



Universitatea  
Transilvania  
din Brașov  
FACULTATEA DE MATEMATICĂ  
ȘI INFORMATICĂ

Programul de studii:  
Informatica aplicata

# Lucrare de licență

## Acustica spațiilor interioare folosind metoda Ray-Tracing

**Autor:**

Lixandru Andreea-Bianca

**Mentori:**

Nilă Costin

Gorobievschi Sebastian

(Siemens Industry Software)

**Coordonator Științific:**

Băicoianu Alexandra

Brașov, 2021

# Cuprins

<b>Scopul lucrării</b>	<b>5</b>
<b>Introducere</b>	<b>5</b>
<b>1 Noțiuni teoretice necesare</b>	<b>8</b>
1.1 Sunetul	8
1.2 Reflexia sunetului	11
1.3 Difracția și interferența sunetului	12
1.4 Funcția de răspuns la impuls	13
1.5 Funcția de răspuns la frecvență	14
1.6 Absorbția sunetului	14
1.7 Simularea binaurală	15
1.8 Tehnica de trasare a razelor (Ray-Tracing)	16
1.9 Convoluția sunetului	18
1.10 Transformata Rapidă Fourier și Inversa Transofrmatei Rapide Fourier	18
<b>2 Tehnologii utilizate</b>	<b>21</b>
2.1 Limbajul de programare C#	22
2.2 Platforma Unity	22
2.3 Platforma Blender	24
2.4 Biblioteca matplotlib	24
2.5 Biblioteca NWaves	25
2.6 Biblioteca XCharts	25
2.7 Biblioteca StandaloneFileBrowser	26
<b>3 Pașii elementari pentru implementarea aplicației</b>	<b>28</b>
3.1 Ghid de utilizare al aplicației	29
3.2 Calcul geometric	31
3.2.1 Sfera lui Fibonacci	31
3.2.2 Propagarea razelor	33
3.2.3 Selecția razelor	33
3.3 Calcul fizic	35
3.3.1 Trecerea în domeniul frecvențelor	35
3.3.2 Calculul intensităților	35
3.3.3 Calculul presiunilor	36
3.3.4 Funcția de răspuns în frecvență și funcția de răspuns la impuls	36
3.3.5 Calculul distanțelor	37
3.3.6 Calculul timpilor	38
3.4 Post-procesarea	38
3.4.1 Convoluția sunetului	38

<b>4</b>	<b>Testarea și analizarea rezultatelor . . . . .</b>	<b>40</b>
4.1	Realizarea experimentelor și analizarea acestora . . . . .	40
4.2	Validarea modelului acustic folosind Simcenter 3D . . . . .	46
<b>5</b>	<b>Concluzii . . . . .</b>	<b>50</b>
5.1	Concluzii generale . . . . .	50
5.2	Concluzii personale . . . . .	50
5.3	Dezvoltări ulterioare . . . . .	51

## ABSTRACT

Această lucrare este organizată în jurul problemei propagării sunetului în contextul spațiilor interioare folosind metoda Ray-Tracing. În zilele noastre, această problemă este adesea întâlnită atât în mediul academic, cât și în situații din viața reală, regăsindu-se în domenii precum industria ingineriei acustice și industria jocurilor. Acest studiu oferă un model acustic care sprijină inginerii să construiască în mod corespunzător spații precum amfiteatre, săli de concerte, biserici, moschee, fabrici și multe altele. Mai mult de atât, fiind un subiect de actualitate, industria jocurilor, care a început să capete un rol foarte important în viețile noastre, a început să folosească diferite tehnici pentru propagarea sunetului pentru a imita cât mai bine realitatea și pentru a îmbunătății calitatea jocurilor. Prin această lucrare se dorește respectarea nevoilor oamenilor de a putea crea încăperi în care sunetul să fie auzit peste tot, de a putea ajuta omul să își mențină sănătatea urechii prin construirea unor încăperi aflate în parametrii acustici, dar și pentru a îmbunătății experiența omului privind calitatea jocurilor. În această lucrare au fost realizate o serie de experimente diferite pentru a determina parametrii adecvați, precum: care ar fi cel mai potrivit material pentru peretei camerei, care ar trebui să fie frecvența maximă a sunetului, care ar trebui să fie dimensiunile camerei, care ar trebui să fie numărul de raze distribuit în încăpere, care ar fi distanța maximă pe care o rază ar trebui să o permită, dar și altele.

# Scopul lucrării

În ultimele decenii, au fost dezvoltate multiple modele ce calculează modul de propagare al undelor acustice în spațiul virtual. Din acest motiv am ales să dezvolt o aplicație software care modelează propagarea sunetului în încăperi folosind metoda Ray-Tracing. Există multiple motive pentru dezvoltarea și îmbunătățirea modelării acustice în încăperi. Această lucrare va prezenta ce presupune implementarea unui model acustic, atât din punct de vedere fizic, cât și din punct de vedere geometric și vizual. Mai mult de atât, va fi prezentată și implementarea modelului de acest tip, dar și rezultatele obținute, însătoare de o serie de experimente realizate, nici un model care promovează propagarea sunetului într-un mediu virtual nu poate să fie 1 la 1 realitatea. Totuși, se încearcă să găsească unui algoritm cât mai fezabil.

Atunci când vorbim despre sunetul din cadrul unei încăperi poate una dintre cele mai importante probleme este să ne dăm seama unde ar trebui să poziționăm sursa audio pentru ca sunetul să fie auzit peste tot și la o calitate cât mai bună. Pentru inginerii acustici propagarea sunetului într-o încăperă este un subiect foarte important, întrucât aceștia au nevoie să știe cum trebuie să realizeze încăperă precum hale, mall-uri, aeroporturi astfel încât acestea să se afle în parametrii acustici pentru a nu deteriora sănătatea urechii omului ce lucrează în medii expuse. De asemenea, atunci când inginerul acustic realizează o biserică sau o catedrală, acesta este nevoit să rezolve problema propagării sunetului astfel încât acesta să fie auzit peste tot și clar.

În ziua de azi industria jocurilor a început să dezvolte foarte mult datorită rolului esențial pe care aceasta îl ocupă în societate. Din acest motiv, au fost realizate o serie de proiecte inovative care să dezvolte și să promoveze importanța propagării sunetului în jocuri și realitatea. În următorul paragraf vom analiza o problemă care ar putea fi rezolvată folosind o soluție precum cea promovată de către mine în această lucrare.

Pe tema propagării sunetului în încăperi au fost realizate o mulțime de studii care să găsească un mod cât mai eficient și relevant de a rezolva această problemă. În următorul paragraf vom analiza o problemă care ar putea fi rezolvată folosind o soluție precum cea promovată de către mine în această lucrare.

Considerând acestea, industria ingineriei acustice și industria jocurilor, am dorit să creez prin această lucrare un model acustic care să simuleze propagarea sunetului în încăperi pentru a veni în ajutorul celor care activează în cele două industrii.

# Introducere

Acustica este știința preocupată de producția, controlul, transmisia, receptia și efectele sunetului. Acest termen provine din limba greacă, de la cuvântul „*akoustos*”. Începând cu originile sale în vîrbițiilor mecanice și al radiației acestor vibratii prin unde mecanice, acustica a cîtă în aproape toate domeniile vieții. A fost esențială în dezvoltarea culturăi prin creezea instrumentelor muzicale.

Originea științei acusticii este atribuită, în general, filosofului grec Pitagora, ale căruia experimente asupra proprietăților corzilor vibrante care produc intervale muzicale plăcute au avut un merit atât de mare încât au condus la un sistem de accordare care îi poartă numele. Aristotel a sugerat corect că o undă sonoră se propagă în aer prin mișcarea aerului - o ipoteză bazată mai mult pe filosofie decât pe fizica experimentală; totuși, el a sugerat în mod incorrect că frecvențele înalte se propagă mai repede decât frecvențele joase - o eroare care a persistat de multe secole [20].

Considerând toate acestea, am să ne dăm seama că omul a acordat permanent o importanță deosebită domeniului acustic. În zilele noastre, ne dăm seama că elaborarea unui model acustic este și astăzi un subiect de mare interes.

Urechea umană este sensibilă la vibratiile aerului cu frecvențe între 20 Hz și 20 kHz și, odată cu vîrbiță, acest interval se restrâne. Modul în care percepem sunetul diferă de la o cîteală altă și acest lucru se întâmplă datorită camerelor care au o anumită dimensiune și formă, sunt construite din diverse materiale și conțin suprafete diferite.

Când discutăm despre o cameră, un aspect foarte important de reținut este unde ar trebui să poziționăm sursele audio și microfoanele pentru a obține cea mai bună calitate a sunetului și pentru a reduce cât mai mult posibil efectul de ecou și de reverberație. În interiorul unei camere, undele sonore lovestesc diferențe suprafete care pot absorbi sau reflecta sunetul. Este foarte important pentru inginerii care proiectează modele acustice să aleagă materialele potrivite pentru a obține rezultate optime. De exemplu, fabricile și halele au multe suprafete metalice care favorizează efectul de ecou. Un fenomen poate fi verificat de inginerii care proiectează spațiile pentru a se asigura că spătări respectă toate standardele, iar sănătatea persoanelor care lucrează în acele spații nu este compromisă.

Un model acustic va implica, în cel mai generic mod, simularea căilor pe care sunetul le parcurge de la sursă la destinație. Cel mai adesea, aceste modele propun rezolvarea integralei Helmholtz-Kirchoff [8] utilizând diverse abordări de calcul, cum ar fi: soluții numerice la ecuațiile de undă, aproximări de frecvență înaltă la ecuația de undă și modele statistice bazate perceptiv. Modelul pe care l-am creat face parte din a doua categorie.

De obicei, termenul de Ray-Tracing este folosit pentru lumină, dar poate fi folosit și pentru sunet, întrucât atunci când avem frecvențe mari, undele sonore au amplitudine foarte mică și deci le putem approxima folosind raze. Din acest motiv, am folosit metoda Ray-Tracing pentru a crea un model acustic.

Această lucrare va presupune elaborarea unui model acustic pentru spațiile interioare

folosind metoda Ray-Tracing, model care va conține patru etape principale: calcularea geometriei încăperii, realizarea calculelor fizice, post-procesarea datelor obținute și realizarea unei interfețe care să se ocupe de vizualizarea simulării acustice și a rezultatelor obținute de către model.

Pentru a putea evalua corectitudinea modelului au fost realizate două încăperi rectangulare și una sferică folosind platforma Unity. În interiorul acestora au fost plasate o serie de microfoane și o sursă audio. De asemenea, studiul propune un GUI (Graphical User Interface) ușor de folosit pentru a seta configurația dorită pentru fiecare încăpere și pentru a putea vizualiza rezultatele obținute de către modelul acustic. Mai mult de atât, ca să putem valida corectitudinea soluției am folosit software-ul Simcenter3D, o platformă de simulare complet integrată pentru modelarea, simularea și analizarea produselor și sistemelor complexe de inginerie.

Modelul acustic propus de această lucrare va presupune parcurgerea unor etape sevențiale, unde output-ul unei etape va reprezenta input-ul următoarei etape. Acesta va conține etapa de calcul geometric ce se va ocupa de simularea razelor și de modul în care acestea vor fi distribuite în încăpere, al doilea pas se va referi la calculele fizice, iar mai apoi va avea loc o etapă de post-procesare a datelor pentru a putea analiza rezultatele fizice obținute pentru fiecare microfon din încăpere.

Ideea acestei lucrări a pornit de la proiectul Triton oferit de Microsoft ce propune rezolvarea problemei propagării sunetului în jocuri și realitate mixtă. Acesta modelează fizic modul în care sunetul se propagă într-o scenă, având în vedere forma și materialele sale. Procedând astfel, modelează automat efecte imersive de propagare a sunetului precum ocluzia și reverberația sunetului. Proiectul Triton este unic în modelarea cu acuratețe a adevărătei fizice a undelor de sunet, inclusiv a difracției. Incubat de peste un deceniu de cercetări concentrate, este o tehnologie testată în luptă, livrată în titluri majore de jocuri precum Gears of War, Sea of Thieves și Borderlands 3. Proiectul Triton modelează fizica reală a undelor de propagare a sunetului prin spații 3D complexe. Sunetele sonore au lungimi de undă de la centimetri la metri, astfel încât efectele de undă trebuie modelate pentru a evita rezultate nenaturale [21].

O altă sursă de inspirație a fost lucrarea lui David Oliva Elorza care vorbește despre modelarea acustică în încăperi folosind metoda Ray-Tracing. Autorul prezintă în studiul său noțiunile teoretice de care are nevoie un cititor specializat atunci când dorește să realizeze un model acustic, propune implementarea acestuia și o serie de evaluări ale modelului. Aceasta discută o introducere generală a principiilor de propagare a sunetului și o discuție despre studiul tehnicii în modelarea acustică a încăperilor [1]. Fiecare capitol propune Elorza în studiul său, această lucrare propune un mod diferit de deosebire, selecție și reducere al razelor.

În lucrarea *Simulare computerizată a acusticii moscheilor și bisericilor bizantine* a lui Christoffer A. Weitze [7] este propus un model destinat simulărilor acustice în moscheele și bisericiile bizantine, pe când în cadrul acestei lucrări este propus un model acustic mult mai generic.

Astfel, în capitolul ce urmează vor fi prezentate noțiunile teoretice necesare pentru a realiza un model acustic, după care va fi un capitol ce prezintă tehnologiile folosite, urmat de prezentarea modelului acustic implementat și o serie de experimente realizate folosind acest model. Penultimul capitol va conține validarea acestuia folosind software-ul Simcenter 3D, urmat de capitolul final unde se vor prezenta concluziile.

# Capitolul 1

## Noțiuni teoretice necesare

### 1.1 Sunetul

Sunetul reprezintă vibrația undelor propagate într-un mediu, fie el gazos, lichid sau solid și prezintă, în multe aspecte, dar nu în toate, comportament similar cu alte mișcări de undă pe care le întâlnim în natură, adică undele de apă și undele de lumină, ale căror fenomene de propagare sunt ușor de observat.

În contextul propagării sunetului, aerul este mediul de interes despre care vom discuta atunci când vorbim despre acustica încăperilor, iar perturbarea reprezintă o alterare a presiunii atmosferice peste și sub valoarea sa medie, care produce o mișcare periodică a moleculelor de aer înapoi și înainte de-a lungul aceleiași direcții în care se propagă unda (unde longitudinale).

Propagarea sunetului prin aer este ilustrată în Figura 1.1. În partea superioară este ilustrată alterarea presiunii atmosferice, în timp ce în partea inferioară este ilustrată mișcarea moleculelor de aer asociate cu propagarea sunetului. Dacă intensitatea sunetului crește, gradientul presiunii crește și, ca urmare, mai multe molecule de aer se află în mișcare.

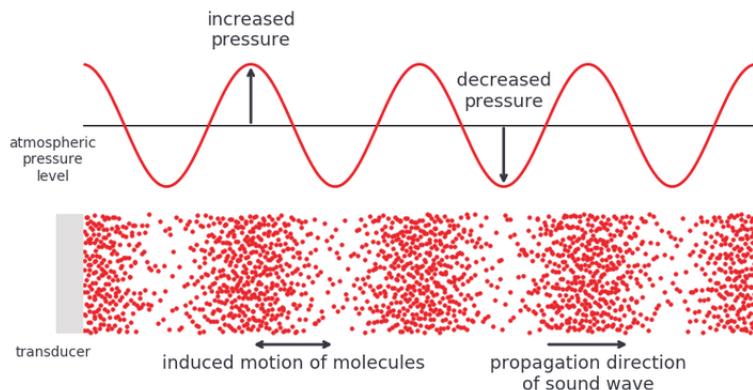


Figura 1.1: Propagarea sunetului prin aer[1]

Un principiu general, stabilit pentru prima dată de Fermat, afirmă că fiecare undă se propagă de la sursă către receptor prin calea cea mai rapidă. Dacă mediu este omogen, precum considerăm că este aerul, atunci viteza sunetului este uniformă prin acest mediu și astfel calea cea mai rapidă devine totodată și cea mai scurtă. Viteza sunetului depinde doar de temperatură, nu și de alte proprietăți precum presiunea și densitatea. Astfel,

putem observa în Tabelul 1.1 cum variază viteza și densitatea sunetului în funcție de temperatură.

Temperatură (°C)	Viteza sunetului $\left(\frac{m}{s}\right)$	Densitatea aerului $\left(\frac{kg}{m^3}\right)$
30	349.02	1.1644
25	346.13	1.1839
20	343.21	1.2041
15	340.27	1.2250
10	337.31	1.2466
5	334.32	1.2690
0	331.30	1.2922

Tabelul 1.1: Efectul temperaturii asupra proprietăților aerului[10]

*Amplitudinea* este valoarea absolută, maximă, a unei cantități care variază periodic. Amplitudinile sunt exprimate fie ca valori instantanee, fie mai ales ca valori de vârf. Aceasta reprezintă fluctuația sau deplasarea unei unde de la valoarea sa medie. În cazul undelor sonore, particulele de aer sunt deplasate, iar această amplitudine a sunetului este exprimată ca intensitatea sunetului. Amplitudinea nu este influențată de frecvență, de lungimea de undă, de perioada de timp sau de viteza sunetului și nici invers.

*Lungimea de undă* este reprezentată de distanța dintre punctele consecutive corespunzătoare ale aceleiași faze de undă, precum două creste adiacente. *Viteza de propagare* ( $c$ ) a undei este dată de lungimea de undă, pe care o vom nota cu  $\lambda$ , fiind distanța parcursă de val între două instanțe de fază egală și de timp,  $T$ , timpul necesar acestei distanțe.

$$c = \lambda/T \quad (1.1)$$

Inversa perioadei este numită *frecvență* ( $f$ ) și indică de câte ori particulele de aer s-au mișcat înapoi și înainte într-o secundă. Frecvența este măsurată în Hertz[Hz].

$$c = \lambda f \quad (1.2)$$

O caracteristică importantă a unei unde sonore este *faza*, care specifică locația unui punct în cadrul unui ciclu de undă al unei forme de undă repetitive. În majoritatea cazurilor, diferențele de fază dintre undele sonore sunt mai importante decât fazele în sine. Diferența de fază dintre două unde sonore cu aceeași frecvență care se deplasează dincolo de o locație fixă este dată de diferența de timp dintre aceleași poziții în cadrul ciclurilor de undă ale celor două sunete, exprimată ca o fracție dintr-un ciclu de undă.

Două unde sonore de aceeași frecvență care sunt perfect aliniate au o diferență de fază de 0 și se spune că sunt „în fază”. Două unde care sunt „în fază” se adaugă pentru a produce o undă sonoră cu o amplitudine egală cu suma amplitudinilor celor două unde.

Dacă una dintre cele două unde sonore cu aceeași frecvență este deplasată cu o jumătate de ciclu față de celalaltă, se spune că undele sonore sunt „defazate”. Două unde sunt „defazate” dacă se anulează reciproc exact când sunt adunate împreună.

Diferența de fază este exprimată ca un unghi, deoarece forma de undă a unui ton pur alcătuită dintr-o singură frecvență poate fi descrisă utilizând funcția sinus trigonometrică.

Cele mai multe sunete sunt mult mai complexe decât o singură frecvență, dar constau în schimb din multe unde sinusoidale diferite la frecvențe diferite. Când mai multe

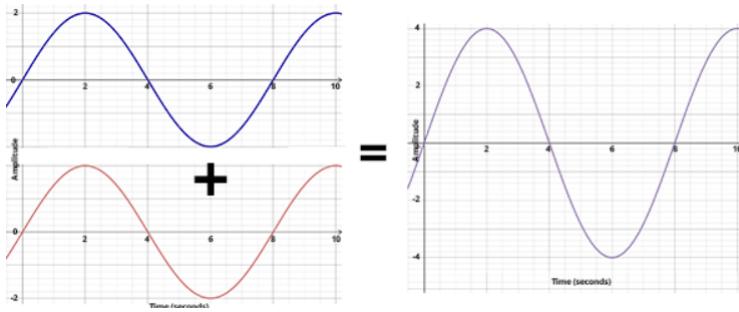


Figura 1.2: Două unde sonore „în fază”

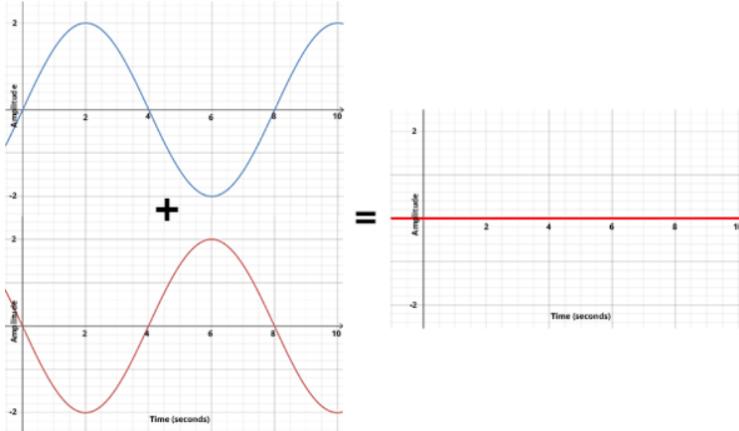


Figura 1.3: Două unde sonore „defazate”

unde sinusoidale se combină pentru a crea un sunet, formele de undă ale tuturor undelor sinusoidale sunt adăugate la fiecare locație de-a lungul formei de undă.

*Energia (W)* poate fi auzită de ființele vii. Sunetul este o undă mecanică și ca atare constă fizic în compresie elastică oscilatorie și în deplasarea oscilatorie a unui fluid. Prin urmare, mediul acționează ca stocare atât pentru energia potențială, cât și pentru energia cinetică. În consecință, energia sonoră dintr-un volum de interes este definită ca suma densităților de energie potențială și cinetică:

$$W = W_{\text{potențială}} + W_{\text{cinetică}} \quad (1.3)$$

*Presiunea acustică* este abaterea presiunii locale față de presiunea atmosferică. În aer, presiunea poate fi măsurată cu ajutorul unui microfon, iar în apă cu ajutorul unui hidrofon. Unitatea de măsură dată de către Sistemul Internațional de Unități de Măsură, SI, pentru presiunea acustică este Pascal (Pa). Formula folosită pentru transformarea intensității în presiune este:

$$p = \sqrt{2I\rho c}, \quad (1.4)$$

unde  $\rho$  este densitatea aerului, iar  $c$  este viteza sunetului.

*Puterea sunetului* este rata la care energia sonoră este emisă, reflectată, transmisă sau recepționată, pe unitate de timp. Unitatea SI pentru puterea sunetului este Watt-ul (W). Pentru o sursă de sunet, spre deosebire de presiunea sonoră, puterea nu este dependentă nici de încăpere, nici de distanță. Presiunea sonoră este o proprietate a câmpului într-un punct din spațiu, în timp ce puterea sonoră este proprietatea unei surse sonore, egală cu puterea totală emisă de acea sursă în toate direcțiile.

*Intensitatea* ( $I$ ) este definită ca putere pe unitate de suprafață purtată de o undă. *Puterea* ( $P$ ) este rata la care energia este transferată de undă. Unitatea de măsură folosă pentru intensitate este  $\frac{W}{m^2}$ , iar formula acesteia este:

$$I = \frac{P}{A} \quad (1.5)$$

*Nivelurile de intensitate* ale sunetului sunt citate în decibeli (dB). Modul în care urechile noastre percep sunetul poate fi descris mai exact prin logaritmul intensității. Nivelul de intensitate ( $\beta$ ) este definit astfel:



$$\beta(dB) = 10 \lg \left( \frac{I}{I_0} \right) \quad (1.6)$$

## 1.2 Reflexia sunetului

Principiile reflexiei pot fi aplicate undelor sonore, care constau din compresii și refracții. Dacă o undă sonoră se deplasează printr-un tub cilindric, în cele din urmă sunetul va ajunge la capătul tubului, acesta reprezentând granița dintre aerul din tub și aerul din afara tubului. La atingerea capătului tubului, unda sonoră va suferi o reflectie parțială (o parte din energia transportată va rămâne în tub și se va deplasa în direcția opusă) și o transmisie parțială (o parte din energia transportată va trece peste graniță, în afara tubului).

Reflexia pe suprafețe poate duce la unul dintre următoarele fenomene: ecou sau reverberație. *Ecou* este o reflexie a sunetului care ajunge la ascultător cu o întârziere față de sunetul direct. Această întârziere este direct proporțională cu distanța suprafeței reflectate de la sursă la ascultător. Undele acustice sunt reflectate de perete sau de alte suprafețe dure, cum ar fi munții. Ecou poate fi auzit atunci când reflexia revine cu o amplitudine și o întârziere suficientă pentru a fi percepută distinct.

*Reverberarea* reprezintă o persistență a sunetului după ce acesta a fost produs. O reverberație este creată atunci când un sunet sau semnal provoacă multiple reflexii care se acumulează și apoi se descompun pe măsură ce sunetul este absorbit de suprafețe. În Figura 1.4, observăm două diagrame ce prezintă pe axa  $Ox$ , timpul, și pe axa  $Oy$ , SPL (Sound Pressure Level) ce ilustrează diferența dintre ecou și reverberație.

Trasarea razelor pentru calcularea unghiurilor de incidentă sau refracție este dată de legea lui Snell prin Ecuația (1.7) (vezi [11]):

$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2} \quad (1.7)$$

unde  $\theta_1, \theta_2$  sunt unghiurile de refracție,  $v_1, v_2$  reprezintă viteza de propagare prin mediu, iar  $n_1, n_2$  sunt indicii de refracție din mediu.

Reflexiile sunetului pe suprafețe nu urmează întotdeauna legile lui Snell. Astfel, nici o reflexie nu este perfect speculară, deci este parțial difuză, iar acest lucru se întâmplă ca o consecință pentru duritatea și dimensiunea suprafeței de coliziune. Când o rază întâlneste o suprafață difuză, se generează un număr aleatoriu în intervalul  $[0, 1]$ . Dacă numărul este mai mic decât un prag ales direcția razei este randomizată pentru a simula difuzia, altfel reflexia este speculară. Acest fenomen poate fi evidențiat prin Figura 1.5. În această lucrare se va folosi reflexia difuză prin randomizare normală.

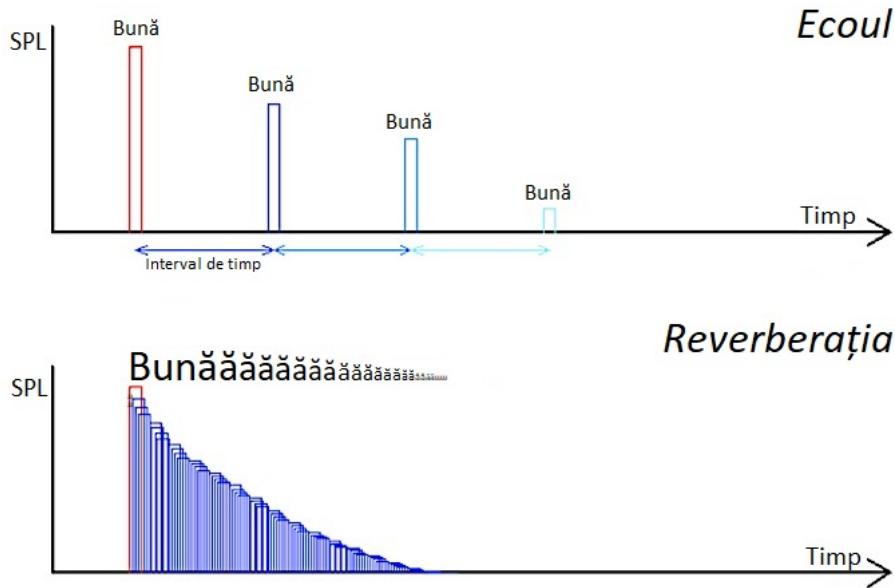


Figura 1.4: Diferența dintre ecou și reverberație

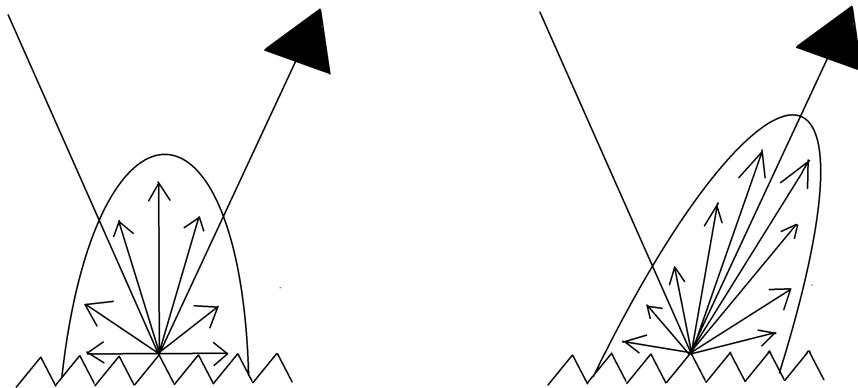


Figura 1.5: Reflexie difuză prin randomizare normală și reflexie difuză prin randomizare ponderată

### 1.3 Difracția și interferența sunetului

Difracția reprezintă schimbarea locală în direcția propagării undelor sonore trecând de marginea unui obstacol. Aceasta este unul dintre cele mai importante fenomene acustice cauzat de natura sunetului, fiind una dintre cele mai complexe probleme de rezolvat. Efectele difracției pot fi împărțite în trei grupe: barieră, margine și deschidere.

Fenomenul de difracție depinde semnificativ de raportul dintre lungimea de undă a sunetului și mărimea obstacolului. Cu cât lungimea de undă este mai mare, cu atât sunetul se intensifică. Când o undă sonoră întâlnește un obstacol, care este mic în raport cu lungimea de undă, valul trece în jurul ei ca și când nu ar exista, formând foarte puțină umbră. Dacă frecvența sunetului este suficient de mare, lungimea de undă este suficient de scurtă și ca urmare se formează o umbră vizibilă.

În Figura 1.6 se pot observa cele trei tipuri de difracții, unde situațiile din partea de sus sunt ideale pentru frecvențele joase, iar cazurile din partea de jos sunt de dorit pentru frecvențele înalte.

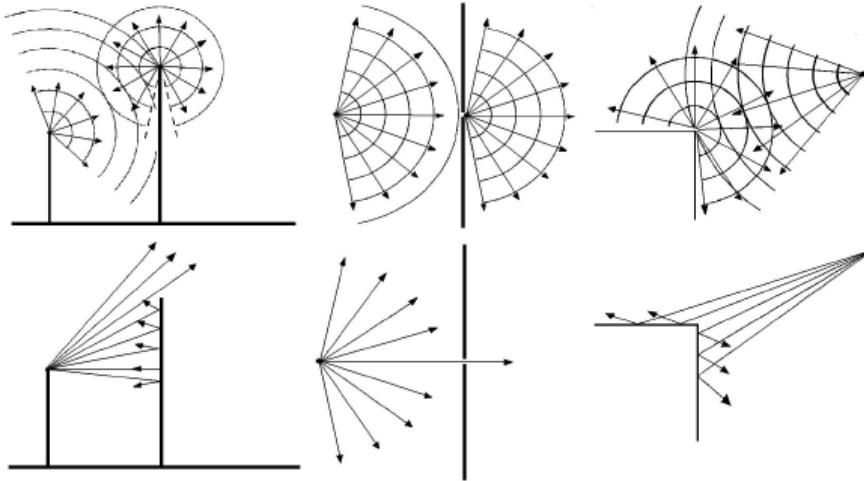


Figura 1.6: Difracție barieră, deschidere, margine[1]

Două unde care călătoresc în același mediu vor interfera una cu cealaltă. Dacă amplitudinile lor se adună, se spune că aceasta este o interferență constructivă. O interferență distructivă se realizează atunci când cele două unde sonore sunt defazate și scad.

În acest studiu, difracția și interferența nu vor fi abordate.

## 1.4 Funcția de răspuns la impuls

Propagarea sunetului de la o sursă audio la un receptor se caracterizează prin funcția de răspuns impuls și informațiile spațiale pe toate căile de propagare posibile. Prima detectare corespunde întotdeauna cu sunetul direct și, după un interval de timp, primește reflexii multiple. Întârzierile lor de timp în ceea ce privește sunetul direct și reflexiile sunt determinate de lungimile căilor parcuse și intensitățile lor de presiune depind de absorbția sunetului prin aer, precum și de caracteristicile de absorbție a suprafețelor implicate în fiecare cale.

Răspunsul la impuls este funcția de răspuns la impuls a unui sistem dinamic atunci când la intrare se aplică o funcție unitară (funcția Dirac). Un impuls este un eveniment sonor foarte puternic și scurt, care este utilizat pentru a mări răspunsul la sunet într-o cameră sau pentru a testa eficiența unui sistem acustic. Un impuls conține toate frecvențele.

O funcție de răspuns la impuls se compune din: sunet direct, prima întârziere, reflecții timpurii și coada reverberantă. Figura 1.7 ilustrează componentele unei funcții de răspuns la impuls.

*Sunetul direct (Direct Sound)* are presiunea acustică ridicată, dar durată scurtă, reprezentând timpul necesar pentru ca sunetul să ajungă la cel primit (ex: ascultător sau microfon).

*Decalaj initial de timp (Initial Time Delay Gap)* reprezintă timpul dintre sunetul direct și primele reflexii și ne spune cât de departe este sursa de sunet. Cu cât decalajul inițial de timp este mai lung, cu atât este mai apropiată sursa de sunet. Cu alte cuvinte, dacă sursa este departe, sunetul direct și primele reflexii vor fi auzite mai apropiate.

*Reflecții timpurii (First Order Reflections sau Early Reflections)* sunt primele auzite și se disting. Este posibil să fie doar câteva într-o cameră simplă dreptunghiulară, dar pot fi mai multe dacă camera este mai complexă. Reflecțiile timpurii indică cât de mare este o cameră.

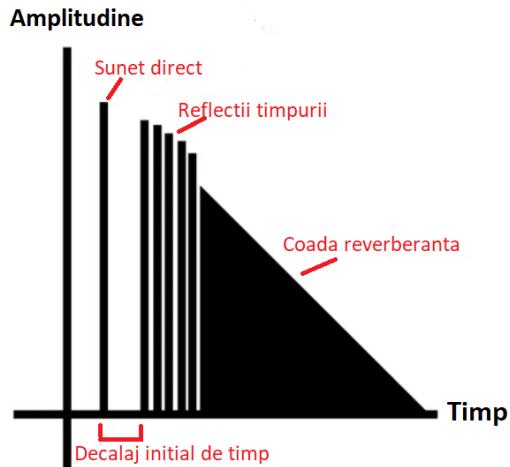


Figura 1.7: Funcția de răspuns la impuls

*Coada reverberantă* constă în reflexii de ordin superior și nu se pot distinge între ele. Pe măsură ce numărul de reflexii crește, undele sonore pierd energie și, în cele din urmă, se descompun. Dezintegrarea reverberantă este adesea liniară atunci când este reprezentată grafic ca mai sus.

## 1.5 Funcția de răspuns la frecvență

Răspunsul în frecvență este măsura cantitativă a spectrului de ieșire al unui sistem sau dispozitiv ca răspuns la un stimул și este utilizat pentru a caracteriza dinamica sistemului. Este o măsură a magnitudinii și a fazei în funcție de frecvență, în comparație cu intrarea. În termeni simpli, dacă o undă sinusoidală este injectată într-un sistem la o frecvență dată, un sistem liniar va răspunde la aceeași frecvență cu o anumită magnitudine și un anumit unghi de fază relativ la intrare.

În contextul unui sistem audio, obiectivul poate fi reproducerea semnalului de intrare fără distorsiuni. Acest lucru necesită o amplitudine de răspuns uniform până la limitarea lățimii de bandă a sistemului, cu semnalul întârziat cu exact aceeași cantitate de timp la toate frecvențele.

Estimarea răspunsului în frecvență pentru un sistem fizic implică, în general, excitarea semnalului cu un semnal de intrare, măsurarea istoricelor de timp de intrare și de ieșire și compararea celor două printr-un proces precum Transformarea Fourier Rapidă (FFT).

Răspunsul în frecvență se caracterizează prin amploarea răspunsului *sistemului*, măsurată, de obicei, în decibeli (dB), și faza, măsurată în radiani sau grade.



## 1.6 Absorbția sunetului

În contextul propagării sunetului în spații închise vom considera două tipuri de absorbții: *absorbția aerului* și *absorbția suprafețelor*.

*Coefficientul de absorbție în aer*, depinde de temperatura atmosferică, de presiunea atmosferică și de frecvență. Absorbția sunetului în acustică este cauzată de caracteristicile de absorbție ale suprafețelor.

Astfel, absorbția este definită ca o disipare a energiei sonore la lovirea unei suprafețe fizice. La fiecare reflexie, o parte  $\alpha$  din energia sau puterea sa este absorbită. Acest factor  $\alpha$  se numește coeficient de absorbție. Absorbția sunetului depinde de unghiul de incidentă.

Un alt mecanism de absorbție a sunetului este *absorbția suprafețelor*, fiind definit ca disiparea energiei sonore la lovirea unei suprafețe fizice. La fiecare reflexie, o parte din energie sau putere este absorbită. Restul de energie care nu este absorbită este fie absorbită de material, fie transmisă mai departe ca sunet reflectat. Formula utilizată în această lucrare este evidențiată prin Ecuatia (1.8).

$$W = W(1 - \alpha) \quad (1.8)$$

unde  $W$  este energia.



## 1.7 Simularea binaurală

*Simularea binaurală* reprezintă o metodă de a realiza semnale binaurale la ambele urechi ale receptorului în spații inexistente prin intermediul unui model. În ultima perioadă, simularea acustică a devenit din ce în ce mai importantă pentru proiectarea spațiilor închise, precum sălile de concerte și de teatru.

Există o mulțime de tehnici de simulare binaurală în combinație cu metode de calcul, de cele mai multe ori tehnici geometrice pentru calcul acustic, precum: metoda sursei de imagine în oglindă, ray-tracing sau metode hibride. Cu toate acestea, genul acesta de tehnici vin la pachet cu creșterea timpului de calcul. Unul dintre cele mai dificile lucruri este directivitatea sursei.

Deși este imposibilă simularea cu exactitate a unui câmp sonor real folosind metode geometrice, lucrările realizate până în prezent pe aceste teme s-au bucurat de un mare succes, iar unele au fost chiar comercializate. O altă provocare în crearea unui astfel de algoritm este dată de coada reverberantă datorită efortului considerabil de timp. Simularea binaurală nu va fi tratată în acest studiu.

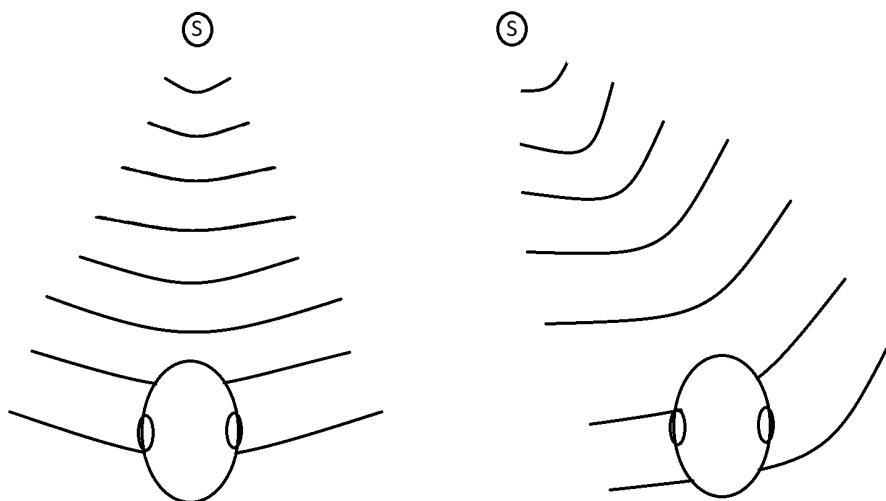


Figura 1.8: Exemplu de simulare binaurală

În Figura 1.8, se poate observa beneficiul unei simulări binaurale. În prima parte a figurii, sunetul ajunge de la sursă (S) la urechile omului în același moment și cu aceeași



intensitate, pe când în cea de-a două urechea stângă decât la urechea dreaptă. În intensități diferite. Urechea stângă va auzi mai puternic sunetul decât urechea dreaptă.

## 1.8 Tehnica de trasare a razelor (Ray-Tracing)

Intensitatea emisă de o sursă este descrisă de un număr finit de raze, care vor fi considerate purtători de intensitate (sau de energie sau de putere). Aceste raze călătoresc prin spațiu la viteza sunetului și sunt reflectate după fiecare coliziune cu limitele camerei. În acest timp, intensitatea lor scade ca o consecință a absorbției aerului și a pereților pe care raza îi intersectează.

Dacă sursa sonoră are caracteristici omnidirectionale, direcțiile razelor sunt create prin distribuții aleatoare omogene. Calea posibilă pentru fiecare rază emisă de la sursă trebuie gasită. Raza este considerată un vector care își schimbă direcția la fiecare reflexie, fiind astfel posibilă determinarea următoarei suprafețe de ciocnire.

De obicei, sursa este împărțită într-un număr mare de piese mici. Metoda deterministă este împărțirea sursei într-o manieră matematică sau geometrică. Piese din sursă ar trebui să fie identice sau cât de mult posibil identice. Punctele selectate aleatoriu pe suprafața unei sfere sursă pot fi reprezentate de trei parametrii: raza, azimutul și elevația [3]. Mai mult decât a propus Jeong Cheol-Ho în studiul său, în această lucrare împărțirea sursei în piese mai mici s-a realizat cu ajutorul sferei lui Fibonacci, algoritm ce urmează să fie descris în capitolele următoare.

Mai departe, aceste puncte reprezintă poziția de start pentru o rază. Fiecare rază poate avea nici una sau mai multe reflexii și, astfel, să fie considerată o rază directă sau indirectă. Fiecare coliziune este reținută și putem observa care a fost traseul pe care fiecare rază l-a parcurs în încăpere de la sursă până la ascultător.

Intensitatea pe care o rază o transmite receptorului este direct dependentă de distanța pe care raza a parcurs-o în încăpere. Intensitatea la un anumit moment se calculează astfel dependent de absorbția aerului, de lungimea traseului parcurs și de proprietățile de absorbție ale peretelui. Coliziunea cu suprafața poate fi considerată speculară, conform legii lui Snell. Tot acest proces trebuie repetat pentru fiecare frecvență.

În Figura 1.9 este ilustrat un mod de reprezentare pentru tehnica Ray-Tracing, unde se poate observa traiectul razelor. Linia continuă reprezintă raza directă, iar liniile punctate semnifică razele indirecte.

S-a demonstrat că algoritmul de Ray-Tracing prezice nivelurile de zgomot cu o precizie foarte bună și este considerat drept unul dintre cele mai elegante metode de reprezentare a reflexiilor. Cu toate acestea, există limitări, excepții și probleme. Există câteva elemente care ar trebui luate în considerare atunci când implementăm sau evaluăm un astfel de algoritm, precum: *intervalul de frecvențe, factori dependenți de frecvență, geometria, numărul de raze*.

În mod normal, singurii factori dependenți de frecvență care pot fi inclusi sunt coeficienții de absorbție și estimarea statistică a proprietăților difuze. O ipoteză de bază în metodele care utilizează raze este că lungimea de undă corespunzătoare celei mai mici frecvențe este mai mică în comparație cu dimensiunile camerei și suprafețele acesteia.

Precum am menționat mai sus, coeficientul de absorbție  $\alpha$  este dependent de unghi. Principalul motiv pentru evitarea măsurătorilor dependente de unghi este că acestea ar trebui să fie foarte precise. Acest lucru fiind destul de dificil, întrucât presupune stocarea

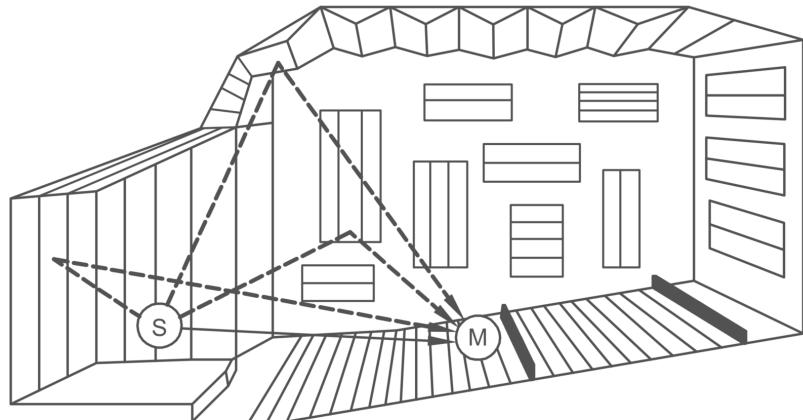


Figura 1.9: Exemplu de ray-tracing

tuturor materialelor de construcție. Prin urmare, a fost acceptată estimarea caracteristicilor de absorbție a unei supafețe printr-o formă care este în medie peste toate unghiurile de incidență.

Erori numerice sunt introduse în rezultate prin utilizarea unui număr limitat de raze, deoarece unghiul dintre raze adiacente rămâne constant, iar această reprezentare devine treptat mai puțin exactă odată cu scăderea numărului de raze. Pentru a atinge criteriile de convergență, numărul de raze trebuie să fie cât mai mare, cu cât numărul de raze este mai mare, cu atât este mai mic unghiul solid pe care îl va acoperi o rază. Acest lucru poate afecta timpul de calcul, dar și un număr prea mic de raze poate genera probleme.

Pentru a putea înțelege mai bine traiectul unei raze vom folosi Figura 1.10. S și M reprezintă sursa și microfonul, iar linia trasată este drumul parcurs de rază până la microfon, drum care presupune 4 reflexii speculare. Modelul de propagare al sunetului ce urmează să fie prezentat în această lucrare va folosi doar reflexii speculare.

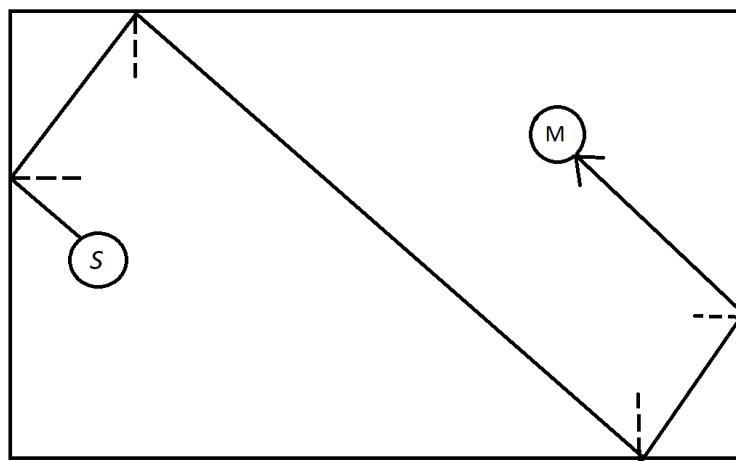


Figura 1.10: Calea unei raze de la sursă la microfon

Atunci când se dorește implementarea unui algoritm de acest gen, există câțiva factori foarte importanți când ne gândim la timpul de calcul, precum: numărul de raze, lungimea maximă pe care o rază o poate avea, numărul maxim de reflexii pe care îl poate avea o rază, chiar și procesorul pe care îl are mașina de pe care lucrăm poate influența semnificativ timpul de calcul.

Astfel, realizarea unui algoritm care modelează acustic spațiile închise presupune două mari etape: realizarea unui model geometric și implementarea unui model fizic. Prima etapă se ocupă de definirea dimensiunilor încăperii, așezarea sursei și a microfoanelor în spațiu, trasarea razelor și determinarea razelor intersectate cu microfoanele. A doua etapă presupune calcularea măsurilor fizice: intensitatea, presiunea, faza, magnitudinea, funcția de răspuns la impuls și funcția de răspuns în frecvență.

## 1.9 Convoluția sunetului

Convoluția în domeniul timpului înseamnă că spectrele sunt multiplicate. Prin „multiplicarea” spectrelor înțelegem că orice frecvență care este puternică în ambele semnale va fi foarte puternică în semnalul convolut și invers orice frecvență care este slabă în ambele semnale de intrare va fi slabă în semnalul de ieșire. Convoluția implică două funcții matematice  $f$  și  $g$  care produc o a treia funcție  $h$  ce reprezintă modul în care forma uneia este modificată de către cealaltă.

Atunci când vorbim despre convoluție, sursa de sunet este numită semnal de intrare, iar fișierul de ieșire este răspunsul la impuls. Fișierul de răspuns la impuls are mereu o lungime fixă.

În practică, o aplicare relativ simplă a convoluției este locul în care avem „funcția de răspunsul la impuls” al unui spațiu. Acest lucru se obține înregistrând o scurtă explozie a unui semnal de bandă largă pe măsură ce este procesat de caracteristicile reverberante ale spațiului. Cu alte cuvinte, a fost procesat de funcția de răspuns în frecvență al spațiului similar cu modul în care ar funcționa acest proces în spațiul real. De fapt, convoluția din acest exemplu este pur și simplu o descriere matematică a ceea ce se întâmplă atunci când orice sunet este „colorat” de spațiul acustic în care apare, ceea ce este deosebit de sănătos vărat pentru toate sunetele din toate spațiile, cu excepția unei camere anecoide. Într-un convolut va apărea, de asemenea, la aceeași distanță ca în înregistrarea originală a sunetului. Dacă convoluționăm un sunet de două ori cu același răspuns la impuls, distanța sa aparentă va fi de două ori mai mare.

Teorema de bază despre domeniul timpului și domeniul frecvenței este că multiplicarea într-un domeniu este echivalentă cu convoluția din celălalt domeniu.

În cele din urmă, există o diferență tehnică între *convoluție directă*, care este un proces foarte lent, dat fiind că fiecare eșantion din fiecare semnal trebuie să fie multiplicat cu fiecare eșantion din celălalt semnal. O variantă mai rapidă de a face acest lucru este folosirea Transformatei Rapide Fourier și folosirea Inversei Transofrmatei Rapide Fourier.

În practică, convoluția se realizează cel mai adesea prin calculul FFT sau analizele spectrale ale fișierelor de intrare și răspunsul la impuls, înmulțind spectrele lor împreună. Aceasta numește *convoluție rapidă*.

## 1.10 Transformata Rapidă Fourier și Inversa Transformatei Rapide Fourier

Transformata Fourier Rapidă (FFT) este un algoritm care calculează transformata directă (DFT) a unei secvențe sau inversa acesteia (IDFT). Analiza Fourier convertește un semnal din domeniul timpului sau al spațiului în domeniul frecvențelor și invers.

Transformata Fourier Discretă (DFT) transformă o secvență de  $N$  numere complexe  $x_n = x_0, x_1, \dots, x_{N-1}$  într-o altă secvență de numere complexe  $X_k = X_0, X_1, \dots, X_{N-1}$  și

are următoarea formulă [2]:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} kn} \quad (1.9)$$

DFT este o operație foarte utilă, dar destul de costisitoare, din acest motiv a apărut FFT, care calculează rapid astfel de transformări factorizând matricea DFT într-un produs de factori rari (în mare parte zero). În acest mod, complexitatea algoritmului este redusă de la  $O(N^2)$  la  $O(N \lg N)$ . Diferența de viteză poate fi enormă, în special pentru seturile de date lungi, unde  $N$  poate de ordinul miilor sau milioanelor. În prezența unei erori de rotunjire, mulți algoritmi FFT sunt mult mai exacti decât evaluarea definitiei DFT direct sau indirect.

Transformata Fourier este utilizată pentru analiza spectrală a seriilor temporale. Cu toate acestea, subiectul procesării statistică a semnalului nu aplică, de obicei, Transformata Fourier asupra semnalului însuși. Chiar dacă un semnal real este într-adevăr tranzitoriu, s-a găsit în practică recomandabil modelarea unui semnal printr-o funcție ale cărei caracteristici sunt constante de-a lungul timpului. Transformata Fourier a unei astfel de funcții nu există în sensul obișnuit și s-a considerat utilă pentru analiza semnalelor.

Reprezentarea în domeniul frecvență presupune descopunerea unui semnal acustic în semnale sinusoidale caracterizate prin frecvență, amplitudine și fază. FFT permite modificarea semnalului prin atenuare/eliminare de frecvențe - filtrare în domeniul frecvență.

Pentru a atinge performanțe există mai mulți algoritmi care calculează FFT, iar acești algoritmi, de obicei, presupun împărțirea polinomului inițial în două polinoame. Polinomul calculat poate fi caracterizat de rădăcinile complexe conjugate prin rădăcini de ordin  $\frac{n}{2}$  a unității. Acest polinom trebuie evaluat la doar  $\frac{n}{2}$  rădăcini ale unității. Una dintre condițiile necesare pentru a atinge aceste performanțe de viteze este ca  $n$  să fie o putere de-a lui 2.

În Figura 3.6, putem observa trei semnale  $x_1, x_2, x_3$  care împreună formează semnalul  $x$  și sunt transformate cu ajutorul Transformatei Rapide Fourier pentru a compune semnalul transformat.

Principala proprietate a Transformatei Fourier este aceea că transformă o funcție de timp,  $x(t)$ , unde  $t$  reprezintă timpul, într-o funcție în frecvență  $\omega$ . Vom defini funcția de răspuns în frecvență astfel:

$$W(j\omega) = \frac{1}{(j\omega)^2 m + j\omega c + k} = \frac{1}{k - m\omega^2 + j\omega c} \quad (1.10)$$

După cum se poate observa, funcția de răspuns în frecvență poate fi rescrisă ca un număr complex de forma:

$$W(j\omega) = Re(\omega) + jIm(\omega) = A(\omega)e^{j\phi\omega} \quad (1.11)$$

unde  $Re(\omega)$  este partea reală,  $Im(\omega)$  este partea imaginară,  $A(\omega)$  este modul și  $\phi$  este faza funcției de transfer de frecvență. Componentele  $Re(\omega)$ ,  $Im(\omega)$ ,  $A(\omega)$  și  $\phi(\omega)$  se numesc caracteristicile funcției de răspuns în frecvență.

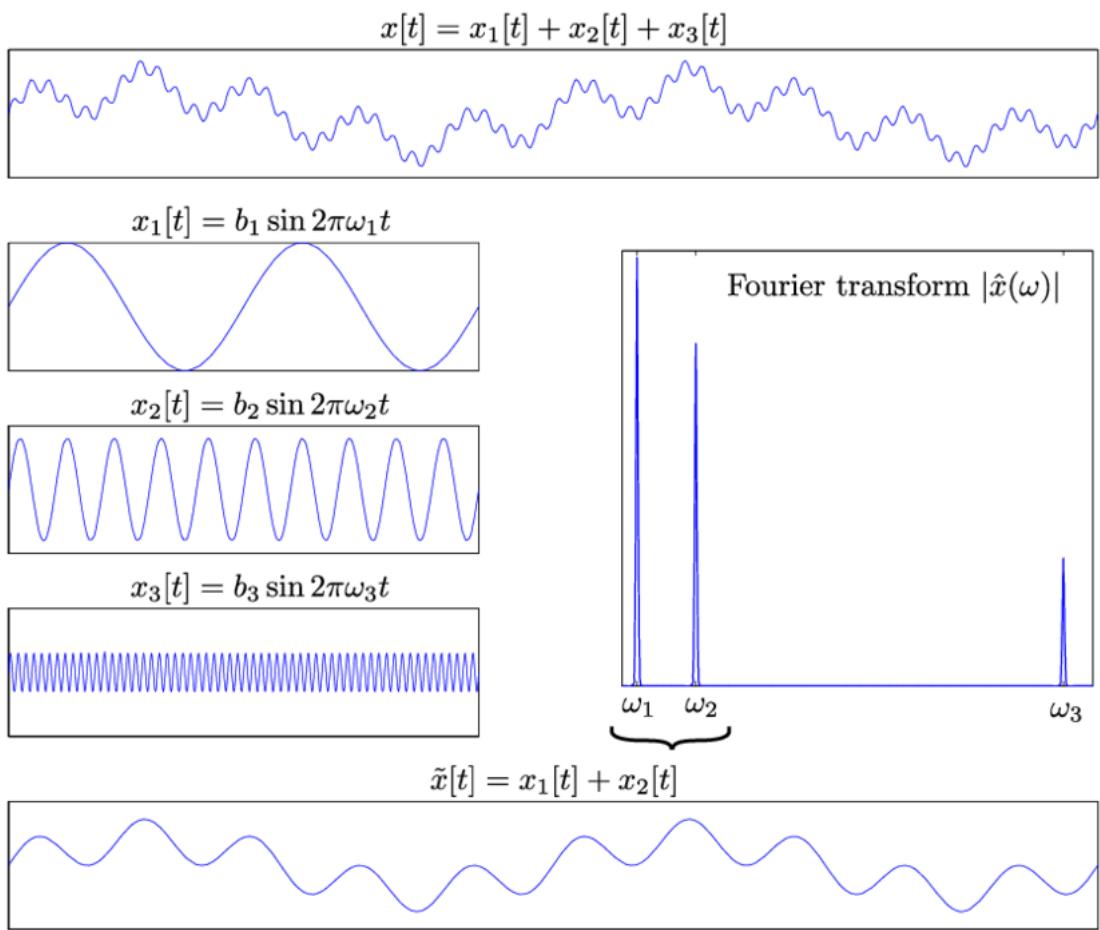


Figura 1.11: Exemplu de transformare al semnalului cu ajutorul Transformatei Rapide Fourier

# Capitolul 2

## Tehnologii utilizate

În zilele noastre calculatorul este folosit în multiple arii și are ca scop soluționarea sau optimizarea unor probleme. Acesta poate rezolva diferite sarcini, deoarece este programabil, adică a fost realizat pentru a putea soluționa orice cerere dată de un program.

Un program reprezintă un set de instrucțiuni pe care calculatorul le îndeplinește pentru a rezolva o problemă. Pe lângă procesul de programare în sine, se regăsesc și alte procese precum testarea, depanarea și mențenanța codului sursă care, asigură astfel o calitate superioară a codului sursă. Aplicația finală trebuie să îndeplinească o serie de proprietăți fundamentale, indiferent de limbajul de programare ales.

O parte din aceste proprietăți sunt:

- *Fiabilitatea*: proprietate care reprezintă corectitudinea programului, adică în ce măsură aplicația îndeplinește scopul pentru care a fost concepută, depinzând de factori externi precum corectitudinea algoritmilor și de cantul de erori care pot apărea în timpul execuției programului. Mai exact, aceasta poate fi văzută ca o probabilitate.
- *Robustetea*: proprietate care definește în ce măsură programul soft reacționează la evenimente mai puțin așteptate cum ar fi accesarea datelor indisponibile sau a unei zone de memorie nealocată, introducere de date eronate, etc.
- *Uzabilitate*: proprietate care vizează în mod direct utilizatorul și se referă la ușurința cu care acesta își poate rezolva problemele prin intermediul aplicației dezvoltate.
- *Portabilitate*: proprietate care definește multitudinea de platforme și sisteme de operare pe care poate rula aplicația dezvoltată. De asemenea, se referă și la ușurința cu care se poate muta codul sursă de pe o platformă pe alta.
- *Menținabilitate*: proprietate care evidențiază ușurința de a modifica aplicația, anume prin adăugare de noi funcționalități pentru a satisface noi cerințe, fixarea problemelor existente sau adaptarea codului la o versiune actuală aplicației.
- *Eficiență/Performanță*: proprietate care măsoară resursele de timp și spațiu de memorie folosite la execuția aplicației. Eficiența programului constă în minimizarea resurselor utilizate. Un alt aspect important este gestionarea corectă a memoriei și utilizarea unor algoritmi eficienți.

În continuare urmează o prezentare detaliată privind tehnologiile folosite în cadrul acestei lucrări.

## 2.1 Limbajul de programare C#

Un limbaj de programare este un limbaj formal care cuprinde un set de instrucțiuni care produc diferite tipuri de ieșire. Limbajele de programare sunt utilizate în programare pentru a implementa algoritmi.

Majoritatea limbajelor de programare constau în instrucțiuni pentru calculatoare. Există mașini programabile care folosesc un set de instrucțiuni specifice, mai degrabă decât limbaje de programare generale.

Limbajul de programare C# este un limbaj de programare imperativ, obiect-orientat, asemănător sintactic cu Java și C++. Acesta a fost creat de Microsoft, inițial în cadrul proiectului .NET, la sfârșitul anilor 90, fiind un concurrent al limbajului Java. Acestea sunt derivate ale limbajului C++. Limbajul C# a fost conceput ca să fie simplu, modern, să aibă un scop general și să fie orientat pe obiecte.

Acet limbaj conține două categorii de tipuri de date: *tipuri valoare* și *tipuri referință*. Prima categorie conține tipuri simple, precum: char, int, float, dar și tipurile enumerare și structură, fiind alocate pe stivă sau inline într-o structură. Cea de-a doua categorie conține tipurile interfață, delegat și tablou. Un aspect important al limbajului de programare C# este faptul că toate tipurile de date sunt derivate direct sau indirect din tipul de date System.Object.

Clasele reprezintă tipuri referință. O clasă poate să moștenească o singură clasă și poate implementa mai multe interfețe. Clasele pot conține constante, câmpuri, metode, proprietăți, evenimente, indexatori, operatori, constructori de instanță, destructori, constructori de clasă, tipuri imbricate. Fiecare membru poate conține un nivel de protecție care controlează gradul de acces la el. Limbajul de programare C# conține următorii modificatori de acces: public, protected, internal, protected internal și private.

Parametrii metodelor permit transmiterea de valori la apel. În general, transmiterea acestora se face prin valoare, ceea ce înseamnă că la apelul unei metode în stiva gestionată de compilator se copiază parametrul actual transmis, iar la revenirea din metodă această valoare va fi pierdută. În cazul variabilelor de tip referință pe stivă se va depune ca valoare adresa de memorie a care este stocat obiectul respectiv, fapt care permite metodei să modifice starea obiectului care se transmite.

Astfel, se pot crea aplicații web prin intermediul ASP.NET, aplicații desktop prin WPF (Windows Presentation Foundation) sau aplicații mobile pe Windows Phone. Common Language Runtime (CLR) gestionează execuția programelor .NET, fiind un Virtual Machine (VM) care rulează Intermediate Language (IL) și oferă multiple servicii, precum gestionarea memoriei, securitate, gestionare de excepții, garbage collector, dar și altele [9].

În cadrul acestui studiu, limbajul de programare a fost folosit pentru a realiza implementarea modelului acustic, iar rezultatele acestuia au fost ilustrate cu ajutorul platformei Unity, dar și cu ajutorul limbajului de programare Python.

## 2.2 Platforma Unity

O platformă (IDE- Integrated Development Environment) este o aplicație software care oferă facilități programatorilor pentru dezvoltarea de soft. O platformă este alcătuită din cel puțin un editor de cod sursă, instrumente de automatizare a construcțiilor și un depanator.

Un singur program în care se realizează dezvoltarea unui soft reprezintă o platformă (IDE). Această platformă oferă multe caracteristici pentru autorizare, modificare, compilare, implementare și depanare a produsului soft. Unele platforme sunt specializate pe un limbaj de programare specific, oferind un set de caracteristici care se potrivesc cu paradigma de programare a acelui limbaj. Cu toate acestea, există multe IDE-uri care suportă mai multe limbi de programare.

Platforma Unity este folosită în general pentru a crea jocuri și poate rula pe mai multe platforme. A fost dezvoltată de Unity Technologies și lansată în 2005 la Apple Inc's Worldwide Developers Conference ca fiind un game engine exclusiv pentru macOS.

Unity este principala platformă mondială pentru crearea și operarea de conținut 3D interactiv, în timp real, oferind instrumente pentru a crea jocuri și pentru a le publica pe o gamă largă de dispozitive. Platforma de bază Unity permite echipelor creative să fie mai productive împreună.

Platforma este capabilă să ofere multe dintre cele mai importante funcții care permit unui joc să funcționeze. Prin acest fapt ne referim la fizică, randare 3D, detectarea coliziunilor. Toate acestea promovează utilizarea unor resurse și tehnologii deja existente, astfel încât dezvoltatorul să nu fie nevoie să înceapă crearea unui joc de la zero, reinventând roata, ci folosind deja lucrurile predefinite.



Aceasta este și mai puternică, întrucât conține o secțiune intitulată „Asset Store”, un loc care permite dezvoltatorilor să își publice creațiile pentru a putea fi folosite și de alți utilizatori ai platformei.

De-a lungul anilor, această platformă s-a dezvoltat reușind astăzi să susțină peste 25 de platforme. Platforma poate fi utilizată pentru a crea jocuri 2D și 3D, realitate virtuală și realitate augmentată. Unity este utilizat nu numai pentru jocuri video, cât și în domeniul filmelor, arhitecturii, ingineriei și construcțiilor [6].

Platforma a fost folosită pentru a:

- modela încaperi de dinamice și dimensiuni
- modela microfoanele și difuzarea audio
- vizualiza individual o rază
- vizualiza toate razele
- difuzarea soluțiilor obținute de solver prin intermediul fișierelor .wav
- realizarea graficelor ce prezintă rezultatele obținute pe microfoane

Mai mult de atât, în acest studiu am folosit metoda *Raycast* din clasa *Physics* pentru a putea detecta coliziunile unei raze cu suprafețele din încăpere. Această metodă are următoarea semnătură:

```
■ public static bool Raycast(Vector3 origin, Vector3 direction, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers, QueryTriggerInteraction queryTriggerInteraction = QueryTriggerInteraction.UseGlobal);
```

unde *origin* reprezintă poziția de start a razei în spațiu, *direction* indică direcția razei, *maxDistance* este distanța maximă pe care ea poate călători și permite detectarea de coliziuni, *layerMask* este folosit pentru a ignora selectiv anumite suprafețe de ciocnire, iar *queryTriggerInteraction* specifică dacă această interogare trebuie să declanșeze o acțiune.



Cu ajutorul platformei, au fost integrate în aplicație bibliotecile: *NWaves*, cu ajutorul căreia se realizează prelucrarea sunetului, *XCharts*, utilizată în realizarea unor grafice menite să prezinte rezultatele obținute de modelul acustic, și *StandaloneFileBrowser*, folosită pentru a încărca fișiere din calculatorul utilizatorului.

## 2.3 Platforma Blender

Blender este o aplicație software de grafică 3D folosită pentru crearea de informații, efecte vizuale, artă, modele de tipărire 3D, grafică de mișcare, aplicații 3D interactive și jocuri pe calculator. Aceasta este cross-platform și permite mapări UV, folosirea materialelor, a shaderelor, a mesh-urilor fără probleme. Mai mult de atât, este o suită de creație 3D gratuită și open-source, care suportă întreagul pipeline 3D.

Platforma a fost dezvoltată de Fundația Blender, o organizație nonprofit înființată în 2002. În anul 2007, a fost creat Institutul Blender ce găzduiește în prezent fundația, având ca scop dezvoltarea continuă a proiectelor creative. Chiar dacă este gratuit, Blender este accesibil și valoros pentru o gamă largă de utilizatori, pornind de la începători și până la profesioniști. Întrucât este un soft actualizat continuu de către utilizatorii avansați, poate fi folosit chiar și de către amatori. Astfel, rămâne o platformă în care totul este despre accesul oamenilor pentru ca aceștia să poată crea orice își doresc.

Blender este o platformă surprinzătoare care acceptă mai mult decât folosirea unor instrumente primare de modelare 3D. Aceasta conține *Grease Pencil* care îți permite să pictezi într-un spațiu 3D folosind pensule 2D, oferind astfel posibilitatea de a crea animații 2D într-un spațiu hibrid. De asemenea, sunt posibile simulările fizice.

Interfața este una intuitivă, ușor de folosit, care permite modificarea și schimbarea ferestrelor în mai multe moduri pentru a crea mai mult spațiu de lucru pentru creator.

În acest studiu, programul a fost folosit pentru a putea realiza încăperea sferică din imagine. Întrucât era nevoie de un soft ce trebuia să permită crearea unei fișier de tipul .obj, să conțină o sferă cu normalele inversate pentru a putea plasa alte elemente în jurul acesteia.

Ca să putem realiza o sferă cu normalele inversate este nevoie ca pentru început să creăm un obiect de tipul sferă. Mai departe, ne folosim de tab-ul din meniu denumit „Mesh” al soft-ului, unde regăsim secțiunea „Normals” pentru care avem opțiunea „Flip normals” care ne ajută să ajungem la efectul dorit. Pentru a putea valida că ceea ce am făcut este corect există o opțiune în tab-ul „Overlays” numită „Display face normals as lines” care ne permite să vedem în care direcție sunt orientate normalele, dacă normalele au fost inverse.

După ce am urmat aceste pași, următorul lucru care mai rămâne de făcut este să salvăm obiectul nostru în format .obj și să îl importăm în aplicația noastră cu ajutorul platformei Unity pentru a realiza încăperea sferică.

## 2.4 Biblioteca matplotlib

Python este un limbaj de programare interpretat, obiect orientat, de nivel înalt, cu semantică dinamică. Structurile sale de date încorporează la nivel înalt, combinate cu tastarea dinamică și legarea dinamică, îl fac foarte atractiv pentru dezvoltarea rapidă a aplicațiilor, precum și pentru a fi utilizat ca limbaj de scriptare sau lipici pentru a conecta componente existente împreună. Sintaxa simplă, ușor de învățat accentuează lizibilitatea și, prin urmare, reduce costul întreținerii programului. Python acceptă module și

pachete, ceea ce încurajează modularitatea programului și reutilizarea codului.

Studiul folosește biblioteca matplotlib pentru a reprezenta rezultate comparative pentru funcția de răspuns în timp sau sunetul după convoluție pe mai multe microfoane. Aceasta este o bibliotecă utilă pentru crearea de vizualizări statice, animate și interactive în Python. Matplotlib poate fi utilizat în scripturi Python, shell-urile Python și IPython, servere de aplicații web și diverse seturi de instrumente grafice de interfață cu utilizatorul [16].



Această bibliotecă a fost folosită independent de platforma Unity, strict pentru a putea realiza câteva grafice ce ilustrează rezultatele obținute de modelul acustic propus în această lucrare.

Pentru a putea folosi limbajul Python și biblioteca matplotlib au fost folosite Anaconda și Jupyter Notebook. În realizarea graficelor au fost folosite metode de bază precum: *plot*, *legend*, *figure*, *title*, *xlabel*, *ylabel* și *show*, dar și noțiuni precum: *variabilă*, *constantă*, *listă*.



## 2.5 Biblioteca NWaves

Biblioteca NWaves a fost inițial destinată cercetării, vizualizării și predării elementelor de bază ale programării DSP (Digital Signal Processing) și a sunetului. Toți algoritmii sunt implementați în limbajul de programare C#, cât mai simplu posibil și au fost proiectați în principal pentru procesarea offline (în prezent există și multe metode online) [17].

Această bibliotecă este una open source destinată procesării digitale a semnalelor audio, înglobând multiple funcționalități pentru acest domeniu. În cadrul studiului, biblioteca a fost folosită pentru citirea/stocarea fișierelor cu extensia „*wav*”, dar și pentru convoluția sunetelor.

Pentru a putea folosi această bibliotecă Unity propune fereastra *Package Manager*, care permite vizualizare, instalarea, ștergerea și actualizarea pachetelor utilizate în aplicație, ce are opțiunea *Add*, care permite adăugarea bibliotecii folosind un link de Git prin *Add package from Git URL*.

În realizarea acestei aplicații au foste folosite următoarele metode:

- *DiscreteSignal Operation.Convolve (DiscreteSignal signal, DiscreteSignal kernel);*
- *void RealFft.Inverse (float[] re, float[] im, float[] output);*

Cu ajutorul acestei biblioteci am realizat convoluția sunetului pe fiecare microfon, am utilizat clasa *DiscreteSignal* pentru a crea și prelucra sunetul difuzat în încăpere și mai mult de atât, am realizat transformarea Fourier din domeniul timpului în domeniul frecvențelor, dar și invers.



## 2.6 Biblioteca XCharts

XCharts este o bibliotecă puternică, ușor de utilizat și de configurat pentru vizualizarea datelor în contextul platformei Unity. Acest proiect a fost dezvoltat sub Unity 2017 și .NET 3.5 [19].

Diferite componente și date pot fi combinate în diferite tipuri de diagrame. O componentă XCharts este împărțită într-o componentă principală și în una sau mai multe componente secundare, unde cea principală conține toate componentele secundare.

Biblioteca a fost folosită în contextul acestei lucrări pentru a ilustra date statistice privind sunetul, precum: funcția de răspuns la impuls în frecvență și în timp, magnitudinea, faza, timpul, dar și altele.

Pentru a putea integra această bibliotecă în cadrul aplicației Unity procesul este un pic mai complicat decât pentru biblioteca NWaves. Astfel, pentru început trebuie să descarcăm pachetul ce conține biblioteca. Aceasta va cuprinde două pachete Unity: *XCharts* și *XChartsDemo*, pe care va trebui să le importăm în aplicația Unity. Acest lucru este posibil prin intermediul ferestrei *Package Manager*, în cazul în care se utilizează o versiune mai mare decât Unity 2018.3. În cazul în care se folosește o versiune mai mică de Unity, importarea se poate face în mod clasic prin copierea fișierelor.

Pasul următor pentru a continua configurarea bibliotecii presupune adăugarea textului: ”*com.monitor1394.xcharts*” : ”*https://github.com/monitor1394/unity-ugui-XCharts.git#2.0-upm*”, în fișierul *manifest.json* din directorul *Packages*. După acest pas, trebuie să ne întoarcem pe platforma Unity și să aşteptăm aproximativ 3-5 min pentru ca biblioteca să se descarce și să se compileze. După aceea, să ne întoarcem pe *Package Manager*, să selectăm *XCharts* și să vedem că aceasta este inclusă în meniul Unity la secțiunea *Components*. În final, să ne întoarcem pe editor și să găsim XCharts.

În realizarea graficelor ce prezintă informații despre modelul acustic au fost folosite următoarele elemente:

- BarChart - un grafic care prezintă date categorice cu bare dreptunghiulare cu înălțimi și lungimi proporționale cu valorile pe care le reprezintă
- LineChart - un grafic care afișează informații ca o serie de puncte conectate prin segmente de linie dreaptă
- Axis - o linie dreaptă care indică unitatea de măsură a unei cantități care este măsurată o cantitate
- Grid - un element din grafic care creează rețea de linii rectangulare într-un grafic
- Legend - descrie elementele ce sunt ilustrate în grafic
- Series - listă de elemente care își determină singure tipul
- Title - este un element grafic ce descrie numele graficul ilustrat

dar și altele.

## 2.7 Biblioteca StandaloneFileBrowser

Biblioteca StandaloneFileBrowser este una cross-platform ce permite lucrul cu fișiere pentru platforma Unity [18]. Aceasta a fost folosită în realizarea aplicației pentru a permite deschiderea și utilizarea fișierelor cu extensia *.txt*, prin intermediul căreia se poate crea input-ul pentru modelul acustic.

Aceasta este un wrapper peste fereastra de dialog din Windows cu care suntem obișnuiți, iar câteva dintre avantajele bibliotecii sunt: se poate folosi atât în editor, cât

și la runtime, oferă posibilitatea de a salva și filtra după extensie fișiere permite selecția multiplă a elementelor, poate fi folosită pe Windows, Mac și Linux.

StandaloneFileBrowser permite încărcarea fișierelor cu extensie „.ogg” pentru a putea converti sunetul în DiscreteSignal, ca mai departe să fie incluse în modelul fizic. Pentru a putea încărca un fișier în aplicația Unity am folosit metoda:

- `public static string[] OpenFileBrowser(string title, string directory, ExtensionFilter[] extensions, bool multiselect)`  `title` este titlul ferestrei, `directory` este directorul din care vom vizualiza fișierele, `extensionFilter` este o clasă ce permite stocarea extensiilor, iar `multiselect` este un parametru de tip bool care ne indică dacă vom putea sau nu să selectăm mai multe fișiere

După ce a fost încărcat fișierul audio folosind biblioteca StandaloneFileBrowser sunetul va fi convertit într-un DiscreteSignal cu ajutorul bibliotecii NWaves, care va transmite mai departe informațiile. Astfel StandaloneFileBrowser devine doar un intermediar ce permite încărcarea unui fișier.



# Capitolul 3

## Pașii elementari pentru implementarea aplicației



În acest capitol va descrie detaliat algoritmul de modelare acustică în spații interioare folosind Ray-Tracing și va conține pașii elementari pentru implementarea acestuia. În funcție de toate acestea, aplicația poate fi împărțită în patru etape:

### ■ Calcul geometric

- Sfera lui Fibonacci - este un algoritm ce presupune distribuirea punctelor în mod uniform pe suprafața unei sfere
- Propagarea razelor - presupune distribuirea unui număr finit de raze în încăpere pornind de la niște puncte de start și ținând cont de numărul maxim de reflexii pe care îl poate avea o rază și de distanța maximă a acesteia
- Selecția razelor - dintr-un număr cunoscut de raze se vor selecta doar acele care îndeplinesc anumite condiții

### ■ Calcul fizic

- Calculul intensităților - pentru fiecare punct de coliziune al unei raze trebuie calculată intensitatea ținând cont de suprafetele de impact
- Calculul presiunilor - fiecare intensitate calculată la pasul anterior trebuie transformată în presiune ținând cont de temperatura din mediu, viteza sunetului prin aer și de densitatea aerului
- Funcția de răspuns la frecvență și funcția de răspuns la impuls - este necesar să cunoaștem valorile acestor funcții pentru a putea obține rezultatele finale
- Calculul distanțelor - pentru fiecare rază trebuie să cunoaștem lungimea acesteia
- Calculul timpilor - pentru fiecare rază trebuie să cunoaștem momentul la care aceasta a lovit un anumit obiect, în cazul acestui studiu obiectul de interes este microfonul

### ■ Post-procesare

- Convoluția sunetului - presupune preluarea funcției de răspuns la impuls și a sunetului care a fost difuzat în încăpere și combinarea acestora pentru a obține soluția

## ■ GUI (Graphical User Interface)

Astfel, pentru a realiza acest model trebuie să încep prin a distribui uniform razele plecând de la sursă, după care trebuie să selectăm razele care intersectează microfoanele din încăpere. Pentru a obține performanță este necesar să excludem acele raze care nu ajung pe nici unul dintre microfoane sau acele raze care intersectează microfoanele, dar sunt foarte asemănătoare și să pastrăm doar una dintre aceste raze asemănătoare pentru a reduce din costul computațional.

Pentru pasul următor calculăm intensitățile pentru fiecare rază pe care mai apoi le transformăm în probabilități. În următoarea etapă presupune calculul frecvențelor și calcularea fazelor. După aceea trebuie să obținem funcția de răspuns la impuls.

Acum că am considerat toate etape mai importante, trebuie să obținem să ascultăm soluția algoritmului, iar modelul nostru acustic este complet.

## 3.1 Ghid de utilizare al aplicației

Acustica face parte din domeniul fizicii și se ocupă cu studiul undelor mecanice în gaze, lichide, solide și este prezent în toate aspectele societății actuale, însă predominant în domeniul industriei pentru controlul zgomotului. Această ramură implică și studiul sunetelor, vibrațiilor, ultrasunetelor și infrasunetelor. Un model acustic va implica, în cel mai generic mod, simularea căilor pe care sunetul le parcurge de la sursă la destinație. Cel mai adesea, aceste modele presupun rezolvarea integralei Helmholtz-Kirchoff [5] folosind diverse abordări de calcul, precum: soluții numerice ale ecuației de undă, aproximări de înaltă frecvență ale ecuației de undă, dar și modele statistice. Modelul pe care îl propune acest studiu face parte din cea de-a doua categorie. Modelul acustic va conține etapa de calcul geometric, etapa de calcul fizic, convoluția sunetului și realizarea unei interfețe ce permite validarea acestor rezultate.

Tinând cont de toate aceste elemente este foarte important să reținem că aceste etape se află într-o legătură strânsă și se bazează unele pe celelalte. Astfel, ne putem da seama că calculul geometric este o etapă deosebit de importantă, întrucât modul în care distribuim razele în încăpere și selecția acestora dictează performanța modelului și optimalitatea acestuia. Algoritmii de post-procesare sunt folosiți, de obicei, pentru a suprima zgomotul sau orice artefact creat în cadrul primelor două etape, adică calculul geometric și cel fizic, concentrându-se pe eliminarea distorsiunii și a ecoului.

Aplicația dezvoltată în această lucrare cuprinde toate etapele enunțate mai sus și urmează a fi detaliate în paginile ce urmează. Pentru început va fi prezentat modul în care arată aplicația și funcționalitățile acesteia, urmând să fie prezentat și modul în care a fost creat modelul acustic.

Interfața aplicației a fost realizată pe platforma Unity. Încăperile dreptunghiulare au fost construite folosind formule și se pot adăuga la dispoziție de către această platformă, iar încăperea sferică a fost creată în Unity. Pentru cea din urmă cameră a fost folosit un obiect de tip sferă pentru care au fost inversate normalele. Fiecare spațiu folosește culori și materiale potrivite utilizând Unity Material, o clasă care permite utilizatorului să animeze un obiect prin intermediul proprietăților materialului și dispune această clasă.

Pentru acest studiu au fost gândite trei încăperi: una dreptunghiulară și una sferică. Prima încăpere dreptunghiulară are dimensiunea de 10x10x10m și cea de-a două are 30x30m.



încăperea sferică are diametru de 10m. Acestea pot fi vizualizate în Figura 3.1. Pentru încăperile dreptunghiulare, s-a făcut poziționată în centru, iar pentru încăperea sferică sursa a fost pozitionată pe o poziție astfel încât să aparțină unui cerc pe care vor fi plasate microfoanele. Formele încăperilor au fost alese pentru a reprezenta o parte din formele de bază pe care o încăpere le poate avea.

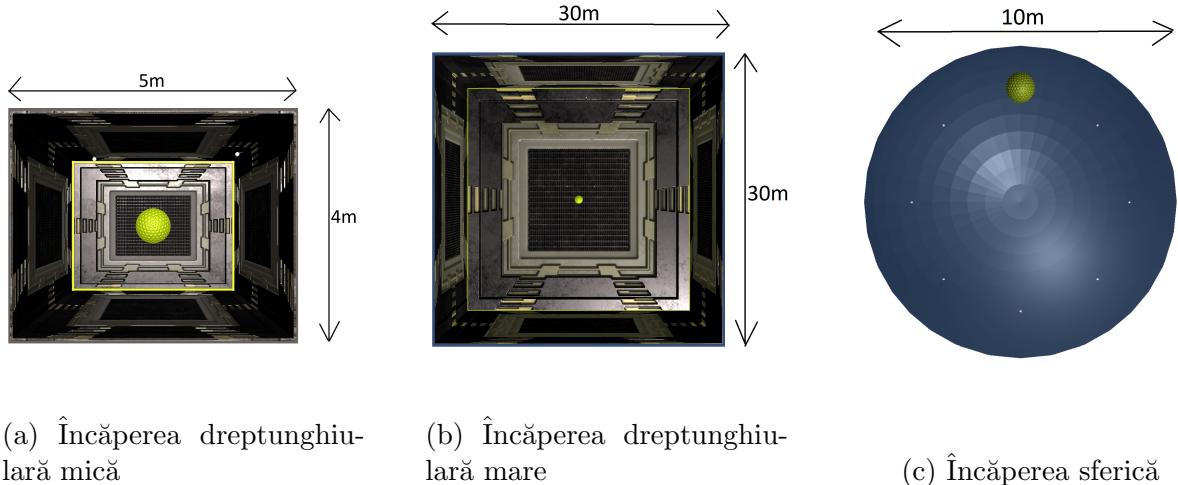


Figura 3.1: Tipuri de încăperi studiate

Meniurile aplicației au fost realizate tot cu ajutorul platformei Unity folosind elemente de UI precum: Canvas, Panel, Button, Dropdown, Label, Input, Text, dar și altele. Canvas-ul este un element ce conține o arie în interiorul căreia ar trebui să se afle toate celelalte elemente de UI, iar un Panel este utilizat pentru a susține mai multe elemente precum Button, Label, Input, Dropdown, Text etc. De asemenea, pentru butoane au fost alese imagini intuitive care să ușureze folosirea acestora, dar și un font adecvat pentru text astfel încât să fie lizibil pentru utilizator.



Figura 3.2: Meniul pentru setarea configurației

În Figura 3.2 putem observa meniul care ne ajută să setăm valorile pentru care dorim să analizăm ce se întâmplă într-o încăpere cu sunetul. Cu ajutorul acestei ferestre avem posibilitatea de a parametriza algoritmul și putem stabili care este numărul maxim de

reflexii permis pentru o rază, numărul de se vor împrăştia în cameră, distanța maximă permisă pentru o rază, pasul de filtre și sunetul ce va fi difuzat în încăpere. De asemenea, flexibilitatea aplicației este și în includerea celor 3 ferestre: *TIME ECHOGRAM*, *FREQUENCY RESPONSE* și *IMPULSE RESPONSE*, oferind o gamă variată de a vizualiza și interpreta rezultatele.



Figura 3.3: Meniul pentru vizualizarea razelor

Aplicația permite atât vizualizarea tuturor razelor din încăpere folosind butonul „Show all”, cât și vizualizarea unei raze individual prin intermediul butonului „Show”, aceste lucruri fiind ilustrate în Figura 3.3.

Mai mult de atât, pentru a realiza graficele ce definesc o parte din rezultatele solverului a fost folosită biblioteca XCharts. Aceasta a fost utilizată în aplicație pentru a crea un mod de a vizualiza funcția de răspuns la impuls, funcția de răspuns în frecvență, dar și ecograma de timp.

Biblioteca StandaloneFileBrowser a fost folosită ca interfață aplicației pentru a permite utilizatorului să încarce un fișier cu extensia „.m4a”. Într-un mod normal, acest buton a putea fi folosit mai departe drept parametru pentru modelul acustic într-un mod mai intuitiv.

## 3.2 Calcul geometric

Calculul geometric presupune simularea geometriei unei încăperi. Pentru acest lucru vom considera ca date de intrare suprafețele încăperii, sursa audio și microfoanele plasate în încăpere. Cu ajutorul acestora vom putea distribui uniform raze în încăpere folosind algoritmul sferei lui Fibonacci și vom păstra doar acele raze de care avem nevoie cu ajutorul unui algoritm pentru reducerea razelor dupicate. Aceste subetape urmează a fi prezentate în următoarele pagini.

### 3.2.1 Sfera lui Fibonacci

De-a lungul timpului, în literatura matematică, au existat multiple încercări pentru soluționarea problemei distribuirii uniforme a punctelor pe o sferă. Din păcate, cu excepția câtorva cazuri speciale, nu este posibil să se distribuie în mod egal punctele pe suprafața unei sfere.

Astfel, vom prezenta modul în care vor fi generate razele pe sursa audio în contextul algoritmului nostru. Sursa va fi reprezentată de un model propus de o sferă pe care se vor alege puncte uniform distribuite din care vor fi generate razele.

Chiar dacă seturile de puncte Fibonacci nu reprezintă cea mai bună distribuție globală a punctelor pe o sferă, ele oferă proprietăți excelente de eșantionare și sunt extrem de simple de construit față de alte modele.

Una dintre cele mai mari provocări pentru acest algoritm este determinat de faptul că distribuția optimă depinde în mod critic de funcția obiectiv pe care o utilizăm, în cazul nostru este dependentă de  $N$ , numărul de puncte pe care vrem să îl distribuim pe sferă.

În Figura 3.4, se poate observa impactul pe care  $N$  îl are în cadrul algoritmului. Cu cât  $N$  este mai mare, cu atât punctele pe sferă sunt mai bine distribuite.

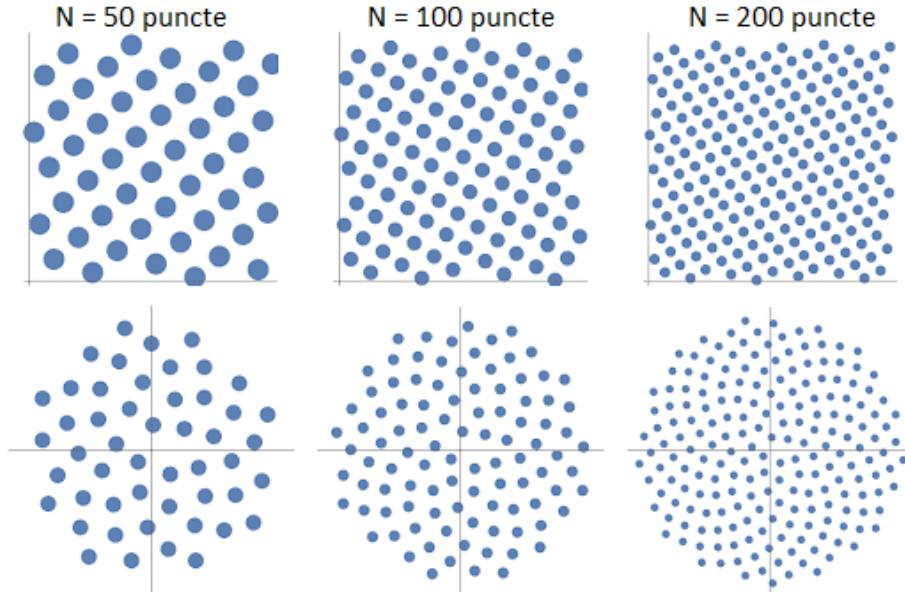


Figura 3.4: Rezultate pentru algoritmul sferei lui Fibonacci [12]

Acest algoritm se bazează pe o deplasare în spirală pe suprafața sferei incremental cu unghiul de aur, care este legat de raportul de aur. Două cantități,  $a$  și  $b$ , se află în raportul de aur dacă:  $\frac{a}{b} = \frac{a+b}{a} = \varphi$ , unde  $a > b$ , iar acest raport este aproximativ egal cu  $\varphi = \frac{1 + \sqrt{5}}{2}$ . Unghiul de aur,  $\vartheta$ , este definit în funcție de raportul de aur astfel:  $\vartheta = 2\pi(2 - \varphi)$ .

---

**Algorithm 1** Sfera lui F

---



```

1: procedure generare()
2:    $\varphi \leftarrow \frac{1 + \sqrt{5}}{2}$ 
3:    $\vartheta \leftarrow 2\pi * (2 - \varphi)$ 
4:   for  $i \leftarrow 0$  to  $N$  do
5:      $\theta \leftarrow \sin\left(-1 + \frac{2 * i}{N + 1}\right)$ 
6:      $\phi \leftarrow \vartheta * i$ 
7:     dir  $\leftarrow$  transformăm  $(1, \theta$  și  $\phi)$  în coordonate carteziene
8:       Scrie dir, direcția pe care se va deplasa viitoarea rază
9:   end for
10: end procedure
```

---

Conform Algoritmului 1, vom distribui razele uniform pe suprafața sferei, urmând ca mai departe să generăm geometria razelor.

### 3.2.2 Propagarea razelor

După ce am stabilit la pasul anterior punctul de start pentru fiecare rază, preocupa de modul în care vor fi generate acestea. Pentru acest lucru este nevoie să ținem cont de câteva aspecte precum: legea reflexiei și cea a refracției, care este distanța maximă pe care o poate acoperi o rază, care este numărul maxim de reflexii pe care îl dorim pentru modelul nostru.

Pentru a calcula lungimea unei raze vom folosi distanța Euclidiană:

$$d(x, y) = \sqrt{(x - y)^2} \quad (3.1)$$

În cazul în care nu am ținut cont de lungimea razei am putea ajunge în situația în care am crea o rază care are mai puține reflexii decât numărul maxim de reflexii, dar care s-ar propaga la infinit pentru că nu ar mai întâlni o suprafață în care să se reflecte.

Ca să putem crea o rază trebuie să stabilim de la bun început și care este numărul maxim de reflexii pe care dorim să îl permitem. Acest parametru este ales dependent de problema pe care vrem să o rezolvăm, dacă dorim să ajungem la o anumită performanță sau dorim să obținem o soluție cât mai apropiată de realitate.

Pentru a rezolva problema propagării razelor am folosit următoarea metodă pusă la dispoziție de platforma Unity:

- `public static bool Raycast (Vector3 origin, Vector3 direction, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers, QueryTriggerInteraction queryTriggerInteraction = QueryTriggerInteraction.UseGlobal)`, unde *origin* este punctul din care pornește raza, *direction* este direcția pe care se deplasează raza, *maxDistance* este lungimea maximă pe care o rază o poate avea, *layerMask* este o mască care este utilizată pentru a ignora selectiv collider-urile atunci când este aruncată o rază, iar parametrul *queryTriggerInteraction* specifică dacă această interogare trebuie să atingă Trigger. Pentru a calcula direcția pe care urmează să se deplaseze raza [13].

Numărul maxim de reflexii influențează în mod direct atât complexitatea, cât și performanța algoritmului. Dacă vom alege un număr foarte mare de reflexii, acesta va duce la creșterea timpului de calcul. O rază va parcurge o suprafață mare a camerei atunci când considerăm un număr mare de reflexii. În schimb, dacă alegem un număr prea mic de raze, vom întâmpina probleme, pentru că putem fi puși în situația în care nici una dintre raze nu a ajuns pe microfon, ceea ce va avea un impact negativ asupra rezultatului pe care îl obținem, întrucât nu vom obține rezultate relevante pentru acea încăpere.

### 3.2.3 Selecția razelor

Unul dintre pașii esențiali pentru a obține performanță este reducerea numărului de raze, păstrându-le doar pe acele care ajung pe unul dintre microfoanele plasate în încăpere. Mai mult de atât, se pot face îmbunătățiri prin eliminarea duplicatelor. Duplicatale sunt acele raze care difează printr-un prag definit, iar păstrarea acestora crește complexitatea algoritmului, fără a aduce valoare.

Pentru a selecta doar acele raze care intersectează microfonul am folosit ecuația sferei [14]:

$$(x - x_{centru})^2 + (y - y_{centru})^2 + (z - z_{centru})^2 = r^2 \quad (3.2)$$

și a liniei:

$$\begin{cases} x = x_1 + x_2 - x_1 \\ y = y_1 + y_2 - y_1 \\ z = z_1 + z_2 - z_1 \end{cases} \quad (3.3)$$

Substituind  $x, y, z$  din ecuația sferei obținem:

$$\begin{cases} a = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \\ b = -2[(x_2 - x_1)(x_{centru} - x_1) + (y_2 - y_1)(y_{centru} - y_1) + (z_2 - z_1)(z_{centru} - z_1)] \\ c = (x_{centru} - x_1)^2 + (y_{centru} - y_1)^2 + (z_{centru} - z_1)^2 - r^2 \end{cases} \quad (3.4)$$

iar pentru a verifica condiția de intersecție avem următoarele situații:

$$\begin{cases} \text{sunt tangente} & b^2 - 4ac = 0 \\ \text{se intersectează} & b^2 - 4ac > 0 \\ \text{nu se intersectează} & b^2 - 4ac < 0 \end{cases} \quad (3.5)$$

Două raze sunt duplicate dacă următoarele condiții sunt adevărate:

1. cele două raze trebuie să aibă același număr de puncte de coliziune;
2. diferența absolută dintre lungimile celor două raze nu trebuie să depășească un prag ales; pragul pe care l-am folosit pentru algoritmul propus a fost  $\epsilon = 10^{-2}$ ;

## Algorithm 2 Reducerea duplicatelor

```

procedure reducere_duplicate(raze)
2:   index = 0
    while index < no(raze) do
4:     if razeindex și razeindex+1 sunt raze directe then
          eliminăm razeindex din raze
6:     else if |distanță(razeindex) - distanță(razeindex+1)| <  $\epsilon$  and
          no(razeindex.puncte_coliziune) = no(razeindex+1.puncte_coliziune) then
8:       if razeindex.puncte_coliziune = razeindex+1.puncte_coliziune then
          eliminăm razeindex din raze
10:      index  $\leftarrow$  index + 1
          end if
12:      else index  $\leftarrow$  index + 1
          end if
14:    end while
end procedure

```



În Algoritm 2 am prezentat modul în care a fost realizată reducerea de duplicate după ce am considerat doar acele raze care ajung pe microfon. Până la acest pas am stabilit cum vom distribui razele în orice încăpere, am stabilit care va fi geometria camerei și am păstrat doar acele informații de interes, adică acele raze care ajung pe microfoanele din încăpere, dar fără duplicate.

Selectia razelor poate crește timpul computațional de la câteva milisecunde, la câteva ore sau chiar zile, în funcție de dimensiunile camerei și numărul de raze distribuit în

încăpere. Complexitatea în timp pentru acest pas va fi liniară, încrucât suntem nevoiți să trecem prin toate razele.

Astfel, descrierile următoare prezintă într-un mod detaliat modelul fizic și matematic pentru algoritmul nostru.

### 3.3 Calcul fizic

Am ajuns astfel la etapa de mijloc a modelului, ce propune calcularea câtorva elemente pentru a putea simula fizica unei încăperi. La pasul anterior, am realizat geometria unui spațiu interior, iar mai departe este nevoie să calculăm intensitățile, presiunile, distanțele, timpii și funcțiile de răspuns în frecvență, operații pentru care se va păstra complexitatea liniară precum la etapa anterioară. Aceste detalii urmează să fie prezentate mai jos.

#### 3.3.1 Trecerea în domeniul frecvențelor

Trecerea valorilor din domeniul timpului în domeniul frecvențelor este un pas esențial în realizarea algoritmului pentru modelarea acustică a spațiilor închise, încrucât ajută la simplificarea calculelor, reducând astfel timpul computațional pe care îl necesită rularea algoritmului de la câteva minute sau ore la câteva secunde sau milisecunde. Acest procedeu este realizat cu ajutorul Transformatei Rapide Fourier.

#### 3.3.2 Calculul intensităților

Pentru a calcula intensitățile pentru fiecare rază am folosit Legea Pătratului Invers [15] care afirmă ca o mărime fizică specificată este invers proporțională cu pătratul distanței de la sursa acelei mărimi fizice. Cauza fundamentală pentru aceasta poate fi înțeleasă ca diluție geometrică corespunzătoare radiației punct-sursă în spațiul 3D.

Pentru a preveni diluarea energiei în timpul propagării unui semnal, pot fi utilizate anumite metode, cum ar fi un ghid de undă, care acționează ca un canal pentru apă sau modul în care un butoi de pistol restricționează expansiunea gazului fierbinte la o dimensiune pentru a preveni pierderea transferului de energie către un glonț.

Legea pătratului invers se aplică în general atunci când o anumită forță, energie sau altă cantitate conservată este uniform radiată spre exterior dintr-o sursă punctuală în spațiul 3D. Deoarece suprafața unei sfere este proporțională cu pătratul razei, pe măsură ce radiația emisă se îndepărtează de sursă, aceasta se întinde pe o zonă care crește proporțional cu pătratul distanței de la sursă. Prin urmare, intensitatea radiației care trece prin orice zonă unitară (direct orientată spre sursa punctuală) este invers proporțională cu pătratul distanței de la sursa punctuală.

Astfel, formula intensității este:

$$\frac{I_n}{I_{n+1}} = \frac{d_{n+1}}{d_n} \quad (3.6)$$

unde  $d_n$  reprezintă distanța de la sursă până la punctul al  $n$ -lea.

În cazul algoritmului ales, atunci când o rază parcurge încăperea aceasta se lovește de diferite materiale ce au un factor de absorbție  $\alpha$  care variază în funcție de material. Pentru a putea ține cont de fenomenul de absorbție am inclus acest factor în formula intensității:

$$\frac{I_n}{I_{n+1}} = \frac{d_{n+1}}{d_n} (1 - \alpha_k)^2 \quad (3.7)$$

unde  $d_n$  are aceeași semnificație ca la pasul anterior, iar  $\alpha_k$  reprezintă coeficientul de absorbție pentru al k-lea material.

### 3.3.3 Calculul presiunilor

Pentru a putea calcula presiunile, ne vom folosi de intensitățile calculate la pasul anterior, ținând cont de densitatea aerului și de viteza sunetului prin aer. Vom considera că densitatea aerului,  $\rho_{aer}$ , are valoarea  $1.2041 \frac{kg}{m^3}$ , iar viteza sunetului prin aer,  $c_{aer}$ , este  $343.21 \frac{m}{s}$ , pentru o temperatură constantă de  $20^\circ C$ .

Prin urmare, formula de transformare a intensității ( $I$ ) în presiune ( $p$ ) este:

$$p = \sqrt{2 \cdot I \cdot \rho_{aer} \cdot c_{aer}} \quad (3.8)$$

Nivelul puterii sonore cuantifică energia sonoră total radiată de la un obiect. Spre deosebire de presiunea sonoră, intensitatea este independentă de distanța față de sursa de sunet, zona încunjurătoare și alte influențe.

Presiunile, de obicei, au valori pozitive, totuși, există și situații în care acestea pot fi negative. Presiunea este definită ca  $\frac{F}{A}$ , unde  $F$  reprezintă forță, iar  $A$  reprezintă aria. Dacă o zonă închisă are o presiune mai mare decât aria din jurul ei, gazul/lichidul împinge pentru a ieși, astfel obținem o presiune pozitivă. Atunci când zona închisă are o presiune mai mică decât în jur acea presiune va avea o valoare negativă.

De exemplu, atunci când suflăm într-un pai se creează o presiune pozitivă în interiorul puiului, pe măsură ce aerul se forțează să iasă din pui. Îndărătând tragem dintr-un pai, se creează o presiune negativă în interiorul puiului, și pe măsură ce intrăm în ura această presiune, aerul sau băutura se ridică în el.

### 3.3.4 Funcția de răspuns în frecvență și funcția de răspuns la impuls

Pentru fiecare frecvență din interval vom calcula faza și magnitudinea:

$$\begin{cases} t = \frac{c_{aer}}{fr} \\ w = \frac{2\pi}{t} \\ \theta = \arctan \frac{-\sin w \cdot d}{\cos w \cdot d} \end{cases} \quad (3.9)$$

unde  $fr$  este frecvența dată,  $\theta$  este faza pe care dorim să o calculăm, iar  $d$  este lungimea razei. Ca să obținem magnitudinea ne vom folosi de presiunea calculată anterior.

În urma acestui calcul obținem faza și magnitudinea în coordonate polare și dorim să le transformăm în coordonate carteziene. Astfel, vom obține pentru fiecare frecvență un număr complex de magnitudine  $m$  și fază  $\theta$ . Lista de numere complexă obținută pentru fiecare microfon o vom numi *ecogramă*.

În Figura 3.5 putem observa care sunt etapele algoritmului. Pentru a putea ajunge la convoluția sunetului trebuie să trimitem ca input valori în domeniul timpului. Obținerea

acestor valori se realizează prin aplicarea Inversei Transformatei Rapide Fourier peste funcția de răspuns în frecvență.

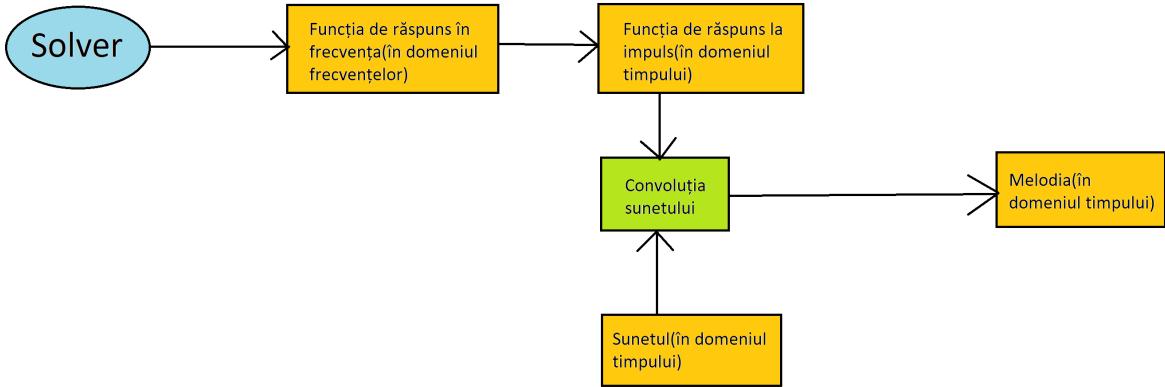


Figura 3.5: Etapele parcuse pentru obținerea sunetului după convoluție

Funcția de răspuns la impuls se calculează pe baza funcției de răspuns în frecvență cu ajutorul Inversei Transformatei Rapide Fourier care va permite trecerea de la domeniul frecvențelor la domeniul timpului pentru a pregăti semnalul de intrare ca mai departe să realizăm convoluția sunetului. Ca să putem trece mai departe și să calculăm funcția de răspuns la impuls trebuie să utilizăm Algoritm 3.

---

#### Algorithm 3 Crearea funcției de răspuns în frecvență

---

  
**procedure** *răspuns\_frecvență(ecograme, microfoane, fr)*  
     *răspunsFr*  $\leftarrow$  dicționar gol  
     3:     **for** *i*  $\leftarrow 0$  to *microfoane* **do**  
         *valori*  $\leftarrow$  listă numere complexe goală  
         **for** *j*  $\leftarrow 0$  to *fr* **do**  
             *sum*  $\leftarrow \sum ecograme[fr[j]][microfoane[i]]  
             *valori.adaugă(sum)*  
         **end for**  
     9:     *răspunsFr[microfoane[i]]*  $\leftarrow$  *valori*  
     **end for**  
**end procedure**$

---

Acum că am obținut răspunsul în frecvență putem trece la a calcula răspunsul la impuls, iar acest lucru se poate face cu ajutorul IFFT, care va permite trecerea de la domeniul frecvențelor înapoi la domeniul timpului.

### 3.3.5 Calculul distanțelor

Pentru a calcula distanțele am folosit formula distanței lui Euler. Am considerat pentru fiecare rază, toate punctele de coliziune și am calculat lungimile pentru acestea iterativ.

$$d(p_1, p_2) = \sqrt{(p_{1x} - p_{2x})^2 + (p_{1y} - p_{2y})^2 + (p_{1z} - p_{2z})^2} \quad (3.10)$$

Prin urmare, lungimea unei raze este calculată astfel:

$$d_{razei} = \sum_{i=1}^{N-1} d(p_i, p_{i+1}) \quad (3.11)$$

unde  $p_1, p_2, \dots, p_N$  sunt punctele de coliziune, iar  $N$  reprezintă numărul de puncte de coliziune.

### 3.3.6 Calculul timpilor

 Calculul timpului de călătorie se aplică calculul lungimilor pentru fiecare rază și viteza sunetului, folosind următoarea formula:

$$t = \frac{d}{c_{aer}} \quad (3.12)$$

unde  $t$  este timpul,  $d$  reprezintă distanța de la sursă până la ultimul punct de coliziune al razei, iar  $c_{aer}$  este viteza sunetului prin aer.

## 3.4 Post-procesarea

Am ajuns și la ultima etapă unde vom aplica operații de post-procesare pentru a putea asculta sunetul pe microfoanele plasate în încăpere. Astfel, la acest pas vom aplica un algoritm pentru convoluția sunetului ce va fi descris în paginile ce urmează.

### 3.4.1 Convoluția sunetului

Convoluția sunetului se face pe baza funcției de răspuns la impuls. Răspunsul este alcătuit dintr-o listă de valori reale care sunt transformate într-un semnal discret. Convoluția presupune un sistem ce primește ca intrare un semnal și pe care îl transformă pentru a obține un semnal de ieșire. În cazul acestui model, semnalul de intrare este funcția de răspuns la impuls și sunetul în domeniul timpului, iar ieșirea este reprezentată de valorile reale ale sunetului.

Teoria Fourier spune că orice semnal digital poate fi exprimat ca o combinație de sinusoide, din acest motiv preferăm să realizăm calculele în domeniul frecvențelor (este mult mai rapid).

Precum este descris și în Figura 3.6, calculatorul sunetului preia ca intrare funcția de răspuns la impuls și sunetul care a fost capturat de sursa audio. Aceste valori sunt în domeniul timpului și pentru a putea să le aplicăm transformările trebuie să trecem în domeniul frecvențelor. Obținem astfel două polinoame ce trebuie înmulțite. Răspunsul obținut este în domeniul frecvențelor și pentru a calcula ieșirea trebuie să trecem înapoi în domeniul timpului cu ajutor Inversei Transformantei Rapide Fourier. Convoluția sunetului, transformările din domeniul timpului în domeniul frecvențelor și invers au fost realizate cu ajutorul metodelor din NWaves:

- *DiscreteSignal Operation.Convolve (DiscreteSignal signal, DiscreteSignal kernel);*
- *void RealFft.Inverse (float[] re, float[] im, float[] output);*

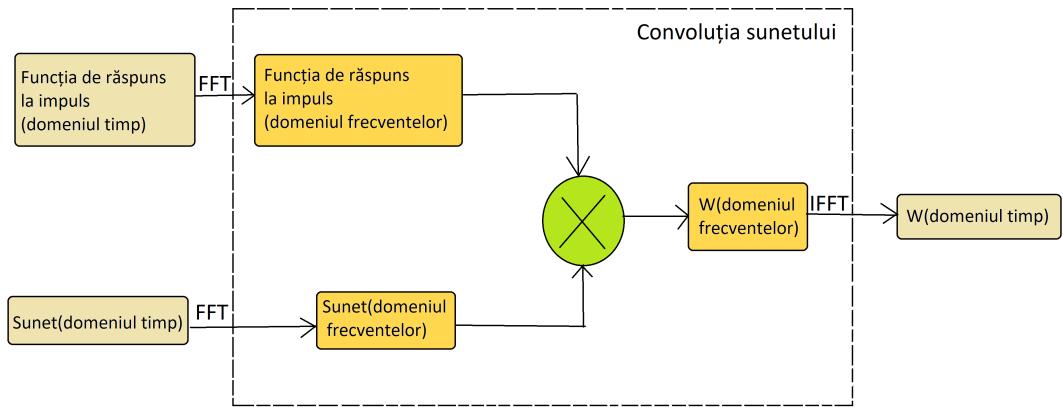


Figura 3.6: Etapele parcuse pentru conoluția sunetului

Procedura este semnificativ mai eficientă din punct de vedere al timpului de calcul decât celelalte două etape, deoarece folosind Transformata Fourier Rapidă (FFT) ajungem la  $n \log_2 n$  operații necesare pentru a realiza conoluția sunetului.

# Capitolul 4

## Testarea și analizarea rezultatelor

### 4.1 Realizarea experimentelor și analizarea acestora

Pentru a putea testa și analiza rezultatele obținute folosind modelul acustic propus de această lucrare am creat o modalitate interactivă pentru a putea vizualiza și compara rezultatele obținute de modelul nostru folosind platforma Unity pentru a gestiona și dezvolta un software de simulare acustică folosind un game engine. Am construit trei încăperi, două dreptunghiulare și una sferică, pornind de la forme de bază și materiale contrastante pentru a putea facilita vizualizarea rezultatelor.

În cadrul experimentelor realizate au fost folosite următoarele spații interioare 3D:

- Cameră rectangulară cu dimensiunile: 4m lățime, 5m lungime și 3m înălțime - Camera 1
- Cameră rectangulară cu dimensiunile: 30m lățime, 30m lungime și 15m înălțime - Camera 2
- Cameră sferică cu raza egală cu 5m - Camera 3

Această serie de experimente ia în considerare mai mulți parametri esențiali în contextul modelului acustic, precum: numărul de raze ce va fi distribuit în încăpere, numărul maxim de reflexii pe care o rază le poate atinge, lungimea maximă pe care o poate avea o rază și pasul de frecvență.

În această lucrare vom considera următoarea serie de notații:

- $S$  - sursă audio
- $M$  - microfon

Atunci când discutăm despre un punct în spațiu am considerat un sistem de coordonate în trei dimensiuni ( $x, y, z$ ), unde  $x$  este axa ce reprezintă lungimea,  $y$  este înălțimea, iar  $z$  este lățimea.

Pentru ca experimentele să fie relevante am considerat aceeași configurație și aceeași melodie ce va fi difuzată pentru toate încăperile. Configurația folosită va fi:

- 100 000 de raze distribuite uniform pentru camerele dreptunghiulare și 10 000 în cazul încăperii sferice
- maxim 10 reflexii pentru o rază

- 200m distanță maximă pe care o rază o poate parcurge
- 8192Hz rezoluția frecvenței

Pentru rularea modelului acustic este nevoie să considerăm o serie de microfoane, întrucât dorim să analizăm sunetul în anumite poziții ale încăperilor. Astfel, în Camera 1 am considerat poziția sursei  $S(0, 0, 0)$ . Următoarele poziții pentru microfoane:

- $M_0(2, 1.6, 1.7)$ , fiind la o distanță de 7m de sursă
- $M_1(-1.5, 1.2, 1.7)$ , fiind la o distanță de 2.56m de sursă

În Camera 2, sursa  $S$  se află la poziția  $S(0, 2.5, 0)$  și am considerat următoarele valori pentru pozițiile microfoanelor:

- $M_0(2, 1.6, 1.7)$ , fiind la o distanță de 3.07m de sursă
- $M_1(-1.5, 1.2, 1.7)$ , fiind la o distanță de 2.56m de sursă
- $M_2(1, 2, 13)$ , fiind la o distanță de 13.19m de sursă

Pentru Camera 3, sursa  $S$  este poziționată în punctul  $S(0, 0, 4)$ . Am utilizat ecuația Cercului pentru a putea împrăștia uniform microfoanele în încăpere, obținând astfel următoarele valori pentru pozițiile microfoanelor:

- $M_0(4, 0, 0)$ , fiind la o distanță de 5.65m de sursă
- $M_1(2.8284, 0, 2.8284)$ , fiind la o distanță de 3.06m de sursă
- $M_2(-2.8284, 0, 2.8284)$ , fiind la o distanță de 3.06m de sursă
- $M_3(-4, 0, 4.8984)$ , fiind la o distanță de 4.09m de sursă
- $M_4(-2.8284, 0, -2.8284)$ , fiind la o distanță de 7.39m de sursă
- $M_5(-7.347, 0, -4)$ , fiind la o distanță de 10.86m de sursă
- $M_6(2.8284, 0, -2.8284)$ , fiind la o distanță de 7.39m de sursă

Mediul în care trăim conține suprafețe și materiale, care absorb sau reflectă sunetul, fapt care atenuază sau împrăștie ceea ce noi auzim. Posibilitatea de a întâlni suprafețe care să absoarbă sau să reflecte total sunetul este una foarte mică, de obicei, acest lucru se întâmplă doar în condiții de laborator.

În funcție de suprafețe, coeficientul de absorbție diferă foarte mult, de obicei, obiecte precum covoarele, pânzele, pernele, absorb sunetul mult mai bine decât materiale precum lemnul, betonul, poliesterul. Astfel, am dorit ca soluția modelului acustic să fie una cât mai realistă și am considerat pentru toate suprafețele din încăperi o serie de coeficienți de absorbție. În cadrul experimentelor realizate, am ales coeficientul de absorbție 0.2.

În Figura 4.1 putem vizualiza cele trei încăperi, microfoanele și sursa audio realizate cu ajutorul platformei Unity. Se poate observa că prima încăpere este mai mică decât cea de-a doua și că cele două spații sunt similară. Pentru Camera 3, se va considera distribuirea uniformă a microfoanelor folosind ecuația cercului.

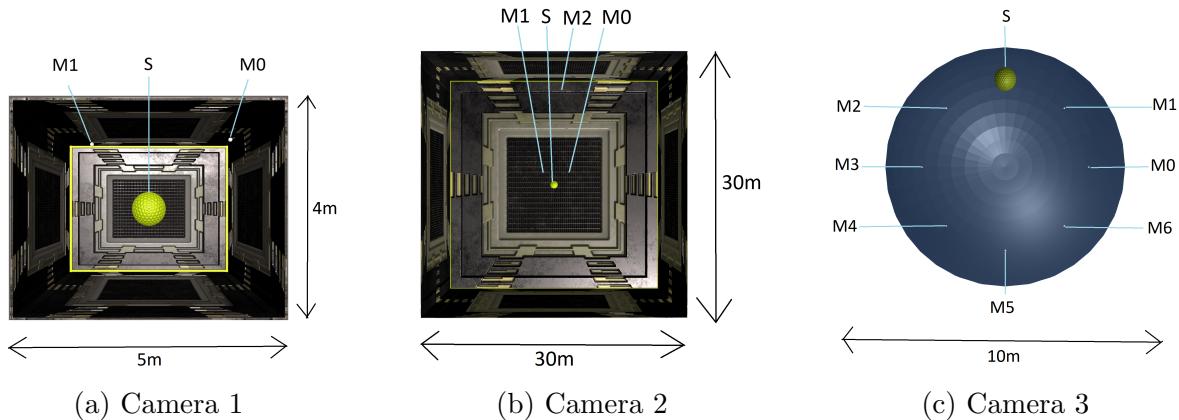


Figura 4.1: Încăperile folosite pentru a realiza experimentele

Figura 4.2 prezintă unul din modurile de vizualizare al razelor din aplicație. În aceste imagini a fost utilizată configurația parametrilor, poziționarea microfoanelor și a sursei audio prezentate mai sus.

Putem observa în cazul Subfigurii 4.1a că datorită configurației potrivite este acoperită cea mai mare parte din suprafața încăperii, nefind prezente „spații goale”, semn care indică faptul că sunetul a fost imprăștiat în toată camera și că poate fi auzit peste tot.

Subfigura 4.1b ilustrează cea de-a două încăpere și razele ce au fost distribuite în aceasta. Datorită configurației alese și a pozițiilor microfoanelor putem observa că multe porțiuni ale camerei au rămas neparcuse.

În Subfigura 4.1c putem observa că datorită configurației potrivite și al modului simetric în care au fost plasate microfoanele razele au fost împărățiate în încăpere în mod uniform.

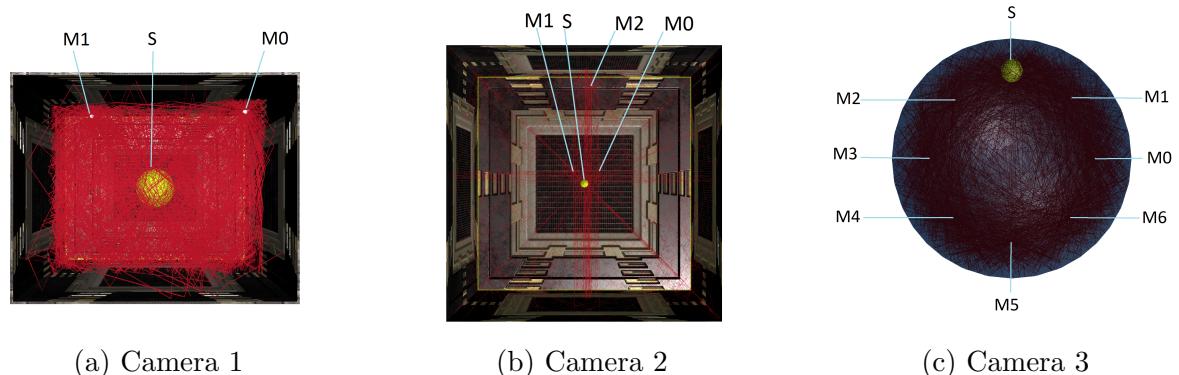
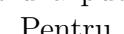


Figura 4.2: Vizualizarea tuturor razelor din fiecare încăpere

De asemenea, am realizat și un experiment în Camera 3 care presupune plasarea unui obstacol sferic între microfoanele  $M_4$  și  $M_5$  pentru a putea vedea dacă în acest mod sunt afectate rezultatele folosind modelul  . Pentru model am considerat aceeași configurație ca cea pentru Camera 1 și  2, cu diferența că în încăperea sferică am considerat numărul de raze egal cu 500. În urma calculelor efectuate, am descoperit că 442 de raze au ajuns, în total, pe microfoane în cazul expus de Subfigura 4.2c și 378 în cazul expus de Subfigura 4.3c. Mai mult, am observat că, în cazul în care avem un obstacol în cameră, razele sunt mai împrăștiate și că anumite porțiuni nu sunt atât de dense, deoarece datorită obstacolului sunt create noi căi de propagare.

Pe baza situațiilor prezentate mai sus putem vedea că în cazul încăperilor dreptunghiulare pe colțurile camerelor razele ajung mai greu, în timp ce în cazul încăperilor sferice razele ajung mai greu la centrul sferei.

Dacă numărul de raze va fi prea mic în comparație cu dimensiunea camerei noastre, nu vom putea propaga sunetul foarte bine, deoarece există posibilitatea ca foarte puține raze sau chiar nici una să ajungă pe unul dintre microfoanele plasate în cameră. Deci, cu cât dimensiunea camerei este mai mare, cu atât este mai mare numărul de raze care vor fi distribuite în cameră și același lucru se întâmplă pentru numărul de reflexii și pentru lungimea maximă posibilă a unei raze.

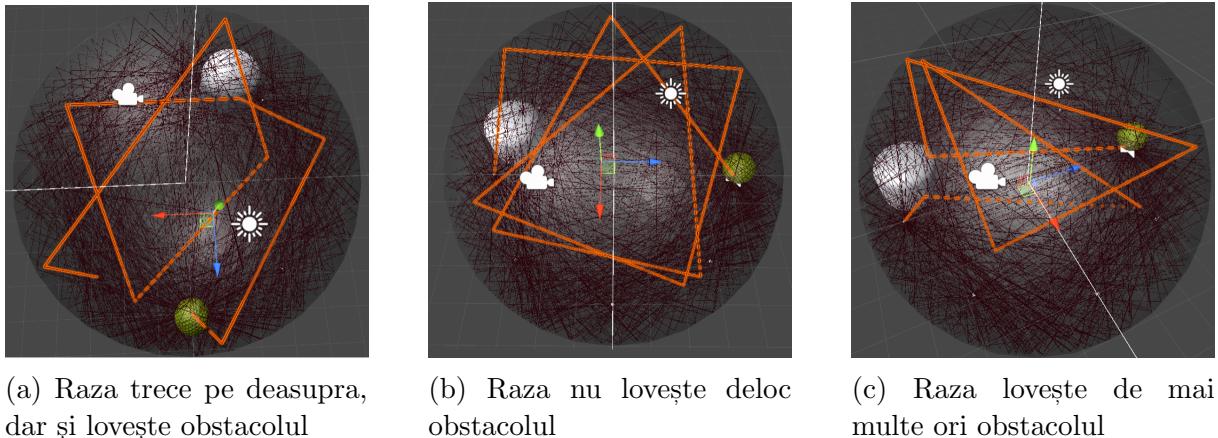


Figura 4.3: Diferite scenarii ale razelor

Urmărind Figura 4.3 putem observa că atunci când plasăm un obstacol în încăpere putem să ne aflăm în una dintre următoarele situații atunci când discutăm despre calea pe care o rază o parcurge: raza poate lovi sau nu obstacolul, poate trece pe deasupra sau pe dedesubtul acestuia.

Considerând aceste cazuri de propagare al razelor în încăperea sferică putem trage concluzia că razele pot avea căi foarte diferite de a ajunge la microfon și că atunci când distanța unei raze de la sursa audio la microfon este foarte mare și sunetul va ajunge mai târziu.

Atunci când discutăm despre ecograma în timp a unui microfon ne vom referi la graficul ce descrie presiunea și timpul. Se poate observa că Figura 4.4 conține o serie de bare verticale, unde fiecare bară reprezintă una dintre razele ce au ajuns pe microfonul 2 în Camera 2. Prima bară se caracterizează printr-o presiune crescută, deoarece corespunde razei care ajunge de pe sursă pe microfon în mod direct, fără a mai lovi alte suprafete din încăpere. După această rază, urmează toate celelalte raze care au ajuns pe microfonul 2. Ordinea va fi dată de către numărul de reflexii și de către momentul în care raza ajunge pe microfon. Astfel, vom avea mai întâi raza directă, iar mai apoi razele care au o reflexie, după aceea două și aşa mai departe până ajungem la raze ce au 10 reflexii.

Funcția de răspunsul în frecvență este măsura cantitativă a spectrului de ieșire al unui sistem sau dispozitiv ca răspuns la un stimул și este utilizat pentru a caracteriza dinamica sistemului. În termeni simpli, dacă o undă sinusoidală este injectată într-un sistem la o frecvență dată, un sistem liniar va răspunde la aceeași frecvență cu o anumită magnitudine și un anumit unghi de amplitudine la intrare.

Funcția de răspuns în frecvență are, în general, valori complexe, cu părți reale și imaginare. Acest lucru este deseori mai util și mai intuitiv atunci când este exprimat

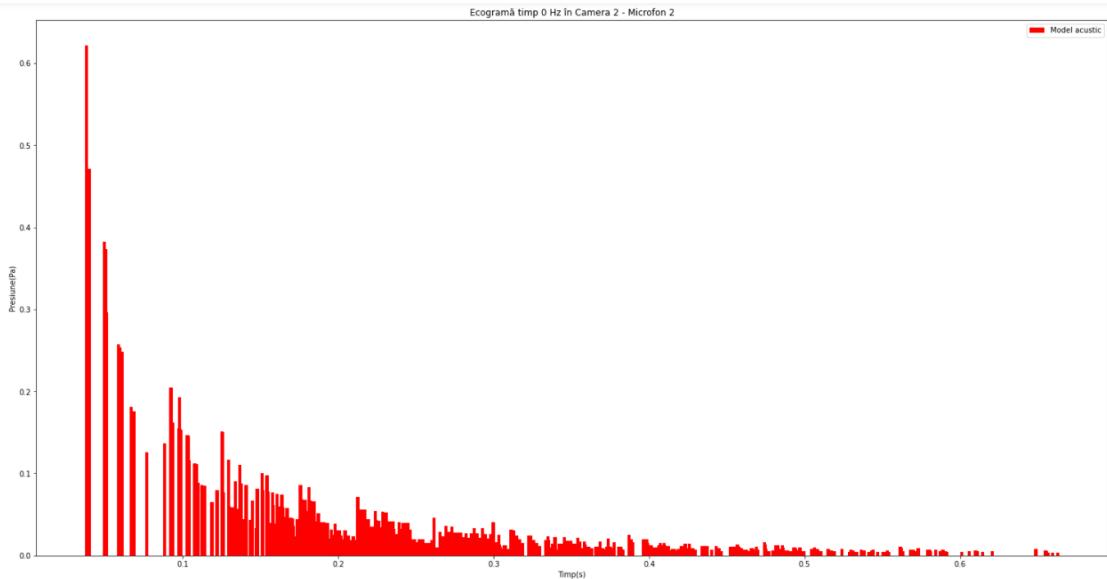
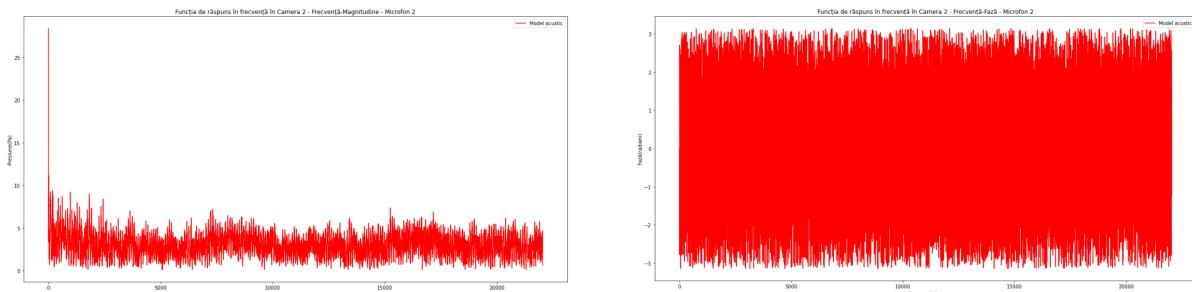


Figura 4.4: Ecogramă timp pe microfonul  $M_2$  în Camera 2

în coordonate polare. Adică îl putem separa în magnitudinea sa (numită răspuns de amplitudine) și în componentă de fază (numită răspuns de fază).

Reprezentarea domeniului de frecvență al unui semnal conține informații despre magnitudinea și faza semnalului la fiecare frecvență. Acesta este motivul pentru care ieșirea calculului FFT (Fast Fourier Transform) este complexă. Ieșirea FFT este un vector complex care conține informații despre conținutul semnalului. Faza prezintă modul în care se aliniază componentele de la 0° la 180° în timp. Aceste lucruri pot fi vizualizate în Figura 4.5, în cazul microfonului  $M_2$ .



(a) Grafic Frecvență-Magnitudine pe microfonul  $M_2$  în Camera 2

(b) Grafic Frecvență-Fază pe microfonul  $M_2$  în Camera 2

Figura 4.5: Funcția de răspuns în frecvență pe microfonul  $M_2$  în Camera 2

În procesarea semnalului, un răspuns la impuls este ieșirea unui sistem atunci când alimentăm sistemul cu un impuls ca semnal de intrare. Un impuls este un semnal de scurtă durată. Cu toate acestea, în procesarea semnalului, folosim o funcție Dirac Delta pentru sistemele analogice/continue. Pentru microfonul  $M_2$  din Camera 2 am obținut funcția de răspuns la impuls prezentată în Figura 4.5(a). În figura 4.5(b) putem observa surse și ciborâșuri ce însuflarează fenomenul de ecou.

Pentru a putea analiza care este diferența de timp cu care ajunge semnalul pe microfoane am realizat un experiment care presupune desenarea tuturor punctelor obținute după pasul de convoluție. În Figura 4.6 putem observa semnalele rezultatele pe toate

microfoanele din Camera 2, unde semnalul ajunge cel mai repede pe microfonul 1, fiind la o distanță de 2.56m de sursă, apoi la o distanță foarte mică de timp pe microfonul 0, aflat la 3.07m de sursă, iar în cele din urmă ajunge pe microfonul 2, care este plasat la o distanță de 13.19m de sursă. Concluzionând, cu cât microfonul se află mai departe de sursa audio, cu atât razele vor parcurge un drum mai lung în încăpere și sunetul va ajunge mai târziu pe microfon.

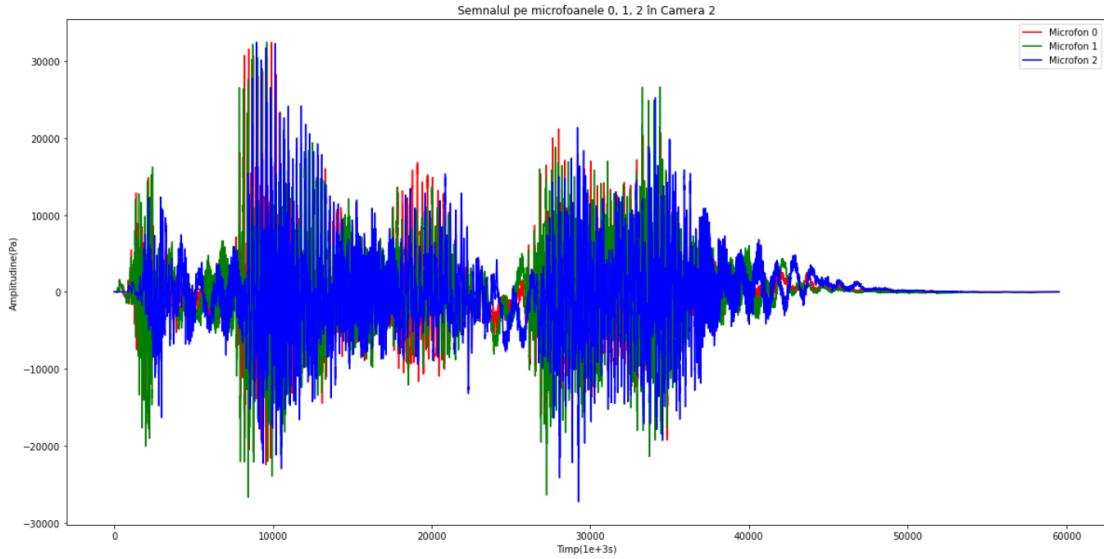


Figura 4.6: Semnalele rezultate pe microfoanele  $M_0, M_1, M_2$  în Camera 2

Rezultatul este impresionant. În aceste camere sunt diferențe chiar și pentru camera mică putând să fie de peste 10000 de Pa diferență între sunetul de pe primul microfon și cel de pe al doilea microfon. De altfel, sunetul obținut folosind modelul realizat de noi pentru al treilea microfon este foarte diferit de celelalte microfoane din cameră, ajungând cu un decalaj de timp semnificativ.

În contextul unei aplicații în care sunt implicate foarte multe calcule o etapă foarte importantă a acesteia o reprezintă testarea. Acest lucru a fost realizat utilizând Unity Test Runner care folosește o integrare Unity a bibliotecii NUnit, bibliotecă open-source de testare a unităților pentru platforma .Net. Un unit test testează o singură unitate de cod. Acesta trebuie proiectat astfel încât să valideze un fragment mic, logic și să ateste că acea bucată de cod va funcționa exact aşa cum ne aşteptăm în orice moment.

Ca să ne putem asigura de corectitudinea calculelor realizate în cadrul modelului acustic au fost realizate o serie de unit teste menite să verifice anumite situații generale și particulare precum: validarea formulei de calcul a lungimii unei raze, calculul timpilor, ce se întâmplă atunci când avem 0 puncte de coliziune pentru o rază, corectitudinea formulei de calcul a intensității, validarea transformărilor din coordinate polare în carteziene, dar și invers, selecția razelor, intersecția razelor, etc. Toate aceste teste pot fi vizualizate în Figura 4.7.

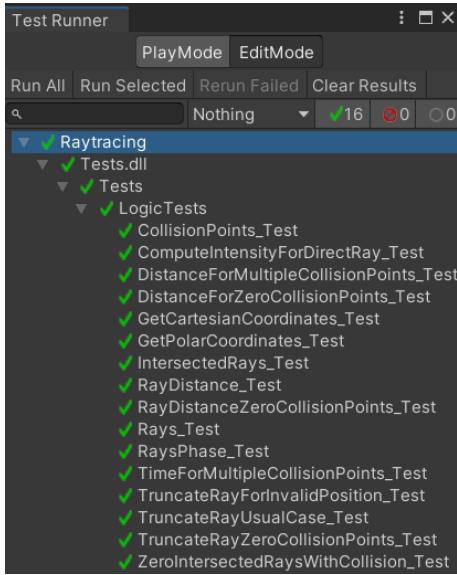


Figura 4.7: Unit testele aplicației

## 4.2 Validarea modelului acustic folosind Simcenter 3D

Atunci când vorbim despre o aplicație ce propune crearea unui model acustic pentru simularea sunetului în încăperi este foarte important să ne ridicăm problema validării acestuia folosind de exemplu un software similar existent pe piață. În cadrul acestui studiu pentru a fi verificată corectitudinea și validitatea acestui soft am realizat o serie de unit teste ce au fost prezentate în capitolul anterior și, mai mult de atât, am realizat o serie de comparații cu software-ul Simcenter 3D.

Simcenter 3D este o platformă de simulare complet integrată pentru modelarea, simularea și analizarea produselor și sistemelor complexe de inginerie. Platforma include soluții puternice de simulare pentru mai multe discipline, inclusiv analize structurale, acustice, de flux, termice, de mișcare și compoziție, precum și optimizare și simulare multifizică. Software-ul își propune să permită simulărilor să joace un rol timpuriu în procesul de proiectare, ducând la creșterea calității, eficienței și inovației.

Simcenter 3D Acoustics Modeling include toate funcționalitățile necesare pentru a crea, în mod eficient, o geometrie de domeniu fluid prin înfășurarea unei suprafete cu un mesh structural și/sau un model complet de asamblare CAD.

În cazul simulării acustice Simcenter oferă o bibliotecă extinsă de modele precise pentru estimarea surselor de zgomot aeroacustice, inclusiv modele de stare staționară, modele directe, modele de propagare și rezolvarea ecuațiilor de perturbare acustică. Zgomotul aeroacustic induc de flux este o componentă semnificativă a semnăturii acustice a unui vehicul sau a altor produse. Prezicerea și înțelegerea mecanismelor de generare a zgomotului, localizarea surselor de sunet, identificarea căilor de transmisie și prezicerea răspunsului acustic al sistemului este cheia unui bun design acustic.

Frecvențele care trebuie calculate folosind Simcenter 3D pot fi controlate în totalitate de către utilizator. Acestea pot fi definite manual sau preluate automat din vibrațiile structurale. Într-o interfață grafică de utilizator flexibilă (GUI), utilizatorul poate controla și calcula cantități acustice. Aceasta poate analiza rezultatele pentru grupuri de elemente sau microfoane individuale.

Modelul de simulare este verificat automat înainte de exportul în solver. Se verifică consistența mesh-ului, materialului și proprietăților. Feedback-ul utilizatorului este furnizat pentru a evita erorile înainte de rezolvare.

Pe lângă funcționalitățile standard de postprocesare pentru vizualizarea proprietăților acustice precum presiunea, intensitatea și viteza, scenarii specifice pot fi definite pentru a filtra și afișa în mod eficient rezultatele funcțiilor legate de microfoane.

Pentru validarea modelului acustic am creat o cameră rectangulară, folosind ambele software-uri, cu dimensiunile: 30m lățime, 30m lungime și 15m înălțime, unde am plasat sursa  $S$  la poziția  $S(0, 2.5, 0)$ , iar microfoanele au fost așezate astfel:

- $M_0(2, 1.6, 1.7)$ , fiind la o distanță de 3.07m de sursă
- $M_1(-1.5, 1.2, 1.7)$ , fiind la o distanță de 2.56m de sursă
- $M_2(1, 2, 13)$ , fiind la o distanță de 13.19m de sursă

Similaritatea celor două camere se poate observa în Figura ??.

Ambele software-uri permit crearea celor două încăperi, crearea unei surse audio și a unor microfoane statice, setarea numărului maxim de reflexii, setarea lungimii maxime a unei raze, setarea pasului de frecvență, folosirea unei melodii, setarea puterii. Diferența dintre cele două software-uri este că Simcenter nu permite setarea numărului de raze ce va fi distribuit în încăpere. Din acest motiv am încercat să găsim un echivalent pentru numărul de raze ce trebuie trasate în încăpere folosind modelul acustic propus de această lucrare. Configurația folosită va fi:

- 1 000 000 de raze distribuite uniform în cazul modelului acustic propus de această lucrare
- maxim 10 reflexii pentru o rază
- 200m distanță maximă pe care o rază o poate parcurge
- 8192Hz rezoluția frecvenței

Numărul de raze obținut de modelul vizat în lucrare este 1143, iar numărul de raze care ajunge pe microfonul  $M_2$  folosind Simcenter este 1400. Conform acestor configurații putem observa similaritatea rezultatelor obținute folosind ambele software-uri urmărind Figura 4.8, Figura 4.9, Figura 4.10 și Figura 4.11.

- versiune simcenter, detalii despre simcenter
- detalii tehnice despre simularea în Simcenter în sine, cum s-a realizat camera și ce a fost folosit pentru asta
- o poza cu modul în care arată încaperea în simcenter

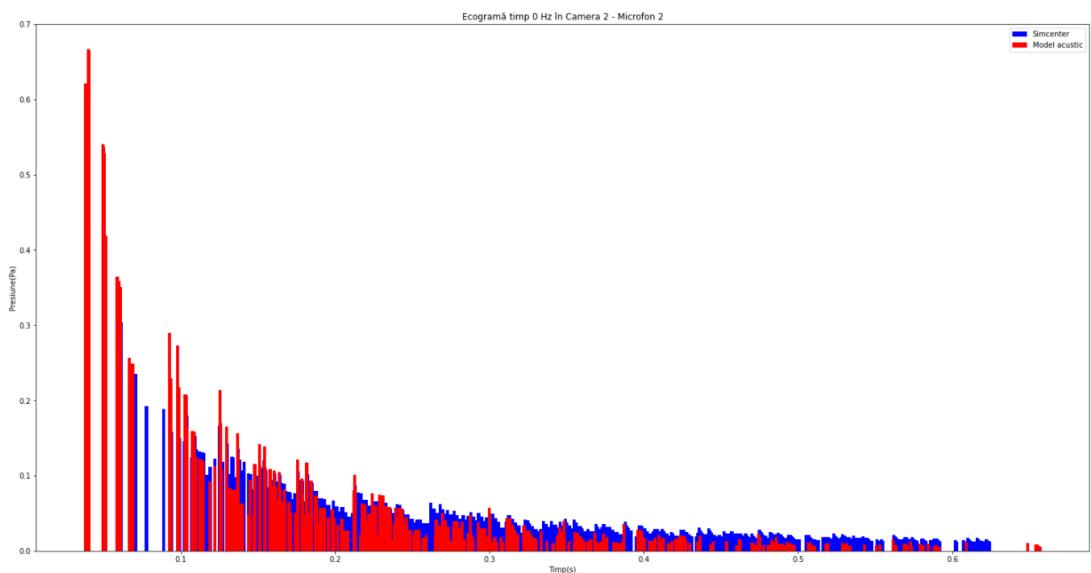


Figura 4.8: Ecogramă timp pe microfonul  $M_2$

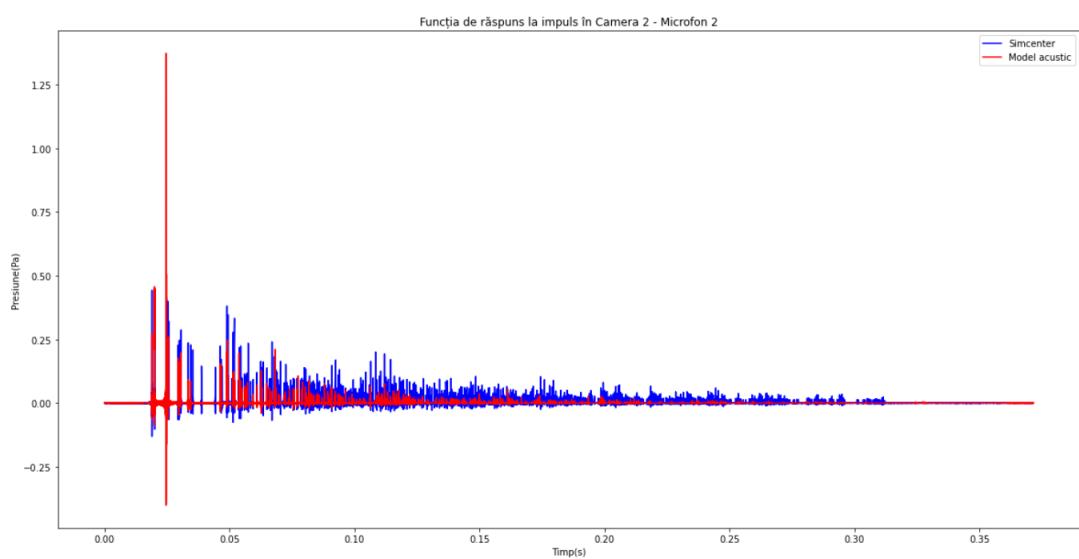


Figura 4.9: Functia de răspuns la impuls pe  $M_2$

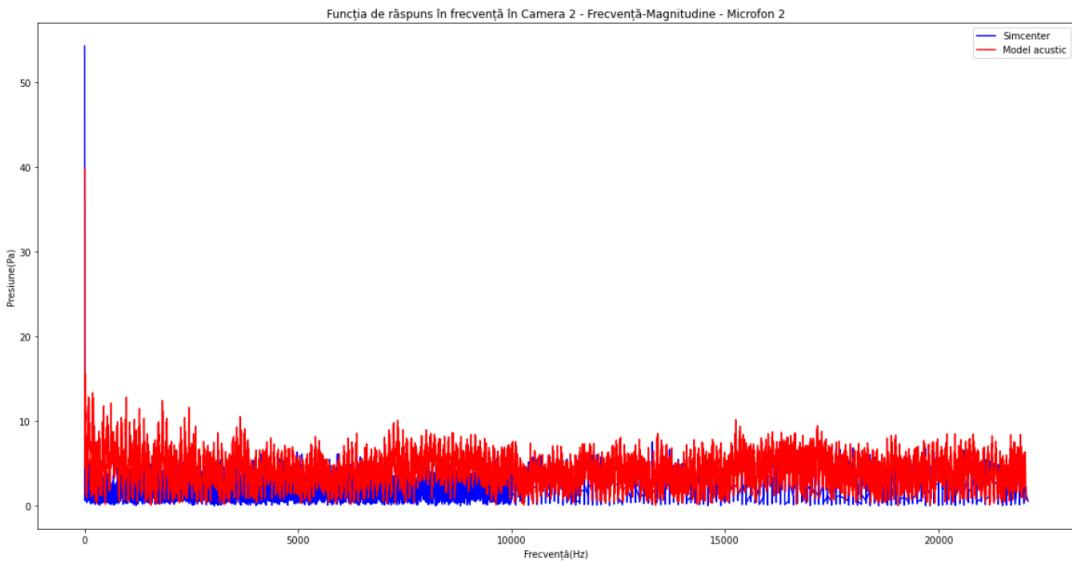


Figura 4.10: Grafic Frecvență-Magnitudine pentru  $M2$

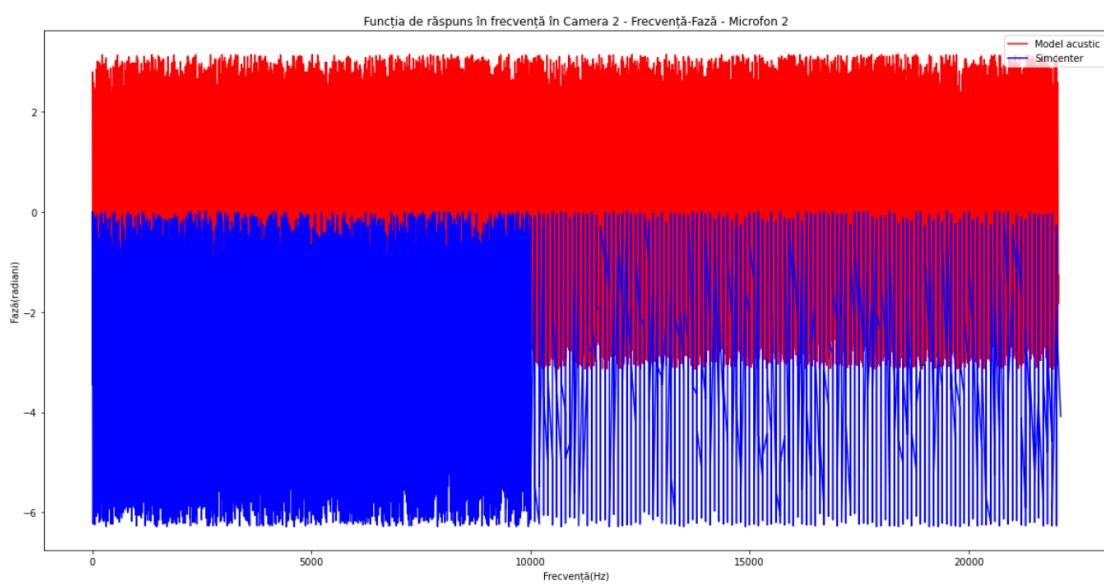


Figura 4.11: Grafic Frecvență-Fază pentru  $M2$

# Capitolul 5

## Concluzii

### 5.1 Concluzii generale

Acest studiu propune un model acustic care vine în sprijinul inginerilor acustici pentru a-i ajuta să poziționeze sursele audio și microfoanele atunci când creează încăperi precum hale, amfiteatre, aeroporturi în parametrii acustici normali.

Modelul acustic propus de noi este unul eficient, care poate fi adaptat și utilizat în contextul oricărui spațiu interior pentru a ajuta la construirea unei camere cu parametrii optimi pentru a crea confortul de care oamenii au nevoie.

Această lucrare necesită ca etapele să fie efectuate secvențial, deoarece ieșirea unui pas este intrarea pasului următor. Calculul geometric este deosebit de important deoarece dictează adesea eficiența algoritmului. Conform complexităților de timp prezentate anterior, putem observa că calculul geometric și calculul fizic au complexitate de timp liniară, iar post-procesarea datelor se realizează în  $n \log_2 n$  operații.

Modelul acustic este unul standalone, deci permite decuplarea sa de interfață realizată cu ajutorul platformei Unity și folosirea acestuia în funcție de dorințele și nevoile utilizatorului. Interfața aplicației permite vizualizarea tuturor razelor de pe microfoane, dar și vizualizarea unei singure raze.

În cadrul aplicației au fost realizate teste pentru a verifica corectitudinea calculelor și anumite situații particulare folosind unit teste și, mai mult de atât, a fost și validată folosind Simcenter 3D, o platformă de simulare complet integrată pentru modelarea, simularea și analizarea produselor și sistemelor complexe de inginerie.

Aplicația realizată nu se limitează doar la prezent, ci este o aplicație pentru care se pot face multiple îmbunătățiri prin realizarea unei simulări binaurale sau adăugarea fenomenului de difracție, dar și altele, fiind astfel orientată spre viitor oferind posibilitatea de dezvoltare continuă.



### 5.2 Concluzii personale

Mi-am dorit ca prin această lucrare să modelez o soluție pentru o problemă inspirată din realitate, care să fie de actualitate și să necesite un proces de învățare și dezvoltare continuu. Am reușit să dobândesc noțiuni din domeniul ingineriei acustice pornind de la ce este sunetul până la a înțelege cum se propagă sunetul în lumea reală și ce fenomene produce acesta, pentru a putea modela aceste comportamente fizice în contextul aplicației mele.

A fost cu adevărat o lucrare provocatoare care m-a determinat să învăț atât noțiuni teoretice, cât și cum se folosește platforma Unity și cum pot integra alte biblioteci într-o aplicație Unity, să ies din sfera mea de confort și să dezvolt lucruri cu adevărat impresionante din punct de vedere programatic și ingineresc.

Evident, au existat momente în care am întâmpinat greutăți în dezvoltarea acesteia, printre aceste momente se numără: pasul în care am trecut de la realizarea calculelor în domeniul timpului la realizarea calculelor în domeniul frecvențelor și calculul intensităților. Poate cel mai dificil pas a fost realizarea GUI-ului, pentru că nu cunoșteam platforma Unity suficient de bine pentru a ști cum se realizează meniuurile și a fost nevoie de mult timp pentru a învăța cum pot face acest lucru astfel încât acesta să fie redimensionabil.

Este o aplicație pe care îmi doresc să o dezvolt și în continuare pentru că aceasta este situată într-un domeniu care permite acest lucru prin adăugarea unor elemente precum difracția, simularea binaurală și chiar îmbunătățirea acesteia din punct de vedere al timpului de execuție prin schimbarea modului de calcul geometric sau mutarea calcului pe GPU. Toate aceste elemente aducând modelul acustic mai aproape de realitate.

## 5.3 Dezvoltări ulterioare

Ca orice model acustic, acesta poate include îmbunătățiri precum abordarea altor tehnici de calcul geometric. O posibilă soluție ar putea fi începerea trasării razelor pornind de la microfoane la sursa audio. O altă soluție ar putea fi păstrarea distribuției razelor cu modificarea că atunci când o rază întâlnește o suprafață se sparge în mai multe raze. Mai mult decât îmbunătățirea modului de simulare a geometriei camerei, se poate realiza o îmbunătățire din punct de vedere al timpului computațional al aplicației, astfel încât toate operațiunile ar putea fi mutate pe GPU.

Mai mult, se poate considera introducerea fenomenului de difracție în aplicație, care reprezintă dispersarea locală în direcția propagării undelor sonore trecând de marginea unui obiect.

Se poate considera și simularea binaurală care reprezintă o metodă de a realiza semnale binaurale pentru ambele urechi ale receptorului în spații inexistente prin intermediul unui model. Pentru fiecare microfon se vor considera două puncte în locul unuia ca și cum ar fi urechile omului.

O altă abordare posibilă ar fi să ținem cont și de absorbția aerului, nu doar de absorbția suprafeteelor din încăperi, fapt care ar putea îmbunătății rezultatele pentru încăperile de dimensiuni mari sau atunci când vorbim despre frecvențe înalte.

# Bibliografie

- [1] David Oliva Elorza, *Room acoustics modeling using the raytracing method: implementation and evaluation*, Licentiate Thesis, University of Turku, 2005
- [2] Steve Haynal, Heidi Haynal, *Generating and Searching Families of FFT Algorithms*, Journal on Satisfiability, Boolean Modeling and Computation 7, 2011
- [3] Jeong Cheol-Ho, Ih Jeong-Guon, Rindel Jens Holger, *Consideration of Wall Reflection and Diffraction in the Room Acoustic Prediction Using the Phased Beam Tracing Method*, 9th Western Pacific Acoustics Conference, Seoul, Korea, 2007
- [4] Alexander Sengpiel, *Conversion: Sound pressure to Sound intensity and vice versa*, Tutorium, Berlin, 2013
- [5] Jonathan Andrew Hargreaves, Yiu Wai Lam, *An Energy Interpretation of the Kirchhoff-Helmholtz Boundary Integral Equation and its Application to Sound Field Synthesis*, Acta Acustica united with Acustica, 2014
- [6] Axon Samuel, *Unity at 10: For better- or worse- game development has never been easier*, Ars Technica, 2016
- [7] Christoffer A. Weitze, Clau Lynge Christensen, Jens Holger Rindel, Anders Christian Gade, *Computer Simulation of the Acoustics of Mosques and Byzantine Churches*, 17th ICA, Rome, Italy (2001)
- [8] Jonathan Andrew Hargreaves, Yiu Wai Lam, *An Energy Interpretation of the Kirchhoff-Helmholtz Boundary Integral Equation and its Application to Sound Field Synthesis*, Acta Acustica united with Acustica (2014)
- [9] Lucian M. Sasu, *Medii vizuale de programare*, 2020
- [10] Effect of temperature on properties of air, 24 noiembrie 2020  
[https://en.wikipedia.org/wiki/Speed\\_of\\_sound](https://en.wikipedia.org/wiki/Speed_of_sound)
- [11] Legea lui Snell- Snell's law, 8 decembrie 2020  
[https://ro.qaz.wiki/wiki/Snell%27s\\_Law](https://ro.qaz.wiki/wiki/Snell%27s_Law)
- [12] How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice, 14 decembrie 2020  
<http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/>

- [13] Physics.Raycast din Unity, 15 decembrie 2020  
<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>
- [14] Intersecția unei sfere cu o linie, 22 decembrie 2020  
[http://www.ambrsoft.com/TrigoCalc/Sphere/SpherLineIntersection\\_.html](http://www.ambrsoft.com/TrigoCalc/Sphere/SpherLineIntersection_.html)
- [15] Legea Pătratului Invers, 25 decembrie 2020,  
[https://ro.xcv.wiki/wiki/Inverse-square\\_law](https://ro.xcv.wiki/wiki/Inverse-square_law)
- [16] Biblioteca matplotlib, 20 ianuarie 2021  
<https://pypi.org/project/matplotlib/>
- [17] Biblioteca NWaves, 3 decembrie 2020  
<https://github.com/gkngkc/UnityStandaloneFileBrowser>
- [18] Biblioteca StandaloneFileBrowser, 10 decembrie 2020  
<https://github.com/gkngkc/UnityStandaloneFileBrowser>
- [19] Biblioteca XCharts, 15 decembrie 2020  
<https://github.com/monitor1394/unity-ugui-XCharts/blob/master/Assets/XCharts/README.md>
- [20] O scurtă istoria a acusticii, 20 august 2020  
<https://www.britannica.com/science/acoustics>
- [21] Proiectul Triton realizat de Microsoft, 20 iulie 2020  
<https://www.microsoft.com/en-us/research/project/project-triton/>
- [22] Simcenter 3D, 2 ianuarie 2021  
<https://www.plm.automation.siemens.com/global/en/products/simcenter/>