Numerical Experiments with Non-linear Double Membrane Drums

Alberto Torin
Acoustics and Audio Group
University of Edinburgh
A.Torin@sms.ed.ac.uk

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh
sbilbao@staffmail.ed.ac.uk

ABSTRACT

Drums with two membranes are very common; snare drums, tom-toms and bass drums can be found in Western music, but there are examples in Eastern music, as well (the Indian mridangam, the Japanese taiko, etc.) These instruments can have considerable physical dimensions; bass drum heads can sometimes reach a radius of half a meter. Given the size, the low tension at which the membranes are generally tuned and the amplitude of vibrations, it is unlikely that a linear model could capture the most salient features of the sound of these instruments.

Pitch glide effects and an increase of high-frequency energy has been observed at high excitation amplitudes for the bass drum [1] and more recently for tom-toms [2]. Similar phenomena have been observed, for example, in strings [3] and plates [4], and are often related to the presence of non-linearities in the system.

In this paper we present a finite difference time domain model of double membrane drums (i.e., tom-toms and bass drums) with air coupling and with non-linear terms (due to von Kármán) in the equations of motion for the two membranes. Some of the computational difficulties stemming from this particular choice will be discussed. Simulation results and sound examples will be presented.

1. INTRODUCTION

Percussion instruments, and drums in particular, have received growing attention in recent years. Several experimental and numerical studies have been performed on kettledrums [5, 6] and snare drums [7, 8], but research on bass drums remains quite scarce [1].

Numerical simulations of musical instruments based on physical models are becoming an attractive approach both in acoustical studies and sound synthesis. Among the various techniques adopted, finite difference time domain methods appear to be a versatile tool to tackle these kinds of problems rigorously, especially in the presence of several coupled components, possibly with non-linear interactions [9]. The main disadvantage of such methods, namely, their heavy computational complexity, is becoming less urgent due to the increasing availability of parallel hardware, such as graphical processing units (GPGPUs). In fact, for algorithms whose update for the various components can be performed with a high degree of independence, as is often the case for FDTD methods over regular grids, the speed-up that one can obtain with respect to a code written for a CPU can be significant (even tens of times) [10].

Despite these improvements, however, some challenges are still open. The solution of a linear system, for example, remains a difficult task even in the case of FDTD methods, where linear systems to be solved are sparse. Simple methods, like Gaussian elimination, because of their essentially serial nature, are not suited for implementation on a GPU. One of the goals of the ongoing NESS Project [11] at the University of Edinburgh is to devise numerical algorithms for sound synthesis that can eventually be parallelized on GPGPUs. The present paper may also be considered, then, as a first step in the study of parallel implementations of large sparse linear systems. The non-linear drum model described below becomes an ideal test case for these kinds of experiments.

The physical description of the system, which is mainly based on approaches recently applied to the snare drum [9] and to the timpani drum [12], is given in Section 2, and includes the coupling of the drum with the surrounding air field. The main novelty of the present model is the introduction of a von Kármán non-linearity in the membrane equation [13]. In Section 3 a finite difference numerical scheme will be presented which is stable and which dissipates energy strictly. The core of the algorithm is the solution, in the run time loop, of a sparse linear system which must be constructed anew at each time step. Given the size of the system, this results in a computationally very demanding task. In Section 4 we will discuss a possible method to accelerate the computation based on purpose-designed iterative methods. Finally, simulation results will be presented in Section 5.

2. MODEL SYSTEM

The drum model under consideration can be schematically divided into several interacting components: two membranes, a rigid shell connecting them, the acoustic field in which the system is embedded. (See Fig. 1.) Typical bass drums have membranes with radius between 40 cm and 50 cm, with height between 35 cm and 45 cm, while tom-toms are generally smaller [14].

The approach adopted here and the geometry of the system are similar to those used recently in [9], therefore some of the details will be omitted. A finite computational space $\mathcal{V}$ is needed in which the virtual drum can be embedded.

This enclosure, or "box", is not intended to simulate a real 3D room, but rather to emulate the behaviour of an infinite space by applying absorbing conditions at the boundaries $\partial\mathcal{V}$ (see Sec. 2.3.)

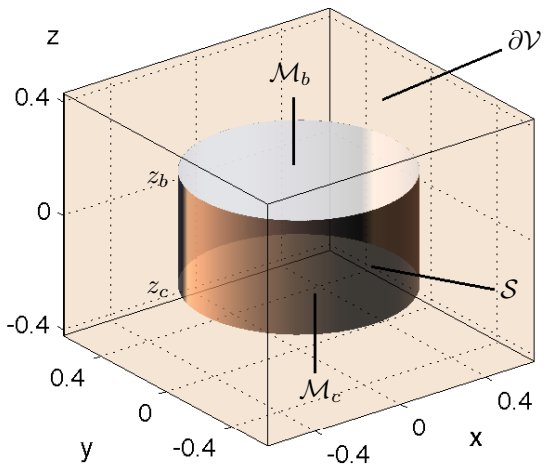


Figure 1. Geometry of the model. Batter membrane $\mathcal{M}_b$ at position z_b, carry membrane $\mathcal{M}_c$ at position z_c, rigid shell $\mathcal{S}$, computational region $\mathcal{V}$ with boundaries denoted by $\partial\mathcal{V}$.

2.1 Membranes

The most important components of bass drums and tom-toms are two heads, which are responsible for the sound production. The upper one, called the batter head, is usually struck with a mallet, while the lower one, called the carry membrane, is set into motion by the air inside the cavity. They are defined over two circular regions $\mathcal{M}_b$ and $\mathcal{M}_c$ of radius R and at positions $z = z_b$ and $z = z_c$, respectively. As is usual in the literature, we will only consider the displacement $w(x, y, t)$ of the membrane normal to the surface, at position (x, y) and time t.

As pointed out in [1], typical blows on the batter head can cause displacements of several millimiters, at least one order of magnitude bigger, therefore, than the thickness of the membrane itself. It is reasonable to expect non-linear effects to become important in these conditions, similarly to what happens for strings [15].

We can write, now, the equations of motion for the two membranes:

$$\frac{\partial^2 w^{(i)}}{\partial t^2} = c_i^2 \Delta_{2D} w^{(i)} - \kappa_i^2 \Delta_{2D}\Delta_{2D} w^{(i)} + \sigma_i \Delta_{2D} \frac{\partial w^{(i)}}{\partial t} \frac{1}{\rho_i H_i}(f_i^+ + f_i^-) + \delta_{i,b} \frac{1}{\rho_b H_b} \delta(x_b - x_0, y_b - y_0) f_{exc} + \frac{1}{\rho_i H_i} \mathcal{L}(w^{(i)}, \Phi^{(i)}). \quad (1)$$

Here, the index $i = b, c$ refers to the batter or carry membranes respectively. The first line groups the terms for a linear stiff membrane with viscoelastic loss [16]. The differential operators Δ_{2D} and $\Delta_{2D}\Delta_{2D}$ are the Laplacian and biharmonic operators, with

$$\Delta_{2D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \quad (2)$$

The membrane speeds c_i and stiffness parameters κ_i are given by:

$$c_i = \sqrt{(T_i/\rho_i H_i)} \qquad \kappa_i = \sqrt{E_i H_i^2 / 12\rho_i(1-\nu_i^2)}, \quad (3)$$

where (dropping the subscripts) T is the membrane tension per unit of length, in kg/s^2, ρ is the membrane density, in kg/m^3, H is the membrane thickness, in m, E is Young's modulus, in kg/s^{2}m, and ν is the dimensionless Poisson's ratio. σ is the coefficient governing viscoelastic losses, in m^2/s. All these parameters can in principle be distinct for the two membranes.

The two terms in the second line of (1) take into account external forces acting on the membranes. f^+ and f^- represent the pressure due to the presence of air acting above and below the surface of the membrane, while f_{exc} is a pointwise excitation force, acting on the batter membrane only.

The last term in (1) introduces in the system geometric non-linear effects, that are widely discussed in relation to plate vibration [13]. The action of the non-linear operator $\mathcal{L}$ on two test functions ξ and χ is given by:

$$\mathcal{L}(\xi, \chi) = \frac{\partial^2 \xi}{\partial x^2}\frac{\partial^2 \chi}{\partial y^2} + \frac{\partial^2 \chi}{\partial x^2}\frac{\partial^2 \xi}{\partial y^2} - 2\frac{\partial^2 \xi}{\partial x \partial y}\frac{\partial^2 \chi}{\partial x \partial y}. \quad (4)$$

$\Phi(x, y, t)$ is the so-called Airy's stress function, and must satisfy the following constraint:

$$\Delta_{2D}\Delta_{2D}\Phi^{(i)} = -\frac{E_i H_i}{2}\mathcal{L}(w^{(i)}, w^{(i)}). \quad (5)$$

(1) and (5) form, then, a set of two coupled equations for each membrane.

Two boundary conditions for both $w^{(i)}$ and $\Phi^{(i)}$ must be supplied; the membranes are assumed to be fixed at the boundary but free to rotate, a condition generally referred to as "simply supported" in the plate literature, while for the Airy's functions a free condition is used [17]:

$$w^{(i)} = 0 = \Delta_{2D} w^{(i)}, \quad \Phi^{(i)} = 0 = \mathbf{n}_{i,ext} \cdot \vec{\nabla}_{2D} \Phi^{(i)}, \quad (6)$$

where $\mathbf{n}_{i,ext}$ is the unit vector normal to the boundary and $\vec{\nabla}_{2D}$ is the gradient.

2.2 Air

Wave propagation in air is modelled by means of a 3D wave equation:

$$\frac{\partial^2 \Psi}{\partial t^2} = c_a^2 \Delta_{3D} \Psi, \quad (7)$$

where c_a is the speed of sound in air (here, 340 m/s), and the 3D Laplacian operator is defined as:

$$\Delta_{3D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \quad (8)$$

Ψ is a velocity potential related to the pressure p and particle velocity $\mathbf{v}$ by:

$$p = \rho_a \frac{\partial \Psi}{\partial t} \qquad \mathbf{v} = -\vec{\nabla}_{3D}\Psi, \tag{9}$$

where ρ_a is the density of air (1.21 kg/m^3) and $\vec{\nabla}_{3D}$ is the 3D gradient.

2.3 Absorbing Boundary Conditions

As mentioned before, it is necessary to house the virtual drum in a finite computational region $\mathcal{V}$. However, since the purpose is that of simulating (ideally!) an infinite space, we must impose absorbing boundary conditions over the walls $\partial\mathcal{V}$ of this box.

One possible way of achieving this is by means of perfectly matched layer, a standard approach drawn from electromagnetism [18]. In the present case, however, in order to reduce the computational complexity, we chose to adopt Engquist Majda equations [19], a family of absorbing conditions with increasing order of accuracy. In the present work, we will use a first order approximation defined as:

$$\left(\frac{\partial}{\partial t} + c_a \mathbf{n} \cdot \vec{\nabla}_{3D}\right) \Psi = 0, \tag{10}$$

where $\mathbf{n}$ denotes the unit vector normal to the wall and pointing outwards.

2.4 Cavity

The drum cavity is modelled as a perfectly rigid shell $\mathcal{S}$ enclosing the air between the two membranes. It is a cylindrical surface of radius R, between $z = z_c$ and $z = z_b$. Neumann conditions are applied to the velocity potential over the shell surface

$$\mathbf{n}_S \cdot \vec{\nabla}_{3D}\Psi = 0 \tag{11}$$

where $\mathbf{n}_S$ is the unit vector normal to the cylindrical surface. This condition must be imposed both inside and outside the cavity.

2.5 Coupling Conditions

As we anticipated in Sec. 2.1, an explicit expression for f^+ and f^- appearing in (1) must be given. The pressure acting on the membranes must be equal to the pressure of the acoustic field above and below the surfaces. Given the relation between Ψ and p, we can write:

$$f_i^+ = -\rho_a \lim_{z \to z_i^+} \partial_t \Psi \mid_{\mathcal{M}_i} \qquad f_i^- = \rho_a \lim_{z \to z_i^-} \partial_t \Psi \mid_{\mathcal{M}_i} . \tag{12}$$

Another condition must be supplied, namely, that the velocity of the membrane be equal to the velocity of the acoustic particles on either side of the membrane. Mathematically,

$$\partial_t w_i = - \lim_{z \to z_i^-} \partial_z \Psi \mid_{\mathcal{M}_i} = - \lim_{z \to z_i^+} \partial_z \Psi \mid_{\mathcal{M}_i} . \tag{13}$$

Conditions (12) and (13) hold over the membrane regions $\mathcal{M}_b$ and $\mathcal{M}_c$.

2.6 Excitation and Output

The excitation mechanism underlying the mallet-membrane interaction has been the subject of several studies. Its inherently non-linear nature is discussed, e.g., in [6]. A finite difference model of such an excitation is indeed possible [20], but is complicated by the interaction with the non-linearity of the membrane. Given the very short duration of the mallet-membrane interaction (on the order of 2-4 ms), and to simplify the implementation, we chose to adopt another approach. Each strike is modelled as a raised cosine impulse in time acting at a single point of the membrane specified by a 2D Dirac function [20]. More realistic models that would allow a finer control of the sound are currently under study.

Output sounds are obtained by sampling the variations of pressure generated at a certain point in the acoustic field by a strike on the batter membrane.

3. FINITE DIFFERENCE SCHEMES

In this section we discuss the numerical implementation of the model presented above using finite difference time domain methods [21].

The system described in the previous section will be approximated over regular Cartesian grids and at discrete time instants, which are integer multiples of a time step k. In sound synthesis simulations, it is customary to choose the sample rate F_s *a priori* for perceptual reasons, and to obtain k consequently as $k = 1/F_s$.

The displacement function $w^{(i)}(x, y, t)$ for each membrane can be approximated by a discrete function $w_{l,m}^{n,(i)}$ defined over a spatial grid of spacing h and time step k, such that:

$$w^{(i)}(x, y, t) \approx w_{l,m}^{n,(i)} \equiv w^{(i)}(lh, mh, nk) \tag{14}$$

for integers l, m and n.

We can now introduce forward, backward and identity shift operators in time, whose action on $w_{l,m}^n$ is given by:

$$e_{t+} w_{l,m}^n = w_{l,m}^{n+1},\ e_{t-} w_{l,m}^n = w_{l,m}^{n-1},\ 1 w_{l,m}^n = w_{l,m}^n. \tag{15}$$

Analogous relations hold for spatial shift operators; along the x axis, for example, we have:

$$e_{x+} w_{l,m}^n = w_{l+1,m}^n,\ e_{x-} w_{l,m}^n = w_{l-1,m}^n,\ 1 w_{l,m}^n = w_{l,m}^n. \tag{16}$$

These operators form the basic units constituting finite difference operators.

The choice of Cartesian vs. polar grids over a circular geometry is dictated by various concerns, discussed in detail in [9], the most important of which is the bandlimited output that is generated by schemes in polar coordinates. Typical polar schemes operating at a sample rate of 44.1 kHz produce outputs bandlimited to approximately 3 kHz.

For computational reasons, it is useful to store 2D grids as column vectors, by taking the values of the array columnwise. (See Fig. 2 for an example.) Difference operators can be written, therefore, as sparse matrices.

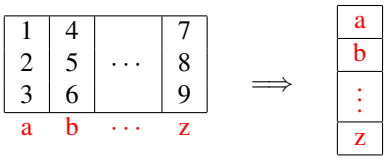


Figure 2. A 2D grid (left) is reshaped into a column vector (right) by placing each column (in red) in a vertical array.

3.1 Membranes

A finite difference scheme for the coupled equations (1) and (5) may be written as:

$$\delta_{tt} w^{(i)} = c_i^2 \delta_{2\Delta} w^{(i)} - \kappa_i^2 \delta_{2\Delta}\delta_{2\Delta} w^{(i)} + \sigma_i \delta_{t-}\delta_{2\Delta} w^{(i)} + \frac{1}{\rho_i H}{}_i \left(f_i^+ + f_i^-\right) + \delta_{i,b} \frac{1}{\rho_b H_b} \delta(l_b - l_0, m_b - m_0) f_{exc} + \frac{1}{\rho_i H_i} \mathfrak{l}(w^{(i)}, \mu_{t\cdot}\Phi^{(i)}), \quad (17)$$

$$\delta_{2\Delta}\delta_{2\Delta}\mu_{t-}\Phi^{(i)} = -\frac{EH}{2}\mathfrak{l}(w^{(i)}, e_{t-}w^{(i)}), \quad (18)$$

where, in order to simplify an already heavy notation, we put $w^{(i)}$ instead of $w_{l,m}^{n,(i)}$. Integers l_0 and m_0 represent the nearest grid point to the continuous excitation point (interpolation may be used, as well.) The explicit actions of the various operators involved in the previous equations can be written in terms of shift operators defined in (15) and (16):

$$\delta_{tt} = \frac{1}{k^2}\left(e_{t+} + e_{t-} - 2\right), \quad (19a)$$

$$\delta_{2\Delta} = \frac{1}{h^2}\left(e_{x+} + e_{x-} + e_{y+} + e_{y-} - 4\right) \quad (19b)$$

$$\delta_{t-} = \frac{1}{k}\left(1 - e_{t-}\right) \quad (19c)$$

$$\mu_{t-} = \frac{1}{2}\left(1 + e_{t-}\right) \quad (19d)$$

$$\mu_{t\cdot} = \frac{1}{2}\left(e_{t+} + e_{t-}\right). \quad (19e)$$

This particular choice for the discretization of the nonlinear terms appearing in (1) and (5) is but one in a family of finite difference schemes for the von Kármán equation [22], and leads to provable energy conservation when the following expression for $\mathfrak{l}$ is used:

$$\mathfrak{l}(\xi,\chi) = \delta_{xx}\xi\delta_{yy}\chi + \delta_{yy}\xi\delta_{xx}\chi - \frac{1}{2}\left(\delta_{x-,y-}\xi\delta_{x-,y-}\chi + \delta_{x+,y-}\xi\delta_{x+,y-}\chi + \delta_{x-,y+}\xi\delta_{x-,y+}\chi + \delta_{x+,y+}\xi\delta_{x+,y+}\chi\right). \quad (20)$$

The finite difference version of boundary conditions (6) can be written as follows:

$$w^{(i)} = 0 = \delta_{2\Delta} w^{(i)}, \quad \Phi^{(i)} = 0 = \delta_n \Phi^{(i)} \quad (21)$$

where δ_n represents the non-centred spatial derivative over the boundary.

Coupling conditions f^+, f^- and f_{exc} in (17) are the discrete counterpart of those in (1), and their explicit expression will be given in Sec. 3.3.

3.2 Acoustic field

The acoustic field $\Psi(x,y,z,t)$ can be approximated by a discrete function $\Psi_{l,m,p}^n$ over a 3D Cartesian grid with spacing h_a. The finite difference discretization of (7) that has been adopted is:

$$\delta_{tt}\Psi = c_a^2 \delta_{3\Delta}\Psi, \quad (22)$$

where the 3D Laplacian operator is defined as:

$$\delta_{3\Delta} = \frac{1}{h_a^2}\left(e_{x\pm} + e_{y\pm} + e_{z\pm} - 6\right), \quad (23)$$

and $e_{j\pm} = e_{j+} + e_{j-}$ for $j = x, y, z$.

The boundary condition (11) over the drum shell can be implemented with a simple staircase approximation [9]. For a grid point Ψ_{in} inside the cavity with a nearest neighbour Ψ_{out} outside, a modified version $\bar{\delta}_{3\Delta}$ of the 3D Laplacian can be written as:

$$\bar{\delta}_{3\Delta}\Psi_{in} = \delta_{3\Delta}\Psi_{in} + \frac{1}{h_a^2}\left(-\Psi_{out} + \Psi_{in}\right). \quad (24)$$

A similar equation holds for Ψ_{out}, with the subscripts exchanged, and it can be easily extended when the nearest neighbours outside or inside are two.

At the boundary of the computational box, Engquist Majda condition (10) can be discretized with centred operators. At $p = 1$, e.g., we can write:

$$\delta_{t\cdot}\Psi_{l,m,1} - c_a \delta_{z\cdot}\Psi_{l,m,1} = 0. \quad (25)$$

This equation can be inserted into (22) to obtain an explicit update form for the acoustic field over the boundary [9].

3.3 Coupling Conditions and Excitation

The coupling mechanism described in Sec. 2.5 can be implemented in an easier way if the vertical positions z_i of both membranes are chosen half way between two neighbouring sets of points of the acoustic field with indeces p_i^- and p_i^+, so that $p_i^- = z_i - h_a/2$ and $p_i^+ = z_i + h_a/2$. This may require to adjust the value of h_a accordingly. (See Fig. 3.)

Since the various grids adopted for the membranes and for the velocity potential have different spacings, interpolation is also necessary. Considering the coupling between the batter membrane and air first, we introduce two interpolants $\mathcal{J}$ and $\mathcal{I}$ operating from the membrane to a 2D subset of the acoustic field grid, and vice versa. When written in matrix form, they must satisfy the following relation for energy conservation reasons [9]:

$$\mathcal{I} = \frac{h_a^2}{h^2}\mathcal{J}^T, \quad (26)$$

where the superscript T denotes matrix transposition. Now, we can write (12) as:

$$f^+ = -\rho_a \mathcal{I}\delta_{t\cdot}\Psi^+, \qquad f^- = \rho_a \mathcal{I}\delta_{t\cdot}\Psi^-, \quad (27)$$

where Ψ^+ and Ψ^- represent the 2D slices of the grid at position p^+ and p^-, respectively. Equations (13) become:

$$\mathcal{J}\delta_{t\cdot}w = -\delta_{z-}\Psi^+ = -\delta_{z+}\Psi^-. \quad (28)$$

Equations (28) can be inserted into (27) to obtain an expression for f^+ and f^- in terms of known values and w^{n+1}, which leads to an implicit update expression for the membrane itself.

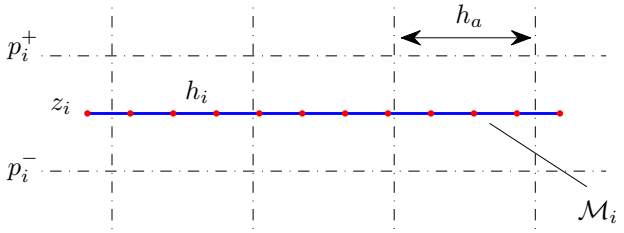


Figure 3. Vertical cross section of the acoustic field grid (in black). The membrane $\mathcal{M}_i$ (in blue) is placed at position z_i, halfway between 3D grid points p_i^+ and p_i^-. Spacings for the 2D and 3D grids are h_i and h_a, respectively.

3.4 Stability and Numerical Energy

Stability conditions for the finite difference schemes written above can be obtained through energy analysis [20]. For the membranes schemes (17) we obtain:

$$h_i^2 \geq c_i^2 k^2 + 4k\sigma_i + \sqrt{(c_i^2 k^2 + 4k\sigma_i)^2 + 16\kappa_i^2 k^2}, \quad (29)$$

while for the acoustic field equation (22) we can write

$$h_a^2 \geq 3c_a^2 k^2. \quad (30)$$

In both cases, spatial grid steps are chosen as close as possible to their minimum value.

It can be shown that a numerical energy $\mathfrak{h}$ exists for the system, which is positive definite and strictly dissipated (that is, $\mathfrak{h}^{n+1} < \mathfrak{h}^n$.) If coefficients σ_i are set to zero and reflecting conditions are applied over $\partial\mathcal{V}$, $\mathfrak{h}$ remains constant to machine precision. This is a powerful tool for debugging purposes, as virtually any error in the code has an influence on $\mathfrak{h}$. See Section 5.4.

4. IMPLEMENTATION DETAILS

The implementation of the model described in the previous sections has been carried out in MATLAB. Operations involving matrices and vectors can be performed in a natural way in this environment. Problems that require the solution of a sparse linear system, for example, could be solved with a direct method without much difficulties. However, with some extra work it is possible to take advantage of the particular characteristics of the system under consideration to devise more efficient ways to perform this operation. In the present case, a naïve implementation of the scheme for the non-linear membranes in (17) and (18) would lead to a code that would barely run, even at low sample rates, on standard machines.

In this section, we will quantify the size of the system and we will describe and compare different implementation strategies for the membrane equation, namely exact methods and an *ad hoc* iterative method. The energy conservation of the scheme can be preserved to machine accuracy in both cases.

4.1 Algorithm and System Size

A schematic description of a possible implementation of the algorithm outlined in the previous section is given in [12]. In the present case, however, the size of the system can be much larger. Typical physical parameters for the membranes and the air at audio sample rate lead to the grid sizes in Table 1.

Component	Dimensions	Total grid points
2D membranes	265×265	70,225
3D space	$110 \times 110 \times 110$	1,331,000

Table 1. Dimensions of finite difference grids for the various components at $F_s = 44.1$ kHz. Here the 2D grid is for a single bass drum membrane of radius $R = 45$ cm with typical physical parameters [1] in a 1.5 m^3 box.

4.2 Linear System Solution

The update scheme for the membranes given by the coupled equations (17) and (18) is inherently implicit, due to two independent factors: the presence of air coupling, on the one hand, and the non-linearity on the other. Their separate action has been analysed in detail in [9] and [22], respectively, but they have never been adopted simultaneously, so far. The Berger-type non-linearity described in [12], in fact, despite its appearance can be written as an explicit update. Though there is no conceptual difficulty in finding the update scheme, it becomes very challenging from a computational point of view to put into practice the algorithm in the present case, especially because of the large number of grid points involved.

It is possible to "vectorize" the grid functions w and Φ for the two membranes and combine them into a single column vector. When written in matrix-vector form, and after taking into account the air coupling conditions (27) and (28), the update recursion for w^{n+1} and Φ^{n+1} can be written schematically as

$$\begin{bmatrix} \mathbf{1} + \alpha\mathcal{I}\mathcal{J} & -L_1 \\ L_2 & \Omega \end{bmatrix} \begin{bmatrix} w^{n+1} \\ \Phi^{n+1} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad (31)$$

where $\mathbf{1}$ is the identity matrix, $\alpha\mathcal{I}\mathcal{J}$ is the symmetric matrix coming from air coupling with $\mathcal{I}$ and $\mathcal{J}$ the interpolants defined in (26) and α a dimensionless coupling constant, $\Omega = \delta_{2\Delta}\delta_{2\Delta}$ is the biharmonic operator, L_1 and L_2 are square matrices related to the von Kármán operator (20), and $a = a(w^n, w^{n-1}, \Phi^{n-1})$ and $b = b(\Phi^n)$ are the vectors constructed from values from previous time steps. It is interesting to notice that the diagonal blocks are constant, while L_1 and L_2 must be computed at every time step, since they depend on the action of the $\mathcal{L}$ operator defined in (4) on w^n.

The system (31) has now the form $Sx = q$, and it is possible to solve it straightforwardly using, e.g., Gaussian elimination. However, the results in terms of computation speed

suggest to find a possible alternative (see Section 5.3.) It becomes natural, then, to take advantage of the fact that (under typical conditions for real drums) the coupling term α is small (order 10^{-2}) compared to the identity matrix to create an iterative method for solving the system. We start by writing $S = S_1 + S_2$, where

$$S_1 = \begin{bmatrix} \mathbf{1} & -L_1 \\ L_2 & \Omega \end{bmatrix}, \qquad S_2 = \begin{bmatrix} \alpha\mathcal{I}\mathcal{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (32)$$

where $\mathbf{0}$ represents a square matrix with all entries equal to 0 and $\mathbf{1}$ is the identity matrix. Incidentally, S_1 is the update matrix that one obtains from the simulation of a non-linear von Kármán model in a vacuum [22]. With these positions, it is possible to write the iteration

$$S_1 x_{j+1} = -S_2 x_j + q, \quad (33)$$

starting from an initial guess x_0. This method produces an approximate solution of (31) with the desired precision, if and only if every eigenvalue λ of $S_1^{-1}S_2$ satisfies $|\lambda| < 1$. Furthermore, the rate of convergence of this algorithm depends on the maximum size of $|\lambda|$ [23]. In the present case, the first condition holds and the algorithm requires 8-12 iterations to converge within machine accuracy. We can write (33) explicitly as

$$\begin{bmatrix} \mathbf{1} & -L_1 \\ L_2 & \Omega \end{bmatrix} \begin{bmatrix} w^{n+1} \\ \Phi^{n+1} \end{bmatrix}_{j+1} = \begin{bmatrix} \bar{a} \\ b \end{bmatrix}_j, \quad (34)$$

with $\bar{a}_j = -\alpha\mathcal{I}\mathcal{J}w_j^{n+1} + a$. The iterative method only concerns w, and $w_{j=0}^{n+1} = w^n$ is the natural choice for the initial guess. Now, dropping the superscript $n+1$, it is possible to write the system in terms of Φ_{j+1} only and subsequently update w_{j+1}:

$$(\Omega + L_2L_1)\Phi_{j+1} = b - L_2\bar{a}_j \quad (35a)$$

$$w_{j+1} = \bar{a}_j + L_1\Phi_{j+1}. \quad (35b)$$

For the solution of the linear system (35a) there are two options: an exact method or an iterative method. For sound synthesis purposes, in particular, single precision solutions may be satisfactory. In this case, an iterative solver with a less stringent tolerance may be adopted, which leads to an additional speed-up of the code (see Section 5.3 for details.) One possibility is to use a preconditioned conjugate gradient (*pcg*) algorithm [24], capitalising on the symmetric (apart from low rank corrections) and positive definite structure of the reduced matrix $S_r = (\Omega + L_2L_1)$. It has already been said that L_1 and L_2 depend on w^n, and they are responsible for the non-linear effects in the system. In fact, the product L_2L_1 in S_r can be seen as a non-linear correction to the biharmonic operator Ω. This contribution is important at high vibration amplitudes of the membrane, but becomes less and less relevant as the energy of the system is dissipated. In the light of these considerations, it is natural to choose Ω as preconditioner, and to feed it to the *pcg* algorithm in the form of a (possibly incomplete) Cholesky factorization [24].

A schematic outline of the algorithm presented above is given in Fig. 4.

$w_{j=0} \leftarrow w^n$; ▷ initial guess for $w_{j=0}^{n+1}$ is w^n
for $j = 0 \rightarrow maxiter - 1$ **do**
  $\bar{a}_j \leftarrow -\alpha\mathcal{I}\mathcal{J}w_j + a$ ▷ calculate $\bar{a}_j$
  $temp \leftarrow b - L_1\bar{a}_j$
  $(\Omega + L_2L_1)\Phi_{j+1} = temp$ ▷ solve lin sys (35a)
  $w_{j+1} \leftarrow \bar{a}_j + L_1\Phi_{j+1}$ ▷ update w_{j+1}
end for
$\Phi^{n+1} \leftarrow \Phi_{maxiter}$ ▷ update Φ^{n+1}
$w^{n+1} \leftarrow w_{maxiter}$ ▷ update w^{n+1}

Figure 4. Pseudocode for the iterative method described in Sec. 4.2.

5. RESULTS

5.1 Non-linear Membrane

The behaviour of a non-linear system can be dramatically different from that of a linear one. The principle of superposition no longer holds, energy exchanges between different modes are allowed, and the system can ultimately exhibit chaotic behaviour [4].

One among typical non-linear phenomena is the pitch glide effect; an example of this is provided in Fig. 5, where an unnaturally high excitation is used for illustration purposes.

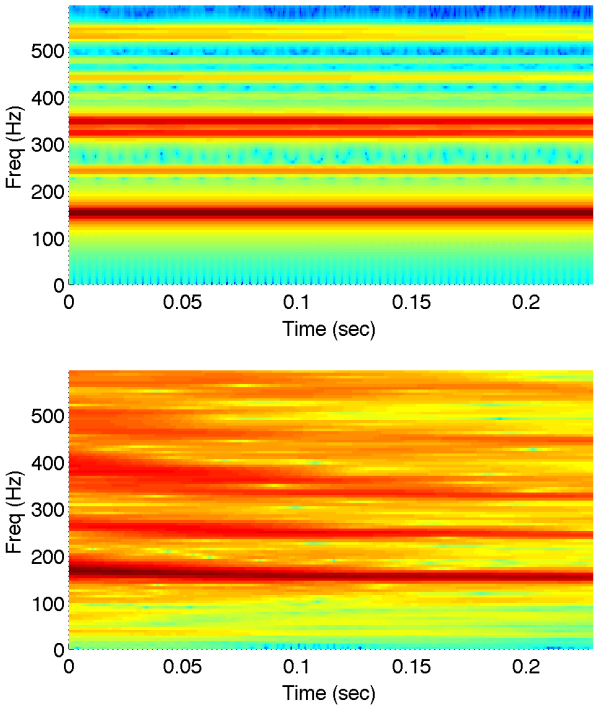


Figure 5. Spectrograms for sound outputs from a virtual tom-tom ($R = 20$ cm) at low (up) and high (down) striking amplitudes. A pitch glide can be seen at high amplitudes of excitation.

5.2 3D Acoustic Field

With an FDTD simulation, one has access to the entire state of the system and, in particular, to the full 3D acous-

tic field. Figure 6 shows, as an example, some snapshots of a cross section of the acoustic field after a short raised cosine excitation (0.3 ms) on the batter membrane. The propagating pressure waves are clearly visible.

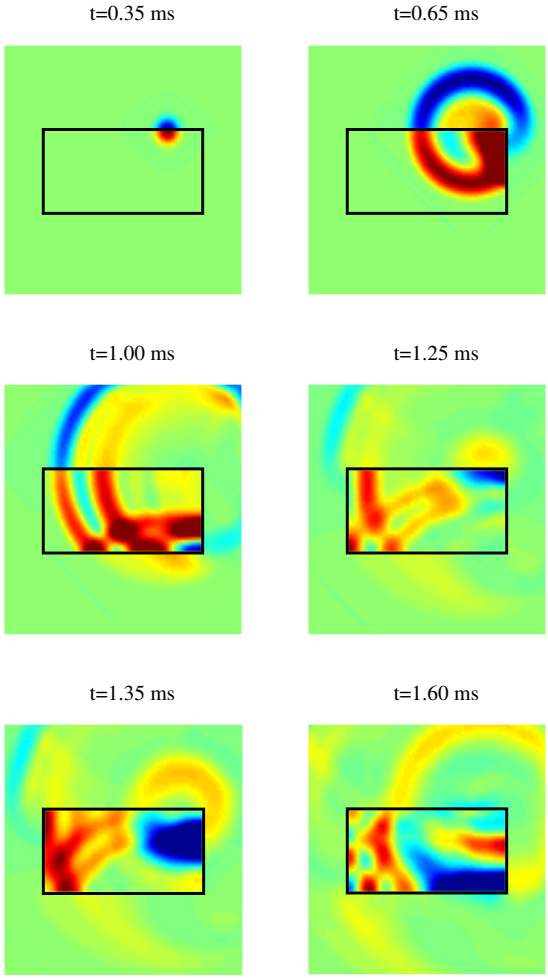


Figure 6. Snapshots of the cross section of the acoustic field after a strike on the batter membrane of the drum, at times as indicated.

5.3 Linear System

In this section we compare different methods for the solution of the linear system described in Section 4.2, the exact solver (31) and the recursive approach (34) based on the reduced system (35). The latter, in particular, can be tackled by means of an exact method or an iterative one. For this second case, one possibility is to use *pcg* solvers, with the Cholesky factorization of Ω as preconditioner and single precision tolerance.

Table 2 shows the computation times at different sample rates for the three methods described. The difference between exact and iterative methods is apparent. This result is even more striking if we think that the solution of the reduced linear system is performed many times (fewer than 10, in this case) inside the recursion.

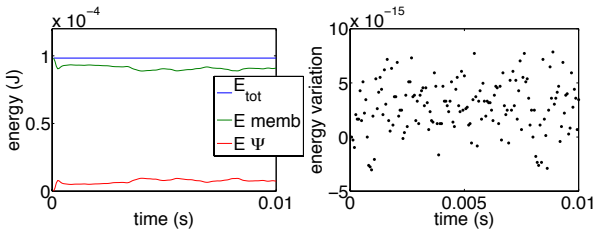
sample rate	dir	iter+dir	iter+pcg
8 kHz	87 s	15 s	8 s
16 kHz	802 s	206 s	102 s

Table 2. Comparison between different approaches to the linear system solution at different sample rates (dir=direct method applied to (31), iter+dir=iterative method (Fig. 4) with exact solution of (35a), iter+pcg=iterative method (Fig. 4) with iterative solution of (35a) using *pcg*.) The computation time refers to 0.01 s of output and for a drum with $R = 0.20$ cm.

5.4 Energy Conservation

In a test simulation with no losses in the membrane equations, and with perfectly reflecting conditions over the boundary $\partial\mathcal{V}$, we can check that the numerical energy of the system be conserved to double precision, even with an iterative solution of the linear system.

Figure 7 shows numerical energy conservation for the system, with energy variations $(\mathfrak{h}^n - \mathfrak{h}^0)/\mathfrak{h}^0$ of the order of machine accuracy.



(a) Energy (b) Energy variations

Figure 7. The energy of the system is shown in (a): total energy remains constant, while there is an energy exchange between the membranes and the air. Normalized energy variations in (b) are of the order of machine accuracy.

5.5 Sound examples

Sound examples produced using the model described above can be found at

`http://www2.ph.ed.ac.uk/~s1164558`

6. FINAL REMARKS

In this paper, we have addressed one of the questions raised in [12] by including a fuller non-linear model in the equations for the membranes. This seems to have been so far (at least one of) the missing pieces of a more realistic model of the drum, perhaps with regard to the dramatic attack that can be heard at high striking amplitudes. The Berger model adopted in [12], in its simplicity, can produce pitch glide effects, but is not able to render the build-up of high-frequency energy typical of high excitations.

As mentioned before, however, this must be considered as a preliminary study of this system, and additional research will be performed in several directions. First of all, an experimental validation of the parameters used here

must be conducted, especially concerning the viscosity parameters of the membranes. Then, an explicit modeling of the drum shell has not been included in the present model. Its importance in the case of the snare drum has been studied in [7], and it is reasonable to expect that a similar behaviour may be found for bass drums and tom-toms, as well. Again, this is something that will be the object of experimental study.

From the computational point of view, the core of the algorithm presented is the solution of a very large sparse linear system. One of the major ongoing efforts at Edinburgh is towards finding an efficient method for the solution of this system that could eventually be parallelized. Some of the progress made is outlined in the present paper, but it is clear that more work needs to be done.

Acknowledgments

This work was supported by the European Research Council, under grant StG-2011-279068-NESS.

7. REFERENCES

[1] H. Fletcher and I. G. Bassett, "Some experiments with the bass drum," *The Journal of the Acoustical Society of America*, vol. 64, p. 1570, 1978.

[2] S. Dahl, "Spectral changes in the tom-tom related to striking force," *Speech Music and Hearing Quarterly Progress and Status Report*, vol. 38, no. 1, pp. 059–065, 1997.

[3] T. Tolonen, V. Välimäki, and M. Karjalainen, "Modeling of tension modulation nonlinearity in plucked strings," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 3, pp. 300–310, 2000.

[4] A. Chaigne, C. Touzé, and O. Thomas, "Nonlinear vibrations and chaos in gongs and cymbals," *Acoustical science and technology*, vol. 26, no. 5, pp. 403–409, 2005.

[5] R. S. Christian, R. E. Davis, A. Tubis, C. A. Anderson, R. I. Mills, and T. D. Rossing, "Effects of air loading on timpani membrane vibrations," *The Journal of the Acoustical Society of America*, vol. 76, p. 1336, 1984.

[6] L. Rhaouti, A. Chaigne, and P. Joly, "Time-domain modeling and numerical simulation of a kettledrum," *The Journal of the Acoustical Society of America*, vol. 105, p. 3545, 1999.

[7] T. D. Rossing, I. Bork, H. Zhao, and D. O. Fystrom, "Acoustics of snare drums," *The Journal of the Acoustical Society of America*, vol. 92, p. 84, 1992.

[8] F. Avanzini and R. Marogna, "A modular physically based approach to the sound synthesis of membrane percussion instruments," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 891–902, 2010.

[9] S. Bilbao, "Time domain simulation and sound synthesis for the snare drum," *The Journal of the Acoustical Society of America*, vol. 131, p. 914, 2012.

[10] L. Savioja, D. Manocha, and M. Lin, "Use of GPUs in room acoustic modeling and auralization," in *Proc. Int. Symposium on Room Acoustics*, 2010.

[11] Next Generation Sound Sythesis (NESS) Project. [Online]. Available: http://www.ness-music.eu

[12] S. Bilbao and C. Webb, "Timpani drum synthesis in 3D on GPGPUs," in *Proc. of the 15th Int. Conference on Digital Audio Effects (DAFx-12)*, 2012.

[13] A. H. Nayfeh and D. T. Mook, *Nonlinear oscillations*. New York: John Wiley and Sons, 1979.

[14] N. Fletcher and T. Rossing, *The physics of musical instruments*. Springer Verlag, 1998.

[15] B. Bank, "Physics-based sound synthesis of stringed instruments including geometric nonlinearities," Ph.D. dissertation, Budapest University of Technology and Economics, 2006.

[16] A. Chaigne and C. Lambourg, "Time-domain simulation of damped impacted plates. I. Theory and experiments," *The Journal of the Acoustical Society of America*, vol. 109, p. 1422, 2001.

[17] O. Thomas and S. Bilbao, "Geometrically nonlinear flexural vibrations of plates: In-plane boundary conditions and some symmetry properties," *Journal of Sound and Vibration*, vol. 315, no. 3, pp. 569–590, 2008.

[18] J.-P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," *Journal of computational physics*, vol. 127, no. 2, pp. 363–379, 1996.

[19] B. Engquist and A. Majda, "Absorbing boundary conditions for numerical simulation of waves," *Proceedings of the National Academy of Sciences*, vol. 74, no. 5, pp. 1765–1766, 1977.

[20] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Wiley Publishing, 2009.

[21] B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Time dependent problems and difference methods*. Wiley New York, 1995.

[22] S. Bilbao, "A family of conservative finite difference schemes for the dynamical von Karman plate equations," *Numerical Methods for Partial Differential Equations*, vol. 24, no. 1, pp. 193–216, 2007.

[23] G. Strang, *Linear Algebra and Its Applications*. London: Academic Press, 1976.

[24] Y. Saad, *Iterative methods for sparse linear systems.* Philadelphia: SIAM, 2003.