MSc Independent Engineering Scholarship (IES) Proposal

Personal Details

1	Family Name	Durbridge
2	Other Names	Simon
3	Email Address (University)	s.durbridge1@unimail.derby.ac.uk
4	Award Title	MSc Audio engineering
5	Proposed Supervisor	Dr. Adam Hill

NOTES: You can paste material into this form if you wish and expand the sections but the proposal must not exceed 8 pages in length overall, excluding the risk assessment record and the ethics form attached at the end of this proposal. All sections of the risk assessment and ethics form should be completed.

Dissertation Proposal

6	**Proposed Title** Efficient Acoustic Modelling of Large Spaces Using Time Domain Methods

What is the rationale for the proposed IES?

7 **Background**

Introduction

The use of acoustic modelling has expanded from theatre and concert hall design using scale models, through to large format loudspeaker system deployment, environmental noise studies, virtual reality applications, and video game auralization, using innovative tools[1][2]. Many of the packages used to simulate acoustic behaviour, do so using estimation and simplified physical concepts that do not directly simulate wave behaviour or produce direct results.

Time domain methods for solving a simplified wave equation for acoustics may produce direct auralization of sound fields including fundamental acoustic phenomena, as well as providing a visual intuition for acoustic behaviour. These methods are still far from suitable for real-time simulation across the full frequency spectrum, but improvements in method application and parallelisation of processes continues to bring real time acoustic simulation closer to reality[3].

Applying these methods to very large acoustic problems such as arenas, stadia & cathedrals is not trivial, due to in part to the increased scale of computations to undertake. The aim of this dissertation is to explore two methods of solving the acoustic wave equation in a time-stepping fashion, that may significantly reduce total computation time for very large problems. These methods are the Sparse Finite Difference Time Domain Method(SFDTD), and the Pseudospectral Time Domain Method(PSTD). In this proposal, the second order Finite Difference Time Domain Method(FDTD) is introduced, as this is the benchmark method for the dissertation. The two methods of interested are introduced. Finally, some concepts around parallelisation are discussed.

Time Domain Numerical Methods

Time domain numerical methods for acoustic simulation have some benefits over geometric and frequency domain wave methods. Specifically, time domain numerical methods produce direct and contiguous[1] results across the problem space for the whole of the time being simulated. These methods can also be inclusive of room acoustic behaviour that is not inherent in geometric methods, such as room modes, scattering, sound source interaction, moving sources and receivers [4]. This performance is relatively insensitive to the number of sound sources and receivers in the simulation, unlike geometric and frequency domain wave methods that require problem specific differential equations to be solved.

Time domain numerical methods may be considered like building blocks, in that identical equations are solved multiple times across the domain, and it is possible to solve for varying behaviours without changing a whole model fundamentally. An example of this is the work evaluated by Oxnard *et al*[5], implementing different frequency dependent absorbing boundary condition methods in identical models. The flexibility of time domain numerical methods allows for flexible implementation of moving sound sources, moving receivers, ambisonic sound-field encoding[6], multiple domain modelling[7], heterogenous domain modelling[8], viscoelastic fluid modelling[9], crossflows, atmospherics.

The Finite Difference Time Domain Method

However, using time domain numerical methods such as the finite difference time domain (FDTD) method, require a significant number of calculations to be undertaken on large matrices. Applying these methods to very large simulations may not allow for feasible calculation times[2].

The FDTD method was first proposed by Yee[10], and developed as a method for calculating the propagation and material interaction of electromagnetic waves by solving Maxwell's equations[11]. The method as applied by Botteldooren in his seminal work[12], applies a similar methodology as that used by Yee and the computational electromagnetics fraternity, to acoustic modelling. By applying the same concepts behind the linearized Navier-Stokes model to the solving method used by Yee, it is possible to realise a time stepping solution to the simplified wave equation for fluids[9].

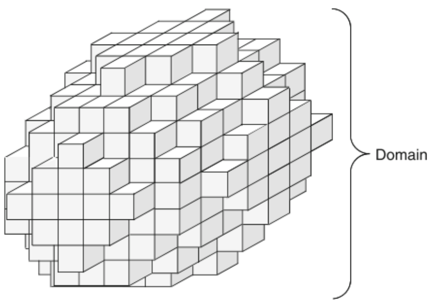


Figure 1Rectiliniarly discretised domain [12]

The explicit second order[2] finite difference time domain method for acoustics is performed as follows:

[1] Audio samples recorded in time
[2] Second order in space, first order in time

- The problem space to be simulated is discretised into a rectilinear grid in all dimensions of interest. The grid is conceptually composed of matrices of singular points, equally spaced at intervals proportional to half the distance of the smallest wavelength of interest and coefficient of CFL stability. Essentially the discretised problem space (known as the Grid for the rest of the document) is sampled at around 10 points per shortest wavelength of interest. The grid is made up of nD+1 matrices, representing pressure values and velocity potentials across the grid.

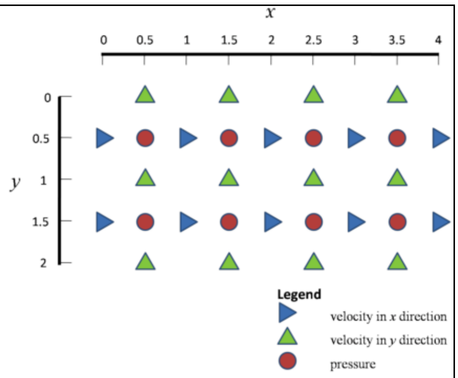


Figure 2Illustration of point discretisation of a domain in an acoustic FDTD simulation [13]

- Once other ‘housekeeping’ tasks are completed, constants, sources and other parameters have been prepared, the main calculating loop is entered. The duration of this loop is proportional to the sampling rate, domain size and predominantly the desired acquisition time of the model. The loop executes as follows (assuming the model is 2 dimensional):
    - Velocity potentials are calculated in turn across the grid. This is done by differentiating across local pressure values to the velocity values being computed. The pressure gradients are summed and multiplied by a constant relative to the speed of sound, grid resolution and density of the medium. This is then subtracted from the stored velocity potential on the grid at the point being calculated.
    - Pressures are calculated in turn across the grid. This is done by carrying out the same process as above. Neighbouring velocity potentials to the pressure point being computed are differentiated, summed, multiplied by the relevant coefficient and then taken from pressure point being calculated.
    - These two calculations are undertaken in a continuing leap-frog style and half steps in time and space relative to the proportions of the computational domain. This is an explicit method of solving the wave partial differential equation(PDE), as one variable is calculated by holding all other variables constant for that computation step.
    - Finally, a source term is imposed at the source location on the pressure grid, and any results are recorded and displayed. The loop is then continually iterated through until the computation is complete.

Stability, and Significantly Large Matrices

This method of solving the acoustic wave equation is simple and flexible, but has a significant drawback. Due to the nature of solving PDEs in this way, there is inherent concern over the stability of the computation. Dispersion error may skew results radically, and sampling theorems such as Nyquist must be satisfied for the simulation to converge correctly. The Courant-Friedrichs-Lewy (CFL) stability condition suggests that for a simulations to converge, the spatial and temporal discretization used must be proportional, and sensitive to the order of the differentiation and so the numerical error[13]. There should be at least 6 if not 10 pressure nodes per the smallest wavelength for the model to converge appropriately. If the highest frequency to be modelled is 20kHz which as a wavelength of 17mm, there must be a node at least every 2.83mm. This is problematic when modelling large spaces, as an arena may be 80m by 120m by 20m and thus have a volume of $192000\mathrm{m}^3$ and would thus require at least 67,844,523 points and ideally 112,941,177 points to be computed stably. With a sampling rate of 48kHz, and assuming the intention of such as simulation were to capture an impulse response over a 10 second period, there would be 54,511,764,705,882 calculations in the computation of the solution alone (upwards of 651 trillion floating point operations excluding moves). When executing such a computation on a single CPU that is also running an operating system and hosting the integrated development environment that is running the simulation, the final solution may take a significant amount of time to compute (anecdotally on the scale of weeks and months).

The Pseudospectral Time Domain Method

The Pseudospectral time domain (PSTD) method for solving PDEs has seen continued development in both acoustics and electromagnetics, even specifically for solving large simulations by utilising the perfectly matches layer (PML) and leveraging periodic boundary conditions[14]. The PSTD method has also been shown to be parallelisable on a general purpose graphical processing unit (GPGPU)[7], [15].

The form of the PSTD method is like that of the FDTD method, with the differentiation occurring in the frequency domain and uniformly across the spatial domain. As with the FDTD method, the domain is discretised into nD+1 matrices of pressure and velocity values. Unlike the FDTD method, 1D sections of the computational domain are operated for the differentiation, as opposed to operating on local points. A discrete Fourier transform (DFT) is applied to an array of pressure or velocity values, to transpose from the spatial domain to the frequency domain. The array is then multiplied with a complex differentiator function that is of equal length to the array. The array is then inversely transformed, and multiplied by a constant that is relevant to the wave equation, as with the FDTD method. The same two step iterating loop as the FDTD method is undertaken.

Applying the differentiation in the frequency domain gives two great benefits over the FDTD method. Numerical methods of simulation have varying orders of error, often relating to the number of points being differentiated over in both space and time. More points in a differentiation often reduces the error of the calculation. A four point FDTD scheme as described above, has less error than a first order method and more error than a fourth order method[16]. The PSTD method differentiates across the span of the domain, and thus has a significantly reduced error[7].

The Sparse Finite Difference Time Domain Method

The Sparse Finite Difference Time Domain (SFDTD) Method is used in optical wavelength electromagnetic simulation, to improve the speed of simulating large and more complex devices (relative to wavelength) such as PIC microcontrollers[17], [18]. This is an adaptation of the traditional FDTD method, and thus follows the same general steps for solving, as outlined above. The difference with SFDTD method is that the computational domain is limited, by reducing the portion of the computational domain that is solved for any iteration by implying a limit or condition.

There are three similar methods of applying the SFDTD concept of reducing the computation domain, including the moving window FDTD (MWFDTD)[19], the Hybrid Ray-FDTD Moving Frame method[20] and the method suggested by Doerr[17] that involves only computing areas local to energy fluctuations above a chosen energy level. These methods differ from the application of solving sparse matrices using the FDTD method, which is a different topic (though could be implemented in the solving of a sparse problem). Unlike the PSTD method, SFDTD has not been applied to acoustic simulation so far in literature.

Parallelism and Matlab

Applying the SFDTD and PSTD methods may show a significant improvement in computation times over the traditional second order FDTD method. These improvements may still be limited by a lack of optimisation, with little control over data throughput and limited processing bandwidth. Maximising process bandwidth has been shown to significantly improve computation speed in a range of PDE solving methods, including FDTD and PSTD[3], [7], [21], [22].

A logical step forward when attempting to solve numerical PDEs quickly is using processor parallelism i.e. implementing the program on several processors or on a processor with multiple cores. An example of this can be found in the literature and example material for Mathworks Parallel Computing Toolbox for Matlab[23]. The parallel computing toolbox provides inherent CPU and GPU parallelising support, enabling functions such as the FFT as well as custom functions to be parallelised by the Matlab IDE with minimal code modification. One demonstration provides an example of how parallelising on a GPU or a CPU are beneficial for different operations. The example shows that decision based tasks such as IF and Switch statement processing are more ideally suited to CPU parallelism, as a CPU is clocked much faster and optimally tuned for executing these forms of commands when compared to a GPU. A GPU is more optimal for recurring numeric computation, and thus is more efficient at performing operations such as FFTs and differentiation on large data sets. This is due to the GPU having many cores, and a potentially large pool of local ram to utilise for the process.

8	**Aims** • To implement time domain acoustic modelling methods on very large problems • To implement an SFDTD algorithm for acoustic simulation • To extend the PSTD method presented in [15] to 3D • To determine if these methods present significant improvements in calculations times when compared to the second order FDTD method	
9	**Objectives** • To implement a pseudo-spectral time domain method engine for 3D simulations • To implement a sparse finite difference time domain method for 3D simulations • To implement a generic second order finite difference time domain method for simulations • To develop a method for indexing large data sets into smaller sets such as the subdomains implemented in [7] • To benchmark both 'fast' methods against the generic method for a simple test problem • To validate the acoustics of the simulations by comparing a generated impulse response with that same room and source analysed using other calculation methods, and the calculated reverberation time of the space	
10	**Plan of work** • Develop a series of Matlab 'kernel' functions for FDTD, SFDTD and PSTD simulation • Prove that the results from these kernels is accurate when implemented on a large simulation, by comparing with results of other calculations for the same domain using other methods • Develop surrounding code to perform simulations using these kernel • Improve the performance of these kernels with the appropriate code profiling and parallelisation tool • Evaluate performance of the algorithms and suggest areas for improvement, relating to solving speed, with a focus on getting closer to real time simulation	

Constraints that may restrict the success of the work

11	**Foreseeable constraints** • Insufficient time available to complete project and & optimise the functions • The nature of parallelism of different functions in Matlab may inhibit optimal performance • Limited understanding of computational physics may produce an inefficient or inaccurate program • Limited control of functions due to using a high-level language and programming interface may restrict the quality of the program

Identifiable risks to the successful completion of the work

12	**Foreseeable risks** • Mathworks could radically change Matlab or the functionality of toolboxes with no warning

	• The code could become corrupted or damaged if not stored or correctly, or version control is not used • The available computer hardware may not be adequate to complete the task using one or all the methods described above

Resources you envisage utilising to help complete the work

13	**Textbooks** Wave and Scattering Methods for Numerical Simulation S. Bilbao Wiley, 2004 Master Handbook of Acoustics Alton, Everest. F McGraw-Hill, 2009
14	**Journals** Journal of the Acoustical Society of America http://asa.scitation.org/journal/jas Journal of the Audio Engineering Society http://www.aes.org/journal/ Applied Acoustics http://www.sciencedirect.com.ezproxy.derby.ac.uk/science/journal/0003682X
15	**Electronic (internet)** Library of the Acoustical Society of America http://asa.scitation.org/ E-Library of the Audio Engineering Society http://www.aes.org/e-lib/ University of Derby E Library http://www.derby.ac.uk/campus/library/ Digital Audio Effects E Library http://www.dafx.de/
16	**Laboratory equipment and software** Computer running MATLAB with: Code Profiler DSP System Toolbox Signal Processing Toolbox Parallel Computing Toolbox Audio Systems Toolbox

Anticipated cost

17	**Please enter all costs. Brief Description or explanation.** (£150 maximum)	Cost
	NVidia GeForce GTX 1080 graphics card – Nvidia graphics card with large number of CUDA cores. The Matlab Parallel Computing Toolbox has inherent CUDA support, allowing for improvement of data processing speed via GPU parallelism with minimal code adaptation.	150.00

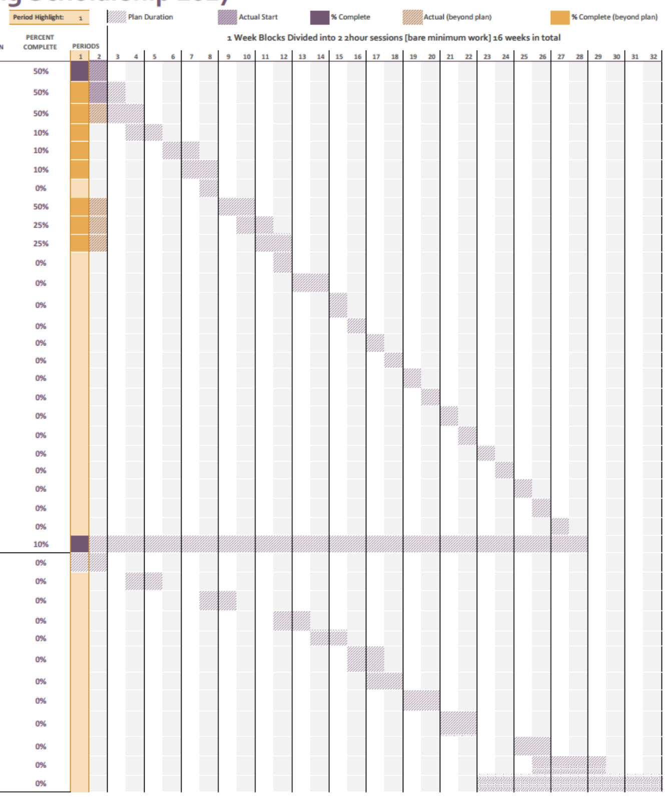
Gantt chart

A readable version of the Gantt chart is available on request, and will be sent to the dissertation supervisor. The chart is split into 16 weeks with 2 sessions per week, to represent the absolute minimum work time required per task (single blocks of between 4 and 8 hours).

S Durbridge Independent Engineering Scholarship 2017

Select a period to highlight at right. A legend describing the charting follows.

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Generic FDTD Algorithm Development 1D	1	2	1	2	50%
Generic FDTD Algorithm Development 2D	2	2	1	2	50%
Generic FDTD Algorithm Development 3D	3	2	1	2	50%
Sparse FDTD Algorithm Development 1D	4	2	1	0	10%
Sparse FDTD Algorithm Development 2D	6	2	1	0	10%
Sparse FDTD Algorithm Development 3D	7	2	1	0	10%
Test SparseFDTD algorithm for speed	8	1	0	0	0%
PTSD Algorithm Development 1D	9	2	1	2	50%
PTSD Algorithm Development 2D	10	2	1	2	25%
PTSD Algorithm Development 3D	11	2	1	2	25%
Test PTSD Algorithm for speed	12	1	0	0	0%
Develop domain handling code	13	2	0	0	0%
Calculate domain response using IM and hand calculated	15	1	0	0	0%
Round 1 Comparitive Testing & Accuracy Confirmation	16	1	0	0	0%
SFDTD Refinement 1	17	1	0	0	0%
PTSD Refinement 1	18	1	0	0	0%
FDTD Refinement 1	19	1	0	0	0%
Round 2 Comparitive Testing & Accuracy Confirmation	20	1	0	0	0%
SFDTD Profiling & Code Refinement	21	1	0	0	0%
PTSD Profiling & Code Refinement	22	1	0	0	0%
Round 3 Comparative Testin & Accuracy Confirmation	23	1	0	0	0%
SFDTD GPU Config	24	1	0	0	0%
PTSD GPU Config	25	1	0	0	0%
FDTD GPU Config	26	1	0	0	0%
Round 4 Comparative Testin & Accuracy Confirmation	27	1	0	0	0%
Develop Matlab UI	1	28	1	0	10%
Report Writing: Problem Intro	1	2	0	0	0%
Report Writing: FDTD Description	4	2	0	0	0%
Report Writing: SFDTD Description	8	2	0	0	0%
Report Writing: PTSD Description	12	2	0	0	0%
Report Writing: Domain & Testing Description	14	2	0	0	0%
Report Writing: Generating results to confirm model Accuracy [BEM, IMS & Analytic]	16	2	0	0	0%
Report Writing: Confirming the Models Accuracy	17	2	0	0	0%
Report Writing: Benchmarking 1	19	2	0	0	0%
Report Writing: Benchmarking 2 Optimisation 1 [REFINEMENT]	21	2	0	0	0%
Report Writing: Benchmarking 3 Optimisation 2 [GPU]	25	2	0	0	0%
Report Writing: Evaluation & Conclusion	26	4	0	0	0%
Report Writing: Intro, Abstract & Error checking	23	12	0	0	0%



Has the IES been agreed with the proposed supervisor? Yes

Explain, if your answer is No………………………………………………………………………………

SUBMISSION: The completed proposal must be submitted electronically by 11.59 pm on Monday 6th Feb 2017.

Record of Risk Assessment

Assessment Reference	***SDIES_01***

Activity assessed	**IES: The Application of Time Domain Acoustical Modelling Methods for Very Large Problems**
Persons who may be affected by the activity	**Simon Durbridge**

SECTION A : Initial Assessment Overview

Consider the activity or work area and identify if any of the hazards listed below are significant.

1	Fall of person		7	Machinery		13	Electricity		19	Substances		25	Drowning	
2	Fall of objects		8	Tools/Equipment		14	Noise or Vibration		20	High Pressure		26	Psychological effects	
3	Tripping/Slipping		9	Mobile work equipment		15	Hot / Cold Surfaces		21	Fire/ explosion		27	Human error	
4	Manual handling operations		10	Mechanical lifting equipment		16	Workstation – layout / space	x	22	Lighting		28	Violence	
5	Repetitive work	**x**	11	Display screen equipment	x	17	Radiation		23	Confined space		29	Peripatetic / lone working	
6	Housekeeping / waste material		12	Sharp objects		18	Temperature / weather		24	Buildings & glazing		30	Other(s)	

SECTION B : Second Stage Assessment

S = Severity

For each hazard identified in Section A complete Section B

L = Likelihood

Hazard No.	Hazard Description	EXISTING CONTROL MEASURES	S	L	RESIDUAL RISK
11	**Prolonged exposure to computer screen**	**Periodic break away from computer**	**1**	**2**	**Tolerable Risk**
5	**RSI Through keyboard and mouse use**	**Periodic break away from computer, with appropriate ancillary/rehab exercises**	**1**	**2**	**Tolerable Risk**
16	**Damage to computer equipment through excess mess and dust**	**Use waste paper/plastic bin, and regularly de-dust computer components**	**1**	**1**	**Trivial risk**

No. of Section B Continuation sheets used:					
Assessor(s)	**S. Durbridge, Dr Adam Hill**			Signed	S Durbridge
Date of Assessment	**6/2/2017**	Revision No.	**1**		

Request for ethical approval for students on taught programmes

Please complete this form and return it to your supervisor as advised in your module handbook. Feedback on your application will be via your supervisor or co-ordinator.

Your Name:	**Simon Durbride**
Student ID:	**100242305**
Unimail address:	s.durbridge1@unimail.derby.ac.uk
Other contact information	
Programme name and code	**MSc Audio Engineering (MH6AB)**
Module name and code	**Independent Engineering Scholarship (7EJ998)**
Name of supervisor	**Dr. Adam Hill**
Name of co-ordinator	**Dr. Ahmad Kharaz**

Title of proposed research study

The Application of Time Domain Acoustical Modelling Methods for Very Large Problems

Supervisor Comments

Are the ethical implications of the proposed research adequately described in this application?	Yes ☒	No ❑	
Does the overall study have low, moderate or high risk in terms of ethical implications?	Low ☒	Moderate ❑	High ❑
Does the study method describe a process of research that is ethically sound?	Yes ☒	No ❑	

Signatures

The information supplied is, to the best of my knowledge and belief, accurate. I clearly understand my obligations and the rights of the participants. I agree to act at all times in accordance with University of Derby Policy and Code of Practice on Research Ethics:
http://www.derby.ac.uk/research/ethics-and-governance/research-ethics-and-governance

Signature of applicant	**Simon Durbridge**
Date of submission by applicant	06/02/2017
Signature of supervisor	
Date of signature by supervisor	

For Committee Use *Reference Number (Subject area initials/year/ID number)*…………………..

Date received……………… Date approved …………… Signed………………………

Comments

1. What is the aim of your study?

- To implement time domain acoustic modelling methods on very large problems
- To implement an SFDTD algorithm for acoustic simulation
- To extend the PSTD method presented in [15] to 3D
- To determine if these methods present significant improvements in calculations times when compared to the second order FDTD method

What are the objectives for your study?

- To implement a pseudo-spectral time domain method engine for 3D simulations
- To implement a sparse finite difference time domain method for 3D simulations
- To implement a generic second order finite difference time domain method for simulations
- To develop a method for indexing large data sets into smaller sets such as the subdomains implemented in [7]
- To benchmark both 'fast' methods against the generic method for a simple test problem
- To validate the acoustics of the simulations by comparing a generated impulse response with that same room and source analysed using other calculation methods, and the calculated reverberation time of the space

2. Explain the rationale for this study (refer to relevant research literature in your response).

Introduction

The use of acoustic modelling has expanded from theatre and concert hall design using scale models, through to large format loudspeaker system deployment, environmental noise studies, virtual reality applications, and video game auralization, using innovative tools[1][2]. Many of the packages used to simulate acoustic behaviour, do so using estimation and simplified physical concepts that do not directly simulate wave behaviour or produce direct results.

Time domain methods for solving a simplified wave equation for acoustics may produce direct auralization of sound fields including fundamental acoustic phenomena, as well as providing a visual intuition for acoustic behaviour. These methods are still far from suitable for real-time simulation across the full frequency spectrum, but improvements in method application and parallelisation of processes continues to bring real time acoustic simulation closer to reality[3].

Applying these methods to very large acoustic problems such as arenas, stadia & cathedrals is not trivial, due to in part to the increased scale of computations to undertake. The aim of this dissertation is to explore two methods of solving the acoustic wave equation in a time-stepping fashion, that may significantly reduce total computation time for very large problems. These methods are the Sparse Finite Difference Time Domain Method(SFDTD), and the Pseudospectral Time Domain Method(PSTD). In this proposal, the second order Finite Difference Time Domain Method(FDTD) is introduced, as this is the benchmark method for the dissertation. The two methods of interested are introduced. Finally, some concepts around parallelisation are discussed.

Time Domain Numerical Methods

Time domain numerical methods for acoustic simulation have some benefits over geometric and frequency domain wave methods. Specifically, time domain numerical methods produce direct and contiguous[3] results across the problem space for the whole of the time being simulated. These methods can also be inclusive of room acoustic behaviour that is not inherent in geometric methods, such as room modes, scattering, sound source interaction, moving sources and receivers [4]. This performance is relatively insensitive to the number of sound sources and receivers in the simulation, unlike geometric and frequency domain wave methods that require problem specific differential equations to be solved.

[3] Audio samples recorded in time

Time domain numerical methods may be considered like building blocks, in that identical equations are solved multiple times across the domain, and it is possible to solve for varying behaviours without changing a whole model fundamentally. An example of this is the work evaluated by Oxnard *et al*[5], implementing different frequency dependent absorbing boundary condition methods in identical models. The flexibility of time domain numerical methods allows for flexible implementation of moving sound sources, moving receivers, ambisonic sound-field encoding[6], multiple domain modelling[7], heterogenous domain modelling[8], viscoelastic fluid modelling[9], crossflows, atmospherics.

The Finite Difference Time Domain Method

However, using time domain numerical methods such as the finite difference time domain (FDTD) method, require a significant number of calculations to be undertaken on large matrices. Applying these methods to very large simulations may not allow for feasible calculation times[2].

The FDTD method was first proposed by Yee[10], and developed as a method for calculating the propagation and material interaction of electromagnetic waves by solving Maxwell's equations[11]. The method as applied by Botteldooren in his seminal work[12], applies a similar methodology as that used by Yee and the computational electromagnetics fraternity, to acoustic modelling. By applying the same concepts behind the linearized Navier-Stokes model to the solving method used by Yee, it is possible to realise a time stepping solution to the simplified wave equation for fluids[9].

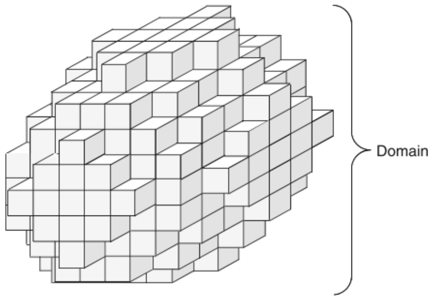


Figure 3Rectiliniarly discretised domain [12]

The explicit second order[4] finite difference time domain method for acoustics is performed as follows:

- The problem space to be simulated is discretised into a rectilinear grid in all dimensions of interest. The grid is conceptually composed of matrices of singular points, equally spaced at intervals proportional to half the distance of the smallest wavelength of interest and coefficient of CFL stability. Essentially the discretised problem space (known as the Grid for the rest of the document) is sampled at around 10 points per shortest wavelength of interest. The grid is made up of nD+1 matrices, representing pressure values and velocity potentials across the grid.

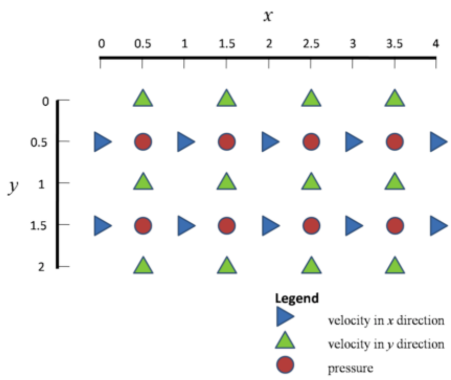


Figure 4Illustration of point discretisation of a domain in an acoustic FDTD simulation [13]

- Once other 'housekeeping' tasks are completed, constants, sources and other parameters have been prepared, the main calculating loop is entered. The duration of this loop is proportional to the sampling rate, domain size and predominantly the desired acquisition time of the model. The loop executes as follows (assuming the model is 2 dimensional):
    - Velocity potentials are calculated in turn across the grid. This is done by differentiating across local pressure values to the velocity values being computed. The pressure gradients are summed and multiplied by a constant relative to the

[4] Second order in space, first order in time

speed of sound, grid resolution and density of the medium. This is then subtracted from the stored velocity potential on the grid at the point being calculated.
  - Pressures are calculated in turn across the grid. This is done by carrying out the same process as above. Neighbouring velocity potentials to the pressure point being computed are differentiated, summed, multiplied by the relevant coefficient and then taken from pressure point being calculated.
  - These two calculations are undertaken in a continuing leap-frog style and half steps in time and space relative to the proportions of the computational domain. This is an explicit method of solving the wave partial differential equation(PDE), as one variable is calculated by holding all other variables constant for that computation step.
  - Finally, a source term is imposed at the source location on the pressure grid, and any results are recorded and displayed. The loop is then continually iterated through until the computation is complete.

Stability, and Significantly Large Matrices

This method of solving the acoustic wave equation is simple and flexible, but has a significant drawback. Due to the nature of solving PDEs in this way, there is inherent concern over the stability of the computation. Dispersion error may skew results radically, and sampling theorems such as Nyquist must be satisfied for the simulation to converge correctly. The Courant-Friedrichs-Lewy (CFL) stability condition suggests that for a simulations to converge, the spatial and temporal discretization used must be proportional, and sensitive to the order of the differentiation and so the numerical error[13]. There should be at least 6 if not 10 pressure nodes per the smallest wavelength for the model to converge appropriately. If the highest frequency to be modelled is 20kHz which as a wavelength of 17mm, there must be a node at least every 2.83mm. This is problematic when modelling large spaces, as an arena may be 80m by 120m by 20m and thus have a volume of $192000\text{m}^3$and would thus require at least 67,844,523 points and ideally 112,941,177 points to be computed stably. With a sampling rate of 48kHz, and assuming the intention of such as simulation were to capture an impulse response over a 10 second period, there would be 54,511,764,705,882 calculations in the computation of the solution alone (upwards of 651 trillion floating point operations excluding moves). When executing such a computation on a single CPU that is also running an operating system and hosting the integrated development environment that is running the simulation, the final solution may take a significant amount of time to compute (anecdotally on the scale of weeks and months).

The Pseudospectral Time Domain Method

The Pseudospectral time domain (PSTD) method for solving PDEs has seen continued development in both acoustics and electromagnetics, even specifically for solving large simulations by utilising the perfectly matches layer (PML) and leveraging periodic boundary conditions[14]. The PSTD method has also been shown to be parallelisable on a general purpose graphical processing unit (GPGPU)[7], [15].

The form of the PSTD method is like that of the FDTD method, with the differentiation occurring in the frequency domain and uniformly across the spatial domain. As with the FDTD method, the domain is discretised into nD+1 matrices of pressure and velocity values. Unlike the FDTD method, 1D sections of the computational domain are operated for the differentiation, as opposed to operating on local points. A discrete Fourier transform (DFT) is applied to an array of pressure or velocity values, to transpose from the spatial domain to the frequency domain. The array is then multiplied with a complex differentiator function that is of equal length to the array. The array is then inversely transformed, and multiplied by a constant that is relevant to the wave equation, as with the FDTD method. The same two step iterating loop as the FDTD method is undertaken.

Applying the differentiation in the frequency domain gives two great benefits over the FDTD method. Numerical methods of simulation have varying orders of error, often relating to the number of points being differentiated over in both space and time. More points in a differentiation often reduces the error of the calculation. A four point FDTD scheme as described above, has less error than a first order method and more error than a fourth order method[16]. The PSTD method differentiates across the span of the domain, and thus has a significantly reduced error[7].

The Sparse Finite Difference Time Domain Method

The Sparse Finite Difference Time Domain (SFDTD) Method is used in optical wavelength electromagnetic simulation, to improve the speed of simulating large and more complex devices (relative to wavelength) such as PIC microcontrollers[17], [18]. This is an adaptation of the traditional FDTD method, and thus follows the same general steps for solving, as outlined above. The difference with SFDTD method is that the computational domain is limited, by reducing the portion of the computational domain that is solved for any iteration by implying a limit or condition.

There are three similar methods of applying the SFDTD concept of reducing the computation domain, including the moving window FDTD (MWFDTD)[19], the Hybrid Ray-FDTD Moving Frame method[20] and the method suggested by Doerr[17] that involves only computing areas local to energy fluctuations above a chosen energy level. These methods differ from the application of solving sparse matrices using the FDTD method, which is a different topic (though could be implemented in the solving of a sparse problem). Unlike the PSTD method, SFDTD has not been applied to acoustic simulation so far in literature.

Parallelism and Matlab

Applying the SFDTD and PSTD methods may show a significant improvement in computation times over the traditional second order FDTD method. These improvements may still be limited by a lack of optimisation, with little control over data throughput and limited processing bandwidth. Maximising process bandwidth has been shown to significantly improve computation speed in a range of PDE solving methods, including FDTD and PSTD[3], [7], [21], [22].

A logical step forward when attempting to solve numerical PDEs quickly is using processor parallelism i.e. implementing the program on several processors or on a processor with multiple cores. An example of this can be found in the literature and example material for Mathworks Parallel Computing Toolbox for Matlab[23]. The parallel computing toolbox provides inherent CPU and GPU parallelising support, enabling functions such as the FFT as well as custom functions to be parallelised by the Matlab IDE with minimal code modification. One demonstration provides an example of how parallelising on a GPU or a CPU are beneficial for different operations. The example shows that decision based tasks such as IF and Switch statement processing are more ideally suited to CPU parallelism, as a CPU is clocked much faster and optimally tuned for executing these forms of commands when compared to a GPU. A GPU is more optimal for recurring numeric computation, and thus is more efficient at performing operations such as FFTs and differentiation on large data sets. This is due to the GPU having many cores, and a potentially large pool of local ram to utilise for the process.

3. Provide an outline of study design and methods. • Develop a series of Matlab 'kernel' functions for FDTD, SFDTD and PSTD simulation • Prove that the results from these kernels is accurate when implemented on a large simulation, by comparing with results of other calculations for the same domain using other methods • Develop surrounding code to perform simulations using these kernel • Improve the performance of these kernels with the appropriate code profiling and parallelisation tool • Evaluate performance of the algorithms and suggest areas for improvement, relating to solving speed, with a focus on getting closer to real time simulation
4. Research Ethics **Does the proposed study entail ethical considerations No** **(please delete as appropriate) If you are unsure please seek advice before submitting this form.** **If 'No' provide a statement below to support this position.** **If 'Yes' move on to Question 5.** **Please note: PROPOSALS INVOLVING HUMAN PARTICIPANTS MUST ADDRESS QUESTIONS 5 - 11.**
5. Please provide a detailed description of the study sample, covering selection, sample profile, recruitment and if appropriate, inclusion and exclusion criteria.
6. Are payments or rewards/incentives going to be made to the participants? Yes ☐ No ☐ **If so, please give details below.**
7. Please indicate how you intend to address each of the following ethical considerations in your study. If you consider that they do not relate to your study please say so. **Guidance to completing this section of the form is provided at the end of the document.**
8. Are there any further ethical implications arising from your proposed research? Yes ☐ No ☐ **If your answer was no, please explain why.**
9. Have / do you intend to request ethical approval from any other body/organisation? Yes ☐ No ☐ **If 'Yes' – please give details**
10. What resources will you require? (e.g. psychometric scales, IT equipment, specialised software, access to specialist facilities, such as microbiological containment laboratories).
11. What study materials will you use? (Please give full details here of validated scales, bespoke questionnaires, interview schedules, focus group schedules etc and attach all materials to the application)
Which of the following have you appended to this application? ❑ Focus group questions ❑ Psychometric scales ❑ Self-completion questionnaire ❑ Interview questions ❑ Other debriefing material ❑ Covering letter for participants ❑ Information sheet about your research study ❑ Informed consent forms for participants ❑ Other (please describe)

References

[1] H. Schmalle, D. Noy, S. Feistel, G. Hauser, W. Ahnert, and J. Storyk, "Accurate Acoustic Modeling of Small Rooms," *Audio Eng. Soc.*, 2011.

[2] J. Van Mourik and D. T. Murphy, “Hybrid Acoustic Modelling of Historic Spaces Using Blender,” no. c, 2014.

[3] L. Savioja, “Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics,” *Proc Int Conf Digit. Audio Eff.*, pp. 1–8, 2010.

[4] J. Van Mourik and D. Murphy, “Geometric and wave-based acoustic modelling using Blender,” *AES 49th Int. Conf. Audio Games*, pp. 1–9, 2013.

[5] S. Oxnard, D. O. Brien, J. Van Mourik, D. Murphy, and M. June, “Frequency-Dependent Absorbing Boundary Implementations in 3D Finite Difference Time Domain Room Acoustics Simulations,” pp. 269–274, 2015.

[6] A. Southern, D. T. Murphy, and L. Savioja, “Spatial encoding of finite difference time domain acoustic models for auralization,” *IEEE Trans. Audio, Speech Lang. Process.*, vol. 20, no. 9, pp. 2420–2432, 2012.

[7] M. Hornikx, T. Krijnen, and L. Van Harten, “OpenPSTD: The open source pseudospectral time-domain method for acoustic propagation,” *Comput. Phys. Commun.*, vol. 203, pp. 298–308, 2016.

[8] J. Shin, C. Shin, and H. Calandra, “Laplace-domain waveform modeling and inversion for the 3D acoustic-elastic coupled media,” *J. Appl. Geophys.*, vol. 129, pp. 41–52, 2016.

[9] L. A. Barba, “CFD Python: 12 steps to Navier-Stokes :: Lorena A. Barba Group,” 2014. [Online]. Available: http://lorenabarba.com/blog/cfd-python-12-steps-to-navier-stokes/. [Accessed: 04-Feb-2017].

[10] K. S. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media,” *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, 1966.

[11] A. Taflove, “Reinventing Electromagnetics: Emerging Applications for FD-TD Computation,” *IEEE Computational Science and Engineering*, vol. 2, no. 4. pp. 24–34, 1995.

[12] D. Botteldooren, “Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid,” *Acoust. Soc. Am.*, vol. 95, no. 5, pp. 2313–2319, 1993.

[13] S. Bilbao, *Wave and Scattering Methods for Numerical Simulation*, 1st ed. London: John Wiley & Sons, 2004.

[14] G. Wojcik, B. Fomberg, R. Waag, L. Carcione, J. Mould, L. Nikodym, and T. Driscoll, “Pseudospectral methods for large-scale bioacoustic models,” *1997 IEEE Ultrason. Symp. Proceedings. An Int. Symp. (Cat. No.97CH36118)*, vol. 2, pp. 1501–1506, 1997.

[15] J. A. S. Angus and A. Caunce, “A GPGPU Approach to Improved Acoustic Finite Difference Time Domain Calculations,” *128th Audio Eng. Soc. Conv.*, 2010.

[16] B. Hamilton and S. Bilbao, “Fourth-Order and Optimised Finite Difference Schemes for the 2-D Wave Equation,” *Research.Ed.Ac.Uk*, vol. 2, no. 2, pp. 1–8, 2013.

[17] C. Doerr, “Sparse Finite Difference Time Domain Method,” *IEEE Photonics Technol. Lett.*, vol. 25, no. 23, pp. 1–1, 2013.

[18] C. Doerr, “3D Sparse Finite-Difference Time-Domain Simulation of Silicon Photonic Integrated Circuits,” vol. i, pp. 4–6, 2015.

[19] J. W. Schuster, K. C. Wu, R. R. Ohs, and R. J. Luebbers, “Application of moving window FDTD to predicting path loss over forest covered irregular terrain,” *IEEE Antennas Propag. Soc. Symp. 2004.*, p. 1607–1610 Vol.2, 2004.

[20] B. Fidel, E. Heyman, R. Kastner, and W. Richard, “Hybrid Ray-FDTD Moving Frame Approach to Pulse Propagation,” pp. 1414–1417, 2009.

[21] P. R. Eller, J.-R. C. Cheng, D. G. Albert, and L. Liu, “Performance Improvement of the 2D Finite Difference Time Domain Acoustic Wave

Simulation Using Multiple GPUs," *Proc. 27th Army Sci. Conf.*, no. CO-05, 2010.

[22] P. Macey, J. Wright, and N. Allsopp, "high performance computing network techniques for vibroacoustic analysis: audio applications," *Audio Eng. Soc.*, vol. 883, 1999.

[23] Mathworks, "Parallel Computing Toolbox - MATLAB," *Overview*, 2016. [Online]. Available: https://uk.mathworks.com/products/parallel-computing.html. [Accessed: 05-Feb-2017].

[24] S. Durbridge, "An Introduction of Time Domain Numerical Methods for Faster Acoustic Modelling," Derby, 2016.