IMPORTING ARBITRARILY COMPLEX OBJECTS WITHIN A FDTD BASED PREDICTION APPLICATION

Ian Drumm, School of Computing, Science and Engineering, The University of Salford, U.K.

1 INTRODUCTION

This paper will demonstrate the incorporation of arbitrary complex objects within the author's 3D Finite Difference Time Domain (FDTD) based application ('Wave Tank') for the prediction of sound propagation in the presence of frequency dependant absorbing surfaces. The 3D laser scanning, subsequent voxellisation and hence inclusion within FDTD of a mouse figurine will be described. Hence a simulation of sound propagation in the presence of the mouse will be demonstrated.

With ever increasingly capable personal computers numerical prediction methods such as FDTD are becoming more appealing despite their demands on processing and memory. Initially developed for electromagnetic problems [1] FDTD can model broadband sound propagation phenomena [2] including frequency dependant absorption, diffraction and interference by representing the relevant continuous partial differential equations as centre-difference equations in a discrete numerical model. The acoustic implementation considers sound propagation as coupled equations (1) (2).

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\nabla p \qquad (1)$$

$$\frac{\partial p}{\partial t} = -K\nabla \cdot w \qquad (2)$$

where p is pressure, w is particle velocity, ρ is the density of medium and K the bulk modulus. Writing equations (1) and (2) in centre difference forms gives equations (3) (4) (5) and (6) where *i*, *j* and *k* to represent grid locations and *n* represents time steps.

$$w_x^{n+1}\left(i+\tfrac{1}{2},j,k\right) = w_x^{n}\left(i+\tfrac{1}{2},j,k\right) - \frac{\delta t}{\rho \delta x}\left\{p^{n+\frac{1}{2}}(i+1,j,k) - p^{n+\frac{1}{2}}(i,j,k)\right\} \qquad (3)$$

$$w_y^{n+1}\left(i,j+\tfrac{1}{2},k\right) = w_y^{n}\left(i,j+\tfrac{1}{2},k\right) - \frac{\delta t}{\rho \delta y}\left\{p^{n+\frac{1}{2}}(i,j+1,k) - p^{n+\frac{1}{2}}(i,j,k)\right\} \qquad (4)$$

$$w_z^{n+1}\left(i,j,k+\tfrac{1}{2}\right) = w_z^{n}\left(i,j,k+\tfrac{1}{2}\right) - \frac{\delta t}{\rho \delta z}\left\{p^{n+\frac{1}{2}}(i,j,k+1) - p^{n+\frac{1}{2}}(i,j,k)\right\} \qquad (5)$$

$$\begin{aligned} p^{n+\frac{1}{2}}(i,j,k) = p^{n-\frac{1}{2}}(i,j,k) - K\delta t \Bigg\{ & \frac{w_x^n\left(i+\frac{1}{2},j,k\right) - w_x^n\left(i-\frac{1}{2},j,k\right)}{\delta x} \\ & + \frac{w_y^n\left(i,j+\frac{1}{2},k\right) - w_y^n\left(i,j-\frac{1}{2},k\right)}{\delta y} \\ & + \frac{w_z^n\left(i,j,k+\frac{1}{2}\right) - w_z^n\left(i,j,k-\frac{1}{2}\right)}{\delta z} \Bigg\} \end{aligned} \qquad (6)$$

The ½ steps indicate that the pressure and velocity grids are interleaved in a 'staggered grid' arrangement. For every iteration a velocity component is given by the previous velocity and neighbouring pressures and likewise a pressure value is given by previous pressure and neighbouring velocities. To avoid reflections at grid terminations Perfectly Matched Layers (PMLs) [3] are implemented to introduce gradual absorption whilst keeping the impedance constant. By adding or setting pressures within specified grid positions, sources can be emulated. Though to implement sources that are ‘transparent’ (i.e. don’t scatter incident waves yet preserve the desired behaviour) the impulse response of the medium must effectively be removed from the source driving function [4].

2 VOXELLISING THE MOUSE FIGURINE

How to incorporate the mouse figurine into the FDTD grid with a frequency dependant absorbing surface will be described below. However, firstly some notes about the scanning process.

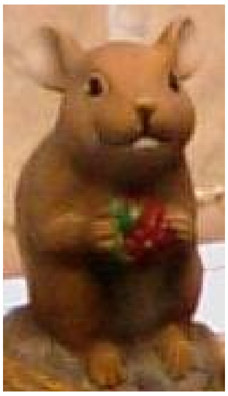


Figure 1: Mouse figurine being scanned

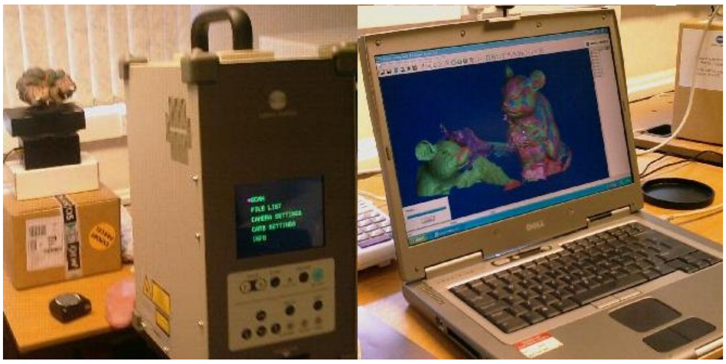


Figure 2: 3D Laser scanner

A Konica Minolta VI-910 3D laser scanner was used. The equipment allowed one to place the mouse figurine on a rotating plate and hence take a series of eight or so scans at varying side angles. A top down scan was also needed. Using Doppler laser interferometry the device is able to generate list of appropriately positioned polygons. Similar points indentified on the scans (e.g. tips of ears, toes and eyes) were subsequently manually registered to allow the scanner’s software to knit together the separate scans and fill any remaining gaps to create a full 3D representation of the mouse. The 3D file was hence saved and converted to X3D, a file format that can be parsed by the author’s ‘Wave Tank’ application. Note the mouse consisted of some 79000 triangles. The author would like to thank Stephen Bowden of Salford University’s Virtual Environments Technology Solutions for his expert help with the scanner equipment.

Although there are some fast voxelisation techniques documented in computer graphics literature the author used a simple ray based approach describe here. Consider a plane given by its vertices $\mathbf{p}_i$ where i indexes into an array of N vertices such that $i = 0,1,2,3...N-1$. The plane can also be described by the equation

$$D = \hat{\mathbf{n}} \cdot \mathbf{p} \qquad (7)$$

where **p** is a point the plane and D is a constant scalar and $\hat{\mathbf{n}}$ is the plane’s unit normal found using the vector cross product

$$\mathbf{n} = (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_0 - \mathbf{p}_1). \qquad (8)$$

Also consider a ray $\mathbf{r}$ such that

$$\mathbf{r} = \mathbf{s} + \mathbf{c}t \qquad (9)$$

where **s** is the start vector position, **c** is the direction vector, and *t* a scalar distance or time value. The ray intersects the plane at

$$t = \frac{D - \hat{\mathbf{n}} \cdot \mathbf{s}}{\hat{\mathbf{n}} \cdot \mathbf{c}} \qquad (10)$$

This gives the point of intersection $\mathbf{r}$ which can be tested as to whether it's within the plane's boundary by considering the normal vectors $\mathbf{n}_i$ formed with respect to $\mathbf{r}$ and all adjacent plane vertices $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ such that

$$\mathbf{n}_i = (\mathbf{p}_i - \mathbf{r}) \times (\mathbf{p}_{i+1} - \mathbf{r}) \qquad (11)$$

If the sense of all dot products

$$dp = \hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_i \qquad (12)$$

are consistent then the point is within the plane boundary.

For each empty FDTD grid element found to correspond to a ray/plane intersection; parameters are set to indicate appropriate reflecting properties.

Figure 3 shows an example of such a ray fired at a plane.

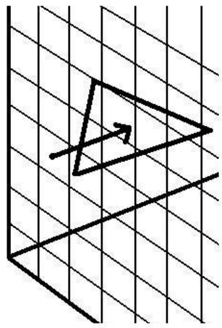


Figure 3: A ray is fired at plane to test if it corresponds with a voxel of the grid

3 EMULATING FREQUENCY DEPENDANT ABSORPTION

After voxelization a key challenge is to model frequency dependant reflection from surfaces that are potentially acoustically non-rigid. The author's approach to emulating frequency dependant absorption will be introduced here. Essentially octave banded absorption coefficients are used to specify the magnitude response of filters. These filter's IIR coefficients are hence computed via a quasi-netwon based optimisation technique. An assumption was made that a velocity component at a surface element can be given as a function of the history of normal velocity components incident and the history previous outputs from this function, for example…

$$w = f_n\left(n_x \cdot w_x^n, n_x \cdot w_x^{n-1}, n_x \cdot w_x^{n-2} ..., f_{n-1}(...), f_{n-2}(...),..\right) \quad (13)$$

where w_x^n is given by equation 3 and n_x is the x component of the surface's absolute unit normal. The output w velocity becomes an input argument to equation 6 hence the approach suggested is easy to integrate into the FDTD iterative cycle. The equation 13 is in effect the difference equation of an IIR filter which could written as

$$w = \left[h_0 + \sum_{i=1}^{N} b_i w_{n-i} - \sum_{j=1}^{N} a_j f_{n-j} \right] \qquad (14)$$

The filter coefficients $h_0, a_1, a_2, ...a_N \quad b_0, b_1, b_2, ...b_N$ can be found from the plane's octave banded absorption coefficients $\alpha(\omega)$ where

$$\omega = {}^{125}\!/_{2\pi}, {}^{250}\!/_{2\pi}, ..., {}^{4000}\!/_{2\pi}$$

The author's implementation considers transmission coefficients such that

$$H_d(\omega) = 1 - \sqrt{1 - \alpha(\omega)} \qquad (15)$$

is the desired magnitude response of a filter.

A least pth norm based filter design approach was hence employed to find the desired coefficients. The approach employ's Newton method in optimisation to minimise the Lp-norm objective function

$$E(x) = \int \left| H(\omega) - H_d(\omega) \right|^2 d\omega \qquad (16)$$

where an estimated response is given by

$$H(\omega) = h_0 \frac{B}{A} \qquad (17)$$

$$B = 1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_N z^{-n}$$

$$A = 1 + a_1 z^{-1} + b_2 z^{-2} + ... + a_N z^{-n}$$

It can be shown that the gradient of the error functions in equation 16 is

$$\nabla E(x) = 2 \int \mathrm{Re}\left[\bar{e}(\omega) \nabla H(\omega) \right] d\omega \qquad (18)$$

where $\bar{e}(\omega)$ is the conjugate of

$$e(\omega) = H(\omega) - H_d(\omega) \qquad (19)$$

and

$$\nabla H(\omega) = \left[\frac{\partial H}{\partial x_1}, ..., \frac{\partial H}{\partial x_m} \right] \qquad (20)$$

where corresponding derivatives

$$\frac{\partial H}{\partial x_m} = \frac{H(\omega)}{h_0} \{ m = 1 \qquad (21)$$

$$\frac{\partial H}{\partial x_m} = -\frac{H(\omega) z^{-m-1}}{1 + A} \{ 1 < m < N + 1 \qquad (22)$$

$$\frac{\partial H}{\partial x_m} = \frac{H(\omega) z^{-m-1-N}}{1+B} \left\{ N+1 < m < 2N+1 \right. \qquad (23)$$

Similarly the Hessian matrix can be given by

$$\nabla^2 E(x) = 2\int_{\Omega} 2\,\mathrm{Re} \left| e(\omega) \nabla^2 H(\omega) + \left(\nabla \overline{H}\right)\left(\nabla H\right)^T \right| d\omega \quad (24)$$

with component derivatives of the matrix easily evaluated [5].

Writing the desired coefficients in equation 17 as the vector

$$\mathbf{x} = \left[h_0, a_1, a_2 ... a_N, b_1, b_2 ... b_N \right]^T \qquad (25)$$

an iterative scheme (equation 26) can be applied to minimise the error function and hence give filter coefficients approximating towards the desired response.

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha\left(\nabla^2 E + \beta I\right)^{-1} \nabla E \qquad (26)$$

where $0 < \alpha < 1$, $\beta \approx 3$, $k = 0,1,2,3...iterations$

However for reasons of computational efficiency the author later adopted a Quasi-Newton approach that uses an approximation of the inverse Hessian matrix $\mathbf{B_k}$. Here the iterative scheme can be written as

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha_k \boldsymbol{\delta}_\mathbf{k} \qquad (27)$$

where the descent direction

$$\boldsymbol{\delta}_\mathbf{k} = -\mathbf{B_k} \nabla \mathbf{E_k} \qquad (28)$$

The step size α_k is found by a line search to ensure the descent satisfies Wolfe conditions and hence gives sufficient decrease. By considering a change in gradient

$$\boldsymbol{\gamma}_\mathbf{k} = \nabla \mathbf{E_{k+1}} - \nabla \mathbf{E_k} \qquad (29)$$

the Davidon-Flecture-Powell update formula (equation 30) was employed changing $\mathbf{B}$ every iteration from an initial identity matrix assumption.

$$\mathbf{B_{k+1}} = \mathbf{B_k} + \frac{\boldsymbol{\delta}_\mathbf{k} \boldsymbol{\delta}_\mathbf{k}^\mathbf{T}}{\boldsymbol{\gamma}_\mathbf{k}^\mathbf{T} \boldsymbol{\delta}_\mathbf{k}} - \frac{\mathbf{B_k} \boldsymbol{\gamma}_\mathbf{k} \boldsymbol{\gamma}_\mathbf{k}^\mathbf{T} \mathbf{B_k}}{\boldsymbol{\gamma}_\mathbf{k}^\mathbf{T} \mathbf{B_k} \boldsymbol{\gamma}_\mathbf{k}} \qquad (30)$$

An outline of the Quasi Newton approach is given in [6] though readers interested implementing in C++, C#, Java, etc may instead choose efficient third party generic optimisation classes, e.g. reference [7] in conjunction with equations 15 to 23 and 25. Note the optimisation should ensure stable filter designs that produce poles within the unit circle. Given fourth order filters were used the author initially employed Ferrari’s method for every iteration to find roots of the quartic equation 31.

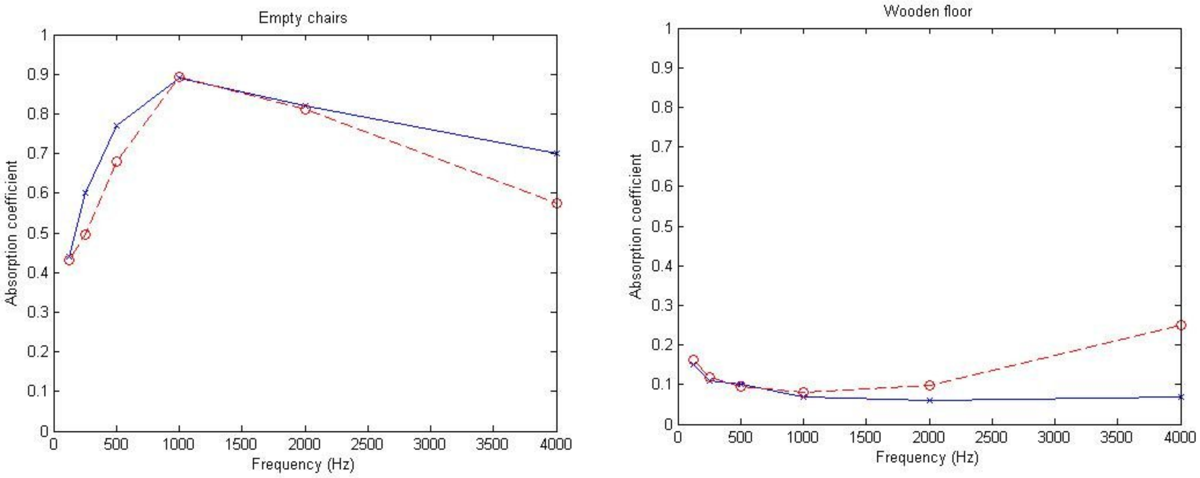
$$1 + a_1 x + a_2 x^2 + a_2 x^3 + a_3 z^4 \qquad (31)$$

If magnitude of a given root was greater than unity then the iteration in equation 27 was rewound with the step size α_k scaled to a lower value. Unfortunately this approach left small deviations from the desired magnitude response near lowest and Nyquist frequencies. Instead a more successful constrained optimisation approach was later adopted where inadequate coefficient constraints were tested for during each iteration step via Rouche's theorem. Essentially the theorem states that if two functions $f(z)$ and $g(z)$ are analytic on a closed contour C and $|g(z)| < |f(z)|$ on C then $f(z)+g(z)$ have the same number of zeros inside C. Applied to the above for denominator coefficients only gives the stability criterion $|\boldsymbol{\delta}_x| \le |\mathbf{x}_k|$. Defining a vector $\mathbf{c}_k$ as

$$c_k = |\boldsymbol{\delta}_x|^2 - |\mathbf{x}_k|^2 \le 0 \qquad (31)$$

If an element of $\mathbf{c}_k > 0$ then a new lower constraint is applied.

To help verify the optimal designed filter approach to modelling surface absorptions within FDTD the author constructed a PML terminated FDTD virtual impedance tube. A range of virtual samples were placed at one end of the tube which was subsequently excited with a set of increasing frequencies. Corresponding standing wave ratios and hence absorption profiles were evaluated. Figure 4 compares the absorption profiles typically specified for empty chairs and a wooden floor with the absorption profiles of corresponding samples tested within the FDTD virtual impedance tube.



Figures 4: Comparison of absorption profiles of predicted in FDTD virtual impedance tube with specified absorption profiles.

The approach is showing great potential for room acoustic prediction where sound propagation occurs within surfaces and the remainder of the filtering propagates away via the PML. For prediction of sound around an object like the mouse the remainder will leak into the object and potentially create unwanted modal behaviour inside that will in turn influence the prediction. For short run predictions this isn't important though flood filling such figurines with voxels whose behaviour are similar to PMLs is a goal of the research.

4 FDTD EMULATED SOUND PROPAGATION AROUND THE MOUSE

By placing a Gaussian pulse source at the front/side of the mouse's face and detectors near the mouse's respective ears; one can demonstrate the technique's potential for say predicting the binaural hearing of the mouse. Figure 5 is a 3D view of the FDTD predicted sound field approaching the mouse. Figure 6 shows the sound wave propagating around the mouse's head in plan and two elevation views. Figure 7 shows the two recorded impulse responses at respective ears. The lower impulse response has the expected delay and frequency dependant attenuation as it reflects off and diffracts around the mouse's head.

Figure 5: A 3D view of mouse in FDTD prediction

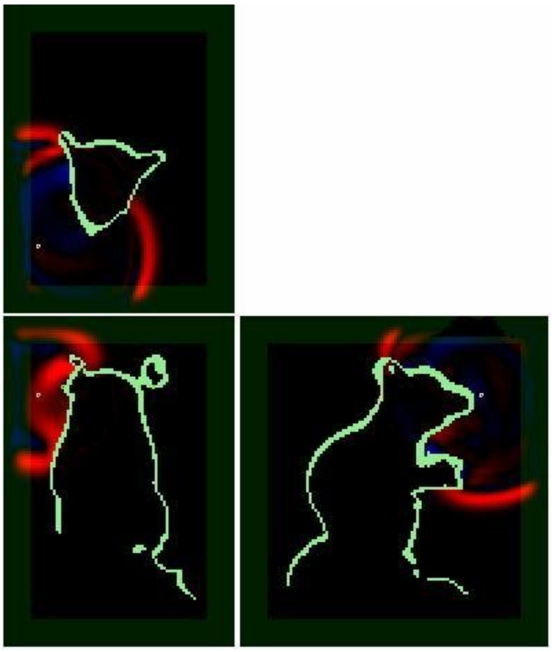


Figure 6: Plan and elevation views of mouse in FDTD based simulation

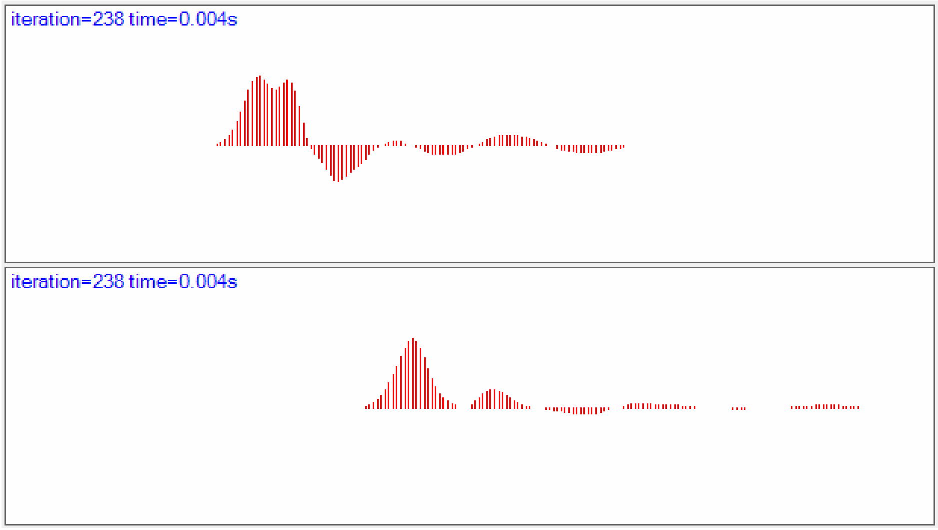


Figure 7: Two impulse responses from FDTD grid. The lower impulse is from a detector at the furthest ear from the source.

The next step will be to do a full evaluation of the validity of a range of predicted results for varying scenarios with reference to real world measurements. There is a need to characterise and mitigate sources of error. For example a key problem with FDTD is the staircase representation of surfaces resulting from a discrete spatial resolution. High spatial resolutions require exponentially more memory potentially beyond the 2GB addressing of typical 32bit operating systems. Newer 64 bit versions of windows offer addressing up to 160GB but the managed XNA API used by the author in application development has yet to catch up. The grid shown above has 84 x 112 x 120 elements including a 10 element depth PML.

5 CONCLUSIONS

Like all numerical methods FDTD has heavy demands on processing and memory and inevitably has errors associated with temporal and spatial resolution including aliasing, anisotropic propagation and dispersion as well as a potential for instability. However with ongoing refinements and optimisations by the FDTD research community and the ever increasing power of personal computers; ways to exploit the technique's great potential to emulate complex scenarios need to be perused and developed. The FDTD approach has great practical potential in predicting sound propagation within rooms, musical instruments, around head/ears, etc. By developing the combination of scanning, voxellisation and the emulation of frequency dependant absorbing surfaces the author hopes to demonstrate one approach to exploiting FDTD for complex real world scenarios.

6 REFERENCES

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Trans. Antennas Propagat. **14**, 302–307 (1966).

[2] J.G. Meloney and K.E. Cummings, "Adaption of FDTD techniques to acoustic modelling",11th Annu. Rev. Prog. Applied Computational Electromagnetics, Vol.2, 724 (1995)

[3] J.P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves'. J. Computational Physics, vol 114, 185-200, 1994.

[4] J.B. Schneider, C.L. Wagner, S.L. Broschat, "Implementation of transparent sources embedded in acoustic finite difference time domain grids", J. Acoust. Soc. Am Vol 133, No. 1, 136-142, January 1998.

[5] W.S.Lu, Y.Chi, R.L.Kirlin,"Least Pth Optimization for the design of 1D filters with arbitrary amplitude and phase responses", ICASSP-93., Volume 3, Issue , 27-30 Apr 1993 Page(s):61 - 64 vol.3

[6] A. Antoniou, Digital Filters: Analysis, Design, and Applications, Second Edition, McGraw-Hill, Inc. 1993.

[7] http://www.alglib.net/