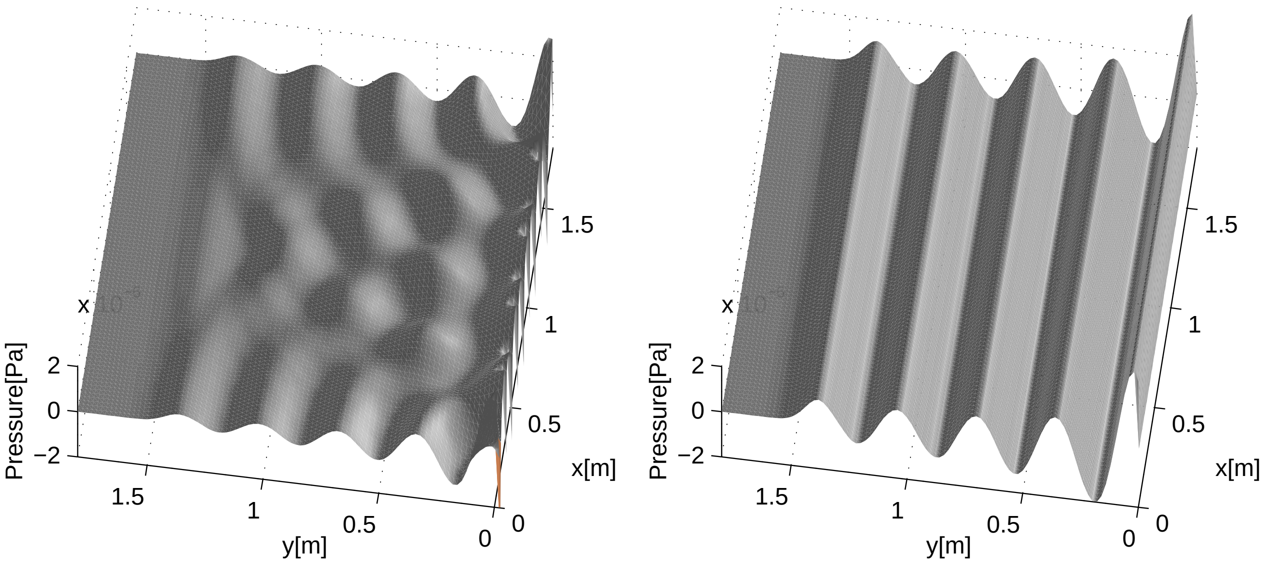
with an Gaussian impulse at $t = 0$ at the shown source position Q. The source Q is also positioned in 90 cm height. Note that the simulation was performed in three dimensions. It can be clearly seen, that the diffraction effects caused by the inner walls are handled correctly by the proposed algorithm.

4.2. Loudspeaker array simulations

The wave field of an acoustic scene can be synthesized by the concept of wave field synthesis (WFS). The theory of WFS is based on the Kirchhoff-Helmholtz integral [6]. WFS uses loudspeaker arrays to synthesize the wavefronts. This example shows spatial aliasing effects caused by an linear line array. The experimental setup consists of a room with the size $1.8 \times 1.8 \times 1.8$ m (w $\times$ d $\times$ h). All walls have a surface reflection factor of $r = 0$ and are therefore absorbing. The line array is mounted along the x-axis in 90cm height. Figure 3 shows a snapshot of the wavefields at $t = 4.4$ ms using 10 or 100 loudspeakers to synthesize a plane sinusoidal wave with frequency $f = 1000$ Hz. The spatial aliasing effects when using only 10 loudspeakers can be seen clearly in Figure 3(a) on the other hand, Figure 3(b) shows that 100 loudspeakers produce an almost perfect plane wave.

5. CONCLUSION

The results show that numerical simulation of acoustic wave propagation through the proposed algorithm is able to reproduce the physical effects of transmission, reflection and diffraction. The algorithm is therefore applicable for high quality room acoustics simulation and auralization. The drawback is a high computationally complexity which does not allow a real time implementation for higher frequencies. This problem can be overcome by combining our proposed *wave-based* algorithm for lower frequencies with an algorithm based on *ray-based models* for higher frequencies. The approximation that sound behaves like optical rays is nearly fulfilled for higher frequencies.



(a) 10 loudspeakers (b) 100 loudspeakers

Figure 3: Snapshots of the wavefield generated by two different loudspeaker arrays ($t = 4.4$ ms).

Another application shown here is the simulation of WFS systems. Advanced numerical simulation tools like the one presented here allow a physically correct design of WFS systems by simulating their properties. Further simulated scenarios and animated simulations can be found in [5].

References

[1] K.H. Kuttruff, "Sound in enclosures," in *Handbook of Acoustics*, M.J. Crocker, Ed. John Wiley and Sons, Inc., 1998.

[2] R. Rabenstein and A. Zayati, "Sound field simulation by computational acoustics. Part I: Simulation algorithm," *Int. Journal of Adaptive Control and Signal Processing*, vol. 14, pp. 663–680, 2000.

[3] A. Fettweis, "Multidimensional wave-digital principles: From filtering to numerical integration," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 94)*. IEEE, April 1994, pp. VI–173 – VI–181.

[4] H. Krauß, "Simulation of coupled transmission lines by multidimensional wave digital filters," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 96)*, 1996, pp. III–1747–III–1750.

[5] `http://www.LNT.de/˜spors/WPSIM`.

[6] A.J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *Journal of the Acoustic Society of America*, vol. 93, no. 5, pp. 2764–2778, May 1993.

Authors

Sascha Spors
Telecommunications Laboratory
University of Erlangen-Nuremberg
Cauerstrasse 7
91058 Erlangen
Phone: +49-9131-85 27105
E-Mail: spors@LNT.de

Rudolf Rabenstein
Telecommunications Laboratory
University of Erlangen-Nuremberg
Cauerstrasse 7
91058 Erlangen
Phone: +49-9131-85 28717
E-Mail: rabe@LNT.de