

Research Article

Numerical Analysis of the Wave Equation

¹Masoud Ghaemi, ²Majid Rajayi, ³Davood Sharifi and ²Morteza Souri

¹Department of Civil Engineering, Gonbad Shams University

²Department of Mechanic Engineering, Bafgh Branch, Islamic Azad University, Yazd, Iran

³Department of Mining Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

Abstract: This study presents an overview of the acoustic wave equation and the common time-domain numerical solution strategies in closed environments. First, the wave equation is presented and its qualities analyzed. Common principles of numerical approximation of derivatives are then reviewed. Based on them, the Finite Difference (FD) and the Finite Element Methods (FEM) for the solution of the wave equation are presented along with algorithmic and practical considerations.

Keywords: Finite difference method, finite element method, numerical analysis

INTRODUCTION

The sensation of sound is due to small variations in air pressure. The variations are governed by the three-dimensional wave equation, a second-order linear partial differential equation, which relates the temporal and spatial derivatives of the pressure field. This study presents an overview of the wave equation and outlines the most common time-domain methods for its numerical solution; namely the finite difference and the finite element methods.

THE WAVE EQUATION

This section presents the wave equation and some of its qualities. We first introduce the nature of the solutions, then discuss the equation of motion along with boundary and initial conditions and conclude with a note on the Helmholtz equation.

When determining the acoustic properties of an environment, we are actually interested in the “propagation of sound”, given the properties and location of a sound source. Sound waves themselves are small fluctuations in air pressure. In simple cases (in the absence of temperature gradients, for instance) these small fluctuations can be treated as small perturbations of an ambient pressure field. The propagation of these fluctuations is governed by the wave equation, which can be derived from purely mechanical considerations or from suitable simplifications of the more general equations of fluid dynamics (Pierce, 1989)? Solutions of the complete, non-linear equations of fluid dynamics are generally not required for acoustic purposes.

The solution of the wave equation is a time-dependent pressure field $u(t, x)$, with $x \in \Omega$ and $t > 0$.

Here Ω denotes the set of points inside the environment to be simulated; in realistic situations Ω is three-dimensional, but we shall often resort to lower dimensional examples for easier presentation. We stress that the solution u to the equations is a scalar function over three spatial dimensions and time; the function describes the acoustic sound pressure for each point x in the environment for each t . This is the key difference between solutions of “normal” algebraic equations and differential equations; roughly speaking, the solutions of differential equations are themselves functions, while the solutions of normal algebraic equations are points within the domain of some equation-dependent function.

The equation of motion and boundary conditions:

The wave equation is a second-order linear partial differential equation:

$$u_{tt} = c^2 \Delta u + f$$

With:

$$u_{tt} = \frac{\partial^2 u}{\partial t^2}, \quad \Delta = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2)$$

Whose u is the pressure field (as described above) and c is the speed of sound, which we assume to be constant in the whole environment. The equation thus relates the second time derivative of the pressure to its spatial Laplacian Δu . $f = f(t, x)$ represents time dependent force terms, which we discuss soon. The equation is called a partial differential equation because it involves derivatives of the solution function u with respect to more than one variable.

Eq. (1) is in itself not uniquely solvable. In addition to this equation of motion, the behavior of the solution on the boundaries of the environment, which we denote by $\partial\Omega$ needs to be determined. These so-called Boundary Conditions (BCs) dictate how the walls of the environment reflect sound waves. Elementary types of boundary conditions prescribe either the solutions' values or the values of the solution's normal derivatives' on the boundary. In this study we leave out most details on boundary conditions. We mentioning passing that it is possible to construct so-called absorbing BCs, which do not reflect any of the waves striking the boundary and the waves, appear to just leave the domain.

Absorbing boundary conditions are useful in analyzing enclosures partly bounded but connected to a "large" open space.

In addition to boundary conditions, initial conditions need to be specified. This means that for $t = 0$, an initial pressure distribution $u(0, x)$ and an initial velocity $u_t(0, x)$ distribution are required.

The force term $f(t, x)$ represents sources of distributions in air pressure; these are the sound sources. Usually, an acoustics application solves the wave equation with describing an initial impulse. The solution of the wave equation then describes the time dependent propagation of the impulse in the environment. The solution u is an univariate function (in t) for each x in the environment and can be used as an impulse response in a naturalization system.

Dispersion: In the case of wave propagation, dispersion means that waves either travelling to different directions or having different frequencies propagate with different speeds. Dispersion can occur both naturally (in a dispersive medium) and artificially. The "pure" wave equation presented above is non-dispersive, i.e., in the exact solution all waves, regardless of direction or frequency, propagate with the speed c . Unwanted artificial dispersion occurs in all numerical methods. The effects often include waves travelling along coordinate axes propagating slower than in diagonal directions and high-frequency waves propagating slower than lower-frequency waves. It is possible to analyze the dispersion introduced by a numerical method either directly by substituting certain "test waves" into the discretized equation by frequency domain analysis, Savioja (1999).

The helmholtz equation: We briefly mention that separating variables in the wave equation, that is, searching for the solution u in the form:

$$u = \Psi(x) e^{i\omega t} \quad (3)$$

leads to the so-called Helmholtz equation, sometimes called the reduced wave (Eq):

$$\Delta\Psi_k + k^2\Psi_k = 0 \quad (4)$$

where, ω is the frequency of an eigenmode and $k = \omega/c$ is the wave number. Mathematically, the problem is about the eigenvalues of the Laplacian operator (Eriksson *et al.*, 1996). For closed domains, solutions only exist for a countable set of different ω ; the solutions Ψ_k for the corresponding wave numbers are the standing waves inherent to the geometry of the domain. The Helmholtz equation is the basis for a large number of numerical methods for computational acoustics; they are called spectral methods since they do not simulate time-dependent pressure fields but the responses of the environment to different frequencies instead. Since the focus on this study is in explicit time-domain methods, we do not discuss this further.

FINITE DIFFERENCES

The exact solutions to the wave equation discussed in the previous section are infinite dimensional that is, no finite number of parameters can fully describe the solution, excepting a very limited set of special cases. Since computers work with finite memories and perform only finite calculations, approximations must be made in order to solve the wave equation numerically. Here we stress that the complete, correct solution is generally unavailable to us in closed form. Numerical analysis deals, among other problems, with issues concerning discrete approximations to continuous problems; these include the methods used to discretize the solutions domains in both time and space, methods of solving the discretized versions of the equations and error analysis.

Discretized derivatives: In this section we describe the simplest possible ways of discretizing derivatives of functions.

We work in one dimension for simplicity.

As an example we look at continuous, bounded, real-valued functions defined on the interval $[0, 1]$. In order to represent general functions, we might scatter a large but finite number N of equidistant points x_i , with $i = 1, \dots, N$, inside the interval and store the value of the function in those points only. Even though this representation does obviously not correspond to a continuous function (generally speaking), we do have some idea of what the function is like. We denote the distance between two node points by h . Now, suppose that we are interested in the derivatives of the function which we have represented by point samples. Since we only know the function's values at the node points x_i , we must somehow combine those values to obtain an estimate for the derivative. The simplest methods are suggested by the usual difference quotient that is used to define the derivative in the continuous case. This yields the approximations:

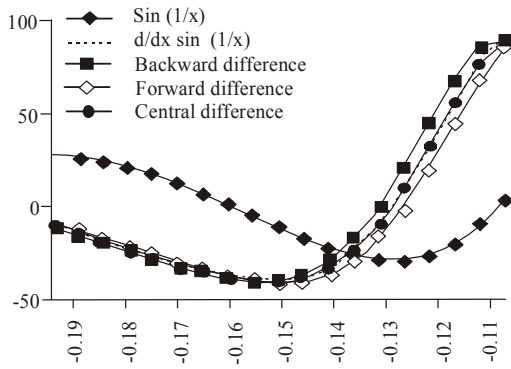


Fig. 1: Comparison of several difference approximations of $\sin(1/x)$ on the interval $v [-0.2,-0.1]$

$$f(x_t) \approx \frac{f(x_{t+1}) - f(x_t)}{h} \quad (5)$$

$$f(x_t) \approx \frac{f(x_t) - f(x_{t-1}))}{h} \text{ and} \quad (6)$$

$$f(x_t) \approx \frac{f(x_{t+1}) - f(x_{t-1}))}{2h} \quad (7)$$

which are called forward difference, backward difference and central difference, respectively (Atkinson and Han, 2009).

The Taylor series provides us an elegant approximation for the second derivative. The expansion gives us:

$$f(x+h) = f(x) + \frac{h^1}{1!} f'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + o(h^4) \quad (8)$$

$$f(x-h) = f(x) - \frac{h^1}{1!} f'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + o(h^4) \quad (9)$$

from where adding the approximations side by side and dividing through with h^2 we get:

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + o(h^2) \quad (10)$$

This is an accurate approximation; for smooth functions the error drops according to h^2 .

We also note that the central difference scheme in Eq. (7) follows from neglecting also the second order terms and subtracting the equations from each other.

Figure 1 shows a comparison of the first order difference approximations of the function $\sin(1/x)$, plotted in blue, on the interval $[-0.2, -0.1]$. The example is arbitrary, with the function and range chosen so that the function is smooth (infinitely differentiable) on the interval. The figure illustrates the derivative approximations' nature as linear combinations of

samples from the original function and that the central difference scheme is more accurate than the other two. The original function has been artificially scaled up to show its form.

Discrete differential operators: By writing values of the point samples of a function u as an N -dimensional vector u_h , the difference approximations of the last section can be written in a matrix form so that multiplication of the vector u_h of function values with the matrix produces a new vector approximating the values of the derivative (or second derivative) at the node points. The semantics have banded structure; on each row, nonzero elements are only found on the diagonal and/or its immediate neighbors. As an example, the central difference and the second derivative approximations result in the matrices:

$$D_h = \frac{1}{2h} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \\ & & & & -1 & 1 \end{bmatrix} \text{ and } \Delta_h = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 & 1 \end{bmatrix}$$

So that $D_h u_h$ is a central difference approximation to the first derivative of the function u with samples placed h units apart and $\Delta_h u_h$ is the approximation of the second derivative in a similar fashion. Here the zero elements of the matrices have been left blank. The values for the ends of the interval are dependent on the initial and boundary conditions of the differential equation at hand and some special care needs to be taken in order to get the boundary values right. We not discuss this further.

This point of view clearly demonstrates the fact that differentiation can be seen as an operator that acts on a function and produces another function, that is, the derivative of the original "input" function. When we deal with functions discretized as described earlier, the discretized differentiation operator is represented by a matrix. This is a consequence of differentiation being a linear operation in the continuous case.

Spatial discretization of the wave equation by finite differences: Here we show how finite difference approximations can be used for discretizing the wave equation. We work in one dimension, but we keep in mind that the development is essentially the same for higher dimensions.

By using the discretized representation Δ_h for the second derivative we derive a semi discrete version of the one-dimensional wave equation; by substituting u_h for u and $\Delta_h u_h$ for u_{xx} and noting that $u_{xx} = \Delta u$ in one dimension we obtain:

$$u_h'' = c^2 \Delta_h u_h + f_h \quad (11)$$

where, f_h denotes the values of the function f in the node points and primes denote differentiation with

respect to time. Remarkably, this semi discretized form of the wave equation is no longer a partial differential equation, since the spatial Laplacian has been “reduced” into a matrix multiply; it is an ordinary differential equation in N unknowns with the unknown vector u_h , whose elements are the values of the solution function u at the node points x_i . This equation can be solved with any standard method for integrating differential equations with respect to time.

These methods are discussed in the following section.

We also mention the digital waveguide methods for the solution of the wave equation.

These methods are finite difference schemes with a digital signal processing point of view. The signal processing approach has many favorable qualities; these include e.g.,

The possibility of using frequency-dependent boundary conditions implemented by digital filters. The interpolated waveguide mesh of Savioja (1999) bears some resemblance to the finite element method. Savioja (1999) for an in-depth treatment of digital wave guide methods.

Two- and three-dimensional problems: The one-dimensional difference approximations discussed in the previous sections are easily extended to two or more dimensions. For instance, the gradient operator, define $\nabla = (\frac{\partial}{\partial x} \frac{\partial}{\partial y})$ in 2D and $\nabla = (\frac{\partial}{\partial x} \frac{\partial}{\partial y} \frac{\partial}{\partial z})$ in 3D, is easily implemented with one-dimensional finite differences along each coordinate axis. In a similar fashion, the Laplacian operator-see Eq. (2)-takes the form:

$$\Delta u_{t_h g_h} \approx \frac{u_h(i+1, j) - 2u_h(i, j) + u_h(i-1, j)}{h^2} + \frac{u_h(i, j+1) - 2u_h(i, j) + u_h(i, j-1)}{h^2} + \frac{u_h(i+1, j) + u_h(i-1, j) + u_h(i, j+1) + u_h(i, j-1) - 4u_h(i, j)}{h^2} \quad (12)$$

in two dimensions, where we have assumed that the two-dimensional domain has been discretized into a regular grid of points, so that the values of the function u are now stored in a matrix (instead of a vector) u_h , so that $u_h(ih, jh) \approx u(i, j)$. For simplicity, we have assumed the same discretization parameter h for both dimensions. The three-dimensional case is analogous.

Writing the difference approximation from Eq. (12) into a matrix form (as above) presents some difficulty due to u_h now being a matrix instead of a vector as in the one dimensional case. Still, as the approximation is again a linear combination of the elements of u_h , a similar matrix representation does exist (Eirola, 2002). This is achieved by first stacking the columns of u_h into a long column vector, after which it is straight forward to derive the required matrix expressions. After this modification the semi discretized wave equation in two or more dimensions is exactly the same as Eq. (11), the one dimensional case.

TIME INTEGRATION

This section presents the necessary tools for obtaining a fully discrete solution of the wave equation by time stepping. The process is called integrating the differential equation. To this end, we first write the semi discretized equation as a first-order system of differential equations and after those present different time-stepping methods for solving the first-order system. We conclude by some remarks on stability of the numerical solutions.

We first recall the semi discretized equation:

$$\begin{cases} u_h'' = c^2 \Delta_h u_h + f_h, & x \in \Omega(\text{equation of motion}) \\ u_h(0, x) = u_h^0(x), & (\text{initial condition for } u_h) \\ u_h'(0, x) = V_h^0(x), & (\text{initial condition for } u_h') \end{cases}$$

where, u_h^0 and v_h^0 are predefined functions defined over the spatial discretization. In addition, the problem-dependent boundary conditions need to be specified. This is a second order system of ordinary differential equations in N unknowns.

Most integration methods work on first-order differential equations. This poses no problems, since all higher-order problems can be transformed into first-order ones by introducing new variables. To transform the semi discretized wave equation into a first order system, we define a new variable $v_h = \frac{du_h}{dt}$ so that Eq. (11) takes the form:

$$\begin{cases} \frac{dv_h}{dt} = c^2 \Delta_h u_h + f_h \\ \frac{du_h}{dt} = v_h \end{cases}$$

This has the effect of doubling the number of unknowns, since we are now left with *two* vectors of length N to solve for. We can further simplify the system (13) by concatenating the vectors u_h and v_h into a new vector w so that we get:

$$\frac{dw}{dt} = A w + g, \quad \text{With} \quad (14)$$

$$W = \begin{bmatrix} u_h \\ v_h \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ f_h \end{bmatrix}, \quad A = \begin{bmatrix} 0 & I \\ c^2 \Delta_h & 0 \end{bmatrix}, \quad \text{and } w(0) = \begin{bmatrix} u_h^0 \\ v_h^0 \end{bmatrix}$$

where, each element of A , printed in bold, denotes a $N \times N$ sub matrix and I denotes the identity matrix. This is the simplest possible form for a first-order, linear system of differential equations.

The numerical solution of the above system is a discrete sequence $w^k, k \in \mathbb{N}$, of vectors corresponding to values of the solution w at different timesteps. We choose to use a constant timestep δ for simplicity, so that we have $w^k \approx w(t_k)$, where $t_k = \delta k$.

Similarly, we denote $g(\delta k)$ by g^k . To make description of the integration methods in the next subsections still simpler, we make use of a more abstract formulation. In general, any first-order system of differential equations 6 can be written as:

$$w'(t) = d(t, w(t)) \tag{15}$$

Eq. (14) maps to this representation by:

$$d(t, w(t)) = Aw(t) + g(t) \tag{16}$$

This more abstract form (15) is most suitable for describing integration methods.

Explicit methods: The most obvious integration method for the system (15) is the Euler method. It follows from substituting the forward difference scheme from above section into (15), yielding:

$$\frac{w^{k+1}-w^k}{\delta} = d(t_k, w^k) \Leftrightarrow w^{k+1} = \delta d(t_k, w^k) + w^k$$

Substituting for d from (16) we have:

$$w^{k+1} = \delta(Aw^k + g^k) + w^k = (\delta A + I)w^k + \delta g^k \tag{17}$$

This method has the advantage of simplicity, but in practice it is not used much, because it is highly *unstable*. We come back to stability issues in the end of the section.

The Euler method is the simplest one in the class of methods generally referred to as explicit Runge-Kutta methods. The classical Runge-Kutta method, often referred to as the Runge-Kutta method, is one of them. All the higher-order R-K methods work by subdividing the time interval into smaller sub-time steps, achieving variable orders of accuracy. The Euler method performs worst of these methods.

Implicit methods: So-called implicit methods help overcome stability problems often associated with explicit schemes. The difference between explicit and implicit methods is best illustrated by the implicit Euler method, which follows from substituting the backward difference approximation into:

$$\frac{w^k - w^{k-1}}{\delta} = d(t_k, w^k) \Leftrightarrow w^k = \delta d(t_k, w^k) + w^{k-1}$$

and after manipulating the indices we have:

$$w^{k+1} = \delta d(t_{k+1}, w^{k+1}) + w^k \tag{18}$$

and substituting for d we finally get:

$$w^{k+1} = \delta(Aw^{k+1} + g^{k+1}) + w^k$$

$$\begin{aligned} \Leftrightarrow (I - \delta A) w^{k+1} &= w^k + \delta g^{k+1} \\ \Leftrightarrow w^{k+1} &= (I - \delta A)^{-1}(w^k + \delta g^{k+1}) \end{aligned} \tag{19}$$

The scheme works by “borrowing” the future value for w in the evaluation of the functioned. The seemingly innocent switching from forward to backward differences has yielded a significantly different difference scheme; one that requires a matrix inversion. (Note that when the timestep_ is constant, the inversion only has to be done once.) The implicit Euler method is still simple and has the virtue of unconditional stability.

The most accurate first-order implicit scheme is the Crank-Nicolson method7. It is an application of the central difference scheme and its idea is to evaluate the function d in the middle of the timestep, with respect to both time and the solution w. The scheme is defined as:

$$\frac{w^{k+1}-w^k}{h} = d\left(\frac{1}{2}(t_k + t_{k+1}), \frac{1}{2}(w^k + w^{k+1})\right)$$

Substituting d from (16) as before we have:

$$\begin{aligned} w^{k+1} &= w^k + \frac{1}{2}\delta A(w^k + w^{k+1}) + \delta g\left(t_k + \frac{\delta}{2}\right) \\ \Leftrightarrow \left(I - \frac{1}{2}\delta A\right) w^{k+1} &= \left(I + \frac{1}{2}\delta A\right) w^k + \delta g\left(t_k + \frac{\delta}{2}\right) \\ \Leftrightarrow w^{k+1} &= \left(I - \frac{1}{2}\delta A\right)^{-1} \left[\left(I + \frac{1}{2}\delta A\right) w^k + \delta g\left(t_k + \frac{\delta}{2}\right)\right] \end{aligned}$$

As is obvious, the increased accuracy over the implicit Euler method comes at the price of a more laborious timestep, since one additional matrix multiply now has to be performed.

Again, if δ is constant, the matrices have to be formed (and the other inverted) only once.

Stability: The continuous, exact solutions of the wave equation have the property of energy conservation.

That is, if the boundaries of the environment are fully reflecting, the solution oscillates infinitely, with its energy content8 staying constant, if we let $f \equiv 0$. This is an important property and numerical methods perform differently with respect to it.

If a numerical method allows the energy of the discretized solution to grow without bound as time passes, the method is called *unstable*. We only state briefly that the magnitudes of the eigenvalues of the matrices used in the explicit iteration schemes determine constraints on the maximum possible timestep size. The eigenvalues’ magnitudes have a dependence on h, so that these constraints usually dictate the maximum allowable time step for a given level of discretization. With larger time steps the solution is soon ruined by high-magnitude noise as calculation errors build up in an uncontrollable fashion. Implicit methods are unconditionally stable. Eirola (2002) for a treatment in practice, the time step

restrictions imposed on explicit methods are so stringent that the additional computation per time step required by implicit methods is outweighed by the gains from using larger time steps.

THE FINITE ELEMENT METHOD

The Finite Element Method (FEM) is a general method for solving both ordinary and partial differential equations. In this section we show how it can be used for solving the wave equation. Our approach is not the only possible one, since our derivation ends up (again) in a system of ordinary differential equations, which we solve by the methods presented earlier—another approach would be to use a FEM formulation also for the time-dependent ordinary differential equation.

The general ideas behind the FE method rely heavily on concepts of so-called Hilbert and Sobolev spaces. We develop the method from bottom up and do not present these more advanced concepts.

Introduction: FEM takes a fundamentally different approach from the point-evaluation based finite difference methods described earlier. The idea is to seek for the solution as a finite linear combination of basis functions, so that the linear combination is, in a sense, the “best approximation” to the real solution from this finite-dimensional set of functions.

Basis functions are best described by an example, again in univariate functions on the interval $[0, 1]$. Suppose that we have scattered N node points x_i onto the interval, just as before. Now, we define a “hat” function $\phi_i(x)$ corresponding to each x_i ; the hat function takes the value 1 at x_i and ramps linearly to 0 towards x_{i-1} and x_{i+1} . Now, define an N -vector ζ , with its components denoted ξ_i and let:

$$u(x) = \sum_{i=1}^N \xi_i \phi_i(x) \tag{21}$$

Figure 2 shows an example with eleven nodes, the corresponding piecewise linear basis functions and an example linear combination of the basis functions. The linear span of the basis functions is a vector space in the sense that the sum of any two such functions is again a function which can be represented in the same way.

By the support of a function we mean the smallest set outside which the function is identically zero. The functions in the previous example are defined piecewise and they have small supports. This has computational advantages, as will become obvious after we have formulated the full finite element method in the next sections. However, the development itself does not rule out use of functions with global supports, meaning functions that are nonzero on the whole

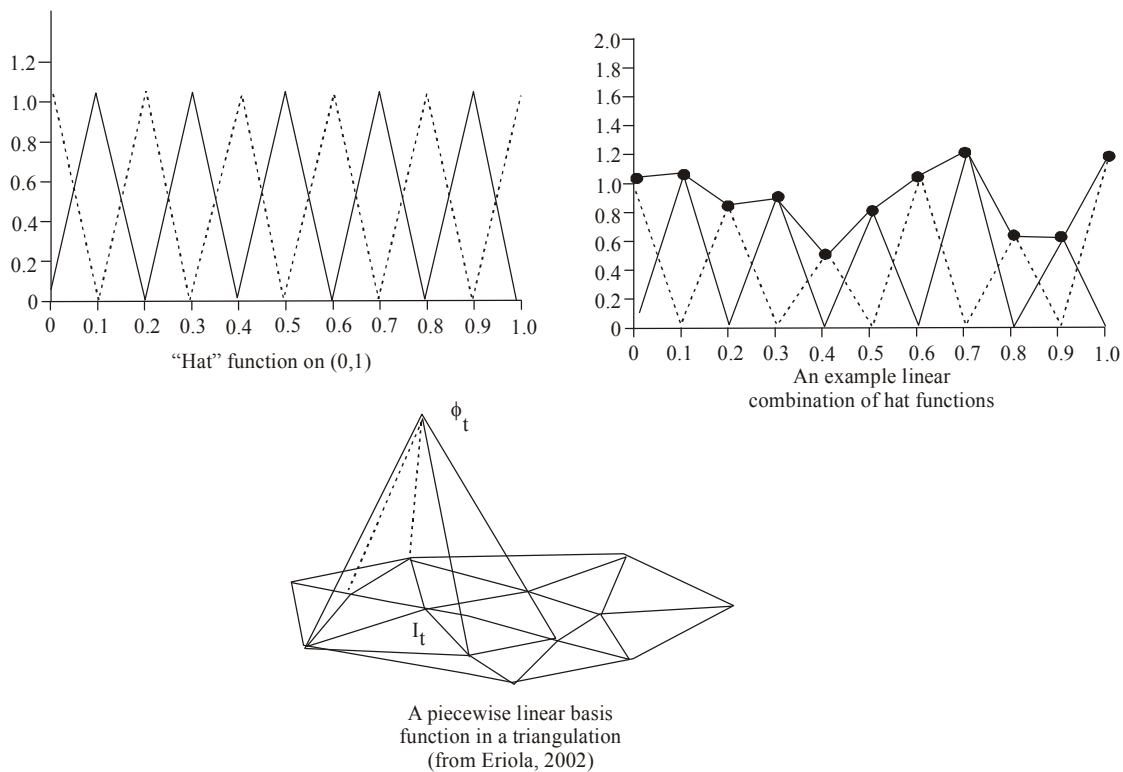


Fig. 2: Illustration of piecewise linear “hat” functions on the interval $[0, 1]$ and their linear combinations. The hat functions themselves are nonzero only at their corresponding nodes and between the adjacent nodes to both directions. Their linear combinations (in red), on the other hand, define piecewise linear functions on the whole interval

domain. For instance, all polynomials on the whole interval $[0, 1]$ have global supports, as well as the usual trigonometric polynomials $\sin(2\pi kx)$, with $k \in \mathbb{N}$. Also, piecewise polynomial functions find common use in FEM applications; the “hat” functions of the example are 1st order piecewise polynomials. To be completely precise, the name finite element method is used only if locally supported basis functions are used; the more general case is the Galerkin method.

The two- and three-dimensional analogue to our hat function example is to scatter points inside the domain Ω and construct a triangulation of them in 2D or a tetrahedralization in 3D. The piecewise linear functions now take a form where the functions have value 1 at the corresponding node and ramp to zero linearly inside the triangles (or tetrahedra) associated with the node. Also higher-order piecewise polynomials can be used. (Globally supported basis functions cannot be used in non-simple higher-dimensional geometries because of incompatibilities with the boundary conditions. This is a more advanced topic, and will not be pursued here.)

We also mention that triangles or tetrahedra are not the only possible geometrical primitives that can be used for constructing the basis functions. For instance, quadrilaterals and parallelepipeds may be used with piecewise bilinear or bicubic functions, respectively.

Variational formulation of the wave equation: In this section we will show how to search for the solution to the wave equation as a linear combination of basis functions. Since the solutions are time-dependent, we will make the coefficient vector ξ time-dependent also, so that the spatially discretized solution has the form $u_h(t, x) = \sum_{i=1}^N \xi_i(t) \phi_i(x) \approx u$.

To start off, we move all the terms of the wave equation onto the other side and get:

$$u_{tt} - \Delta u - f = 0 \tag{23}$$

Now, let V denote the set of bounded, continuous functions defined on having piecewise continuous first derivatives (gradients in higher dimensions) and fulfilling the spatial boundary conditions of the problem. (We note that the “hat” functions, used as examples in the previous section, are members of V , when $\Omega = [0, 1]$, neglecting boundary conditions.)

V is obviously infinite-dimensional and the exact solution u to (23) is also a member of this space. The fundamental theorem of calculus of variations (Guenther and Lee, 1996) states that if we multiply Eq. (23) by any function in V and integrate the product over Ω , we must still get 0:

$$\int_{\Omega} (u_{tt} - \Delta u - f)v dx = 0, \forall u \in V_0 \tag{24}$$

The goal of the following development is to simplify this equation in this general case and then in the end to switch focus to a finite-dimensional subspace

V_h of V spanned by the basis functions $\phi_i(x)$. This will allow us to write out a linear system of equations for the unknown coefficients $\xi(t)$.

It is easy to verify by Gauss’ divergence theorem 10 that:

$$\int_{\Omega} v \Delta u dx = - \int_{\Omega} \nabla v \cdot \nabla u dx + \int_{\partial \Omega} v \nabla u \cdot n d\sigma \tag{25}$$

where $d\sigma$ is an area element on $\partial \Omega$. Applying this result to Eq.(24) we have:

$$\int_{\Omega} (u_{tt} v + \nabla u \cdot \nabla v - f v) dx - \int_{\partial \Omega} v \nabla u \cdot n d\sigma = 0, \forall v \in V \tag{26}$$

which is called the variational formulation of the wave equation. The solutions to (26) are called weak solutions to the wave equation. Classical theory of partial differential equations shows that any sufficiently smooth u that solves (26) is also a classical solution of the wave equation (Eirola, 2002).

Now we approach the heart of the matter. Substituting the approximation (22) into (26) and requiring that (26) only holds for the members ϕ_k of V_h (and not the “full” V) yields:

$$\sum_{i=1}^N \xi_i'' \int_{\Omega} \phi_i(x) \phi_k(x) dx + \sum_{i=1}^N \xi_i \int_{\Omega} \nabla \phi_k(x) \cdot \nabla \phi_i(x) dx - \tag{27}$$

$$\sum_{i=1}^N \xi_i \int_{\partial \Omega} \phi_k(x) \nabla \phi_i(x) \cdot n d\sigma = \int_{\Omega} f(x) \phi_k(x) dx, \quad \forall \phi_k \in V_h \tag{28}$$

where, we have used:

$$u_h'' = \sum_{i=1}^N \xi_i''(t) \phi_i(x), \quad \nabla u_h = \sum_{i=1}^N \xi_i(t) \nabla \phi_i(x)$$

with primes denoting differentiation with respect to time. We have also switched the order of summation and integration in the terms. Because V_h is finite dimensional, (27) is actually a set of N linear equations (one for each ϕ_k , with $k = 1, \dots, N$) for the coefficient vector ξ . Hence it can be written in a matrix form as:

$$\xi'' A + \xi S - \xi B = f \tag{29}$$

with:

$$A_{ij} = \int_{\Omega} \phi_i(x) \phi_j(x) dx, \tag{30}$$

$$S_{ij} = \int_{\Omega} \nabla \phi_i(x) \cdot \nabla \phi_j(x) dx,$$

$$B_{ij} = \int_{\partial \Omega} \phi_i(x) \nabla \phi_j(x) \cdot n d\sigma, \text{ and } f_j = \int_{\Omega} f(x) \phi_j(x) dx. \tag{31}$$

Again, we are left with a linear system of ordinary differential equations, for which all the time integration methods described in section 4 apply directly.

As mentioned before, the sizes of the supports of the basis functions affect the computational load associated with FEM. The cost of evaluating the integrals in the expressions for the above matrices' elements benefits from locally supported functions, since small supports mean small nonzero regions in the integrands.

A teaser: In the previous section we derived a spatial discretization of the wave equation as a linear combination of a finite number of prescribed basis functions. We have left out most of the mathematical structure that helps to understand the FE method from a more geometrical viewpoint because of lack of space. We do still mention a remarkable property; the finite element solution to the wave equation is optimal in the sense of a certain squared difference between the real solution and the approximate one, which means that no other linear combination of the basis functions could achieve a smaller error in this least squares-sense. This is quite surprising, considering that we do not have knowledge of the exact solution! This fact is best explained by stating (albeit cryptically) that the finite element solution is an orthogonal projection of the exact solution onto the finite dimensional linear span of the basis functions. For more information on vector spaces of functions, see for instance (Kreyszig, 2007).

DISCUSSION

The previous sections have shown how the wave equation can be reduced into a system of ordinary differential equations either by finite difference approximations or by the finite element method. This section discusses some practical aspects of the methods presented above and outlines some differences between them.

Solution of large linear systems: In general, time integration requires solution of linear systems at each time step. These systems are generally very large, routinely in the order of several millions of unknowns. It is clear that direct solution, e.g., by Gaussian elimination, of the resulting equations is not feasible. The rescue lies in iterative methods, which do not manipulate the matrix (as Gaussian elimination does), but instead work by starting from an initial guess vector for the solution and then improving upon it in a successive series of iterations, until some degree of convergence is reached. Classical iterative methods include the Jacobi and Gauss-Seidel iterations. So-called Krylov subspace methods, such as the conjugate gradient method, search for the solution in the span of $\{x, Ax, A^2x, \dots\}$, where x is the initial guess and A is the matrix in the problem. Golub and Van Loan (1996) for an in-depth review of iterative methods.

Differences between the methods: The key difference between the finite difference method and FEM lies in

the composition of the matrices. Once the matrices have been formed, the time stepping solution to the wave equation proceeds similarly.

Generally, the matrices inherent to the finite difference method have regular coefficients for nodes inside the domain; that is, all the nodes behave numerically in the same way. The matrices of FEM are more irregular, since their elements are integrals of the basis functions' products and in general domains the basis functions do not form a regular structure. This makes construction of the FEM matrices much more involved than that of finite difference matrices.

Boundary conditions need special treatment in the finite difference method; detailed calculations on how to discretize different types of BCs are required. As stated earlier in the FEM section, the space V , from where the solutions are being searched for, is defined such that the basis functions "fulfill the boundary conditions" in a certain way; the rest of the BCs are enforced weakly in the form of the boundary integral in Eq. (27). In other words, FEM incorporates boundary conditions into its formulation in a unified way that alleviates some of the need for their special treatment.

Also, generation of the triangulations and tetrahedralizations used frequently in FEM applications is far from trivial and a wealth of research on the construction and quality of the subdivisions exists. Finite difference methods rely on structured grids and hence are not dependent on such algorithms.

Practical considerations: The methods presented above solve for the sound field inside an enclosure in a rigorous way, i.e., the error in the solution can be made arbitrarily small (bound, of course, by machine precision) by adding more discretization nodes, using higher-order basis functions (FEM) or higher-order derivative approximations (FD) and using better time integrators.

Despite the correctness of the algorithms, direct application of these methods does not yield a practical system for solving for the impulse responses, at least if the whole frequency range of human hearing is to be simulated. This is obvious from geometry alone; depending on the particular method used, the spatial discretization needs 6-10 nodes per wavelength in order to resolve the frequencies faithfully. At 22 kHz one wavelength is approx. 1.5 cm and thus the spacing between the nodes needs to be 1.5-2.5 mm. Thus, one cubic meter of space needs to be filled with 64-300 million nodes and this translates directly to the same number of unknowns to solve for in the simulator. Here from it is obvious that full frequency range simulation of spaces of realistic size is as of yet completely unfeasible and some hybrid methods combining direct numerical simulation by wave field decomposition techniques (such as the image source method) must be utilized. For reference, modern scientific

computing uses meshes with up to ten million elements. If a rectangular room with a 45 m³ volume was discretized with ten million elements, the average density of mesh points would be approx. 15 cm, corresponding to a maximum frequency of 220-370 Hz only. Clearly, direct numerical simulation of acoustic phenomena in the whole frequency range remains out of our grasp at present. This is perhaps not too unfortunate, since for instance the image source method augmented by edge diffraction sources produces an exact solution for planar geometries [12].

REFERENCES

- Atkinson, K. and W. Han, 2009. *Theoretical Numerical Analysis*. Springer, New York, pp: 625.
- Eirola, T., 2002. *Osittais differentiaaliyht*. Lecture Notes.
- Eriksson, K., D. Estep, P. Hansbo and C. Johnson, 1996. *Computational Differential Equations*. Cambridge University Press, Lund Studentlitteratur, pp: 554, ISBN: 0521567386.
- Golub, G. and C. Van Loan, 1996. *Matrix Computations*. 3rd Edn., Johns Hopkins University Press, Baltimore, MD.
- Guenther, R.B. and J.W. Lee, 1996. *Partial Differential Equations of Mathematical Physics and Integral Equations*. Courier Dover Publications, New York, pp: 562.
- Kreyszig, E., 2007. *Introductory Functional Analysis with Applications*. Wiley India, New York, pp: 704.
- Pierce, A.D., 1989. *Acoustics-An Introduction to its Physical Principles and Applications*. Acoustical Society of America, Woodbury, NY, pp: 678.
- Savioja, L., 1999. *Modeling techniques for virtual acoustics*. D.Sc. Thesis, Publications in Telecommunications. Software and Multimedia, TML-A3, Helsinki University of Technology.