

はじめての
音響数値シミュレーション
プログラミングガイド

日本建築学会 編

コロナ社

本書関係委員

建築音響数值解析環境整備 [若手奨励] 特別研究委員会 (2009年4月-2011年3月)

委員長 大嶋 拓也
幹事 石塚 崇
委員 江田 和司 大久保 寛 岡本 則子
奥園 健 鈴木 久晴 谷川 将規
富来 礼次 星 和磨

執筆担当者

第1章 大嶋 拓也
第2章 2.1 大嶋 拓也
2.2 鈴木 久晴 大嶋 拓也
2.3 大嶋 拓也
2.4 大嶋 拓也
第3章 鈴木 久晴
第4章 石塚 崇
第5章 鈴木 久晴
第6章 大久保 寛
第7章 星 和磨
付録 A.1 大嶋 拓也
A.2 星 和磨
A.3 石塚 崇

まえがき

簡単に本書のご紹介と、執筆に至った経緯を述べてみようと思います。本書では、音響数値計算のための数値シミュレーションの手法をいくつか取り上げ、その特徴、基礎理論と定式化、コーディングについて説明しています。対象としている読者は、理工系を専門とする大学4年生以上の方々です。取り上げている計算手法は

- 有限要素法
- 境界要素法
- 時間領域有限差分法
- CIP法
- 音線法

の5つです。学生時代に私どもが数値計算に取り組んだときは、何だか敷居の高い理論書と、何だか敷居の高いプログラミング言語しかなくて、こんなに多くの解析手法を流して学ぶのはけっこう大変でした。もっと概要が簡単にわかって、簡単なコードも載っていて、簡単な解析ならそのコードを改造していけばできてしまう、そんな本があればいいのに、との願望がありました。

それから少し時間も経って、環境はだいぶ変わり、理論書もわかりやすいものが出てきて、コンピュータも速くなって、プログラミング言語も可読性と速度をある程度両立するものも出てきました。もちろん、オープンソースという社会の動きの盛り上がりもありました。今なら全体像を簡単に把握できる、ドキュメントやサンプルコードが作れるのではないかとの欲望に変わってきました。

そんな話を、有志のメンバーで話している間に、[若手奨励]特別研究委員会公募制度を知り、公募が採択されて建築音響数値解析環境整備の特別研究委員会という形で活動を行うことができ、執筆に至った、というのがこのプロジェクトの開始経緯です。この委員会では終始、どんな音響数値シミュレーションの本なら読みたくなるか、じっくり議論しながら書くことができました。そんな経緯もあって、本書では

- はじめて数値計算に取り組む方でも短時間で手法の概要をつかめる
- 簡単なサンプルコードが付いていて、計算を走らせることができる

ということを目的にしています。解析理論の説明については数学的な正確さはもちろん心がけていますが、大まかな論理の流れを提供し、コーディングに支障のない程度に記述することを最優先としています。また、コーディングにはPythonという言語を採用しました。Pythonを採用したのは、言語としての理解のしやすさはもちろんですが、ScipyやMatplotlibなどの科学技術計算用のパッケージが揃っているのが理由です。もちろん、Pythonはすべてオープンソースで作られていてコミュニティも活発なので、無料で言語の処理系を入手できて、覚えた後も継続して使っていくことが可能です。数値解析のコーディングでは

- 節点情報、要素情報を保持するデータ構造
- 密行列・疎行列などの行列データの取扱い
- 逆行列を数値的に解く(それも、密行列・疎行列の)
- 解かれた節点や格子点の物理量に応じた可視化

などの実装テクニックが必要となりますが、それらをすべて自分で組んでいくのは大変です。PythonにSciPyやmatplotlibなどのパッケージを導入していくことで、それらを自分で書く時間を節約できますし、何より解析をしたいという意欲がそがれることもありません。

本書を読んだ後に

- 何となく理論がわかったな
- 自分の解析のためのコーディングを始めてみようかな

という感覚を持っていただければ、本書の目的は達成かな、と思っています。本書で用いているサンプルコードやライブラリは(<http://www.openacoustics.org>)というURLで公開しています。サポートフォーラムも準備していますので、読んでいて何か質問や意見があれば、いつでも著者らにコンタクトを取ることができます。

本書が、読者のみなさまのお役に立てれば、これ以上の喜びはありません。

2012年11月

日本建築学会 建築音響数値解析環境整備 [若手奨励] 特別研究委員会

目 次

1. はじめに

1.1 音響数値シミュレーション手法の紹介	1
1.2 古典的手法	2
1.2.1 波動音響学的手法	2
1.2.2 幾何音響学的手法	4
1.3 比較的新しい手法	4
1.3.1 時間領域有限要素法	5
1.3.2 高速多重極境界要素法	6
1.3.3 CIP 法	6
1.4 本書の概要	6
1.5 波動方程式の導出	7
1.5.1 連続の式	7
1.5.2 運動方程式	9
1.5.3 波動方程式	10
引用・参考文献	11

2. Python, NumPy, および SciPy 入門

2.1 はじめに	13
2.2 Python について	13
2.2.1 実行と基本構文	14
2.2.2 関数の定義と呼び出し	16
2.2.3 パッケージのインポート	17
2.2.4 変数とデータ型	18
2.2.5 数値演算	18
2.2.6 コンテナ型	19

2.2.7 制 御 文	20
2.2.8 文 字 列 操 作	21
2.3 NumPy	22
2.3.1 NumPy パッケージの使用	22
2.3.2 NumPy 配列の作成	22
2.3.3 NumPy 配列要素へのアクセス	24
2.3.4 配 列 の 演 算	26
2.3.5 定 数	27
2.3.6 格子点の生成	27
2.4 SciPy	28
2.4.1 SciPy パッケージの使用	28
2.4.2 NumPy との関係	28
2.4.3 疎行列の取扱い	29
2.4.4 特 殊 関 数	30
2.4.5 数 値 積 分	30
引用・参考文献	31

3. 有限要素法

3.1 有限要素法の数学的基礎	32
3.1.1 音場の離散化	32
3.1.2 基礎積分方程式	33
3.1.3 離 散 化	34
3.2 コーディングの基礎 -sampleFem1.py-	39
3.2.1 音場の形を決め、節点と要素に分解する	39
3.2.2 要素ごとに積分計算を行い、要素マトリクスを計算する	41
3.2.3 要素マトリクスを重ね合わせ、全体マトリクスを構築する	42
3.2.4 マトリクス方程式を解き、各節点での音圧を計算する	43
3.3 コーディングの応用 -sampleFem2.py-	45
3.3.1 メッシュを使った離散化	45
3.3.2 可 視 化	47
引用・参考文献	53

4. 境界要素法

4.1 境界要素法の数学的基礎	54
4.1.1 手法の概要	54
4.1.2 基礎積分方程式	56
4.1.3 離散化	60
4.1.4 境界積分	63
4.2 コーディングの基礎 -sampleBem1.py-	70
4.2.1 モジュールのインポート	72
4.2.2 解析条件の初期設定	73
4.2.3 関数群の定義	74
4.2.4 連立方程式の構築	76
4.2.5 連立方程式を解く	80
4.2.6 観測点における音圧の計算	80
4.2.7 2次元境界要素法サンプルコード1	82
4.3 コーディングの応用 -sampleBem2.py-	85
4.3.1 外部メッシュを使った離散化	85
4.3.2 被積分関数の効率化	89
4.3.3 可視化	90
4.3.4 2次元境界要素法サンプルコード2	94
引用・参考文献	98

5. 時間領域有限差分法

5.1 時間領域有限差分法の数学的基礎	100
5.1.1 特徴	100
5.1.2 音場の離散化	101
5.1.3 基礎方程式と離散化	102
5.1.4 音源条件	104
5.1.5 境界条件	106
5.2 コーディングの基礎 -sampleFDTD1.py-	107

5.2.1	音圧, 粒子速度について格子点を確保する	107
5.2.2	格子点ごとに更新式を実行し, 次時刻の音響量を計算する	108
5.2.3	音源位置の格子点に音圧波形を与え, 目的の時間波形を得る	110
5.2.4	可視化	111
5.3	コーディングの応用 -sampleFDTD2.py-	114
5.3.1	省メモリ化	115
5.3.2	インピーダンス境界条件	115
5.3.3	より進んだ可視化	117
5.3.4	高速化	118
	引用・参考文献	121

6. CIP(constrained interpolation profile) 法

6.1	CIP 法の数学的基礎	122
6.1.1	特徴	122
6.1.2	数値分散と数値散逸による誤差について	123
6.1.3	外部吸収境界について	125
6.1.4	支配方程式と特性曲線法(移流方程式)を用いた定式化	125
6.1.5	CIP 法による 2 次元音場解析のための離散化	128
6.1.6	CIP 法の計算手順のまとめ	132
6.1.7	音源設定	136
6.1.8	境界条件	136
6.2	コーディングの基礎 -sampleCIP1.py-	137
6.2.1	音圧, 粒子速度, および特性曲線について格子点分の変数を確保する	138
6.2.2	初期音圧分布を与える	139
6.2.3	格子点ごとに音圧, 粒子速度から特性曲線を計算する	140
6.2.4	格子点ごとに更新式を実行し, つぎの時刻の特性曲線を計算する	140
6.2.5	格子点ごとに特性曲線から音圧, 粒子速度を計算する	142
6.3	コーディングの応用 -sampleCIP2.py-	151
6.3.1	計算量の削減のための工夫	151
6.3.2	2 次元音場のスナップショット表示	152
6.4	コーディングの応用 -sampleCIP3.py-	157

引用・参考文献	164
---------	-----

7. 音線法

7.1 音線法に必要となる幾何音響学の基礎	166
7.1.1 音波の表現	166
7.1.2 境界面における音波の振る舞い	168
7.1.3 波動性の考慮	169
7.2 幾何音響学に基づくシミュレーション	169
7.3 音線法を実行するうえでの注意点	170
7.3.1 音粒子の数, 受音エリアの大きさ, および室容積の関係	171
7.3.2 入力形状と音波の関係	173
7.4 プログラミングの前に	173
7.5 コーディングの基礎 -sampleRay1.py-	177
7.5.1 グローバル変数	178
7.5.2 メインルーチン	179
7.5.3 関数の定義	184
7.5.4 音源の生成	189
7.5.5 sampleRay1.py 実行例	191
7.5.6 sampleRay1.py 全プログラム	192
7.6 コーディングの応用 -sampleRay2.py-	201
7.6.1 プログラムの概要	201
7.6.2 sampleRay2.py メインルーチンプログラム	204
7.6.3 sampleRay2.py 実行例	206
引用・参考文献	207

付録	208
----	-----

A.1 Windows への OpenAcoustics パッケージのインストール	208
A.1.1 Python のインストール	208
A.1.2 NumPy, SciPy, および matplotlib のインストール	209
A.1.3 OpenAcoustics パッケージのインストール	209
A.2 Mac OS X への OpenAcoustics パッケージのインストール	210

A.2.1 OS のバージョンと Python のバージョン関係	210
A.2.2 Python, NumPy, SciPy, および matplotlib のインストール	210
A.2.3 Gmsh のインストールと設定	211
A.2.4 OpenAcoustics パッケージのインストール	211
A.3 Linux への OpenAcoustics パッケージのインストール	212
A.3.1 Ubuntu	212
A.3.2 NumPy, SciPy, および matplotlib のインストール	212
A.3.3 Gmsh のインストール	212
A.3.4 OpenAcoustics パッケージのインストール	213
引用・参考文献	214
索 引	215

1 はじめに

1.1 音響数値シミュレーション手法の紹介

音響数値シミュレーション手法は、音響伝搬における音の回折、干渉のような、いわゆる音の波動性の考慮の有無によって

- 波動音響学的手法：音の波動的な性質を含めた、音響伝搬現象の微視的なモデル化による手法
- 幾何音響学的手法：音の波動性を無視し、音響エネルギーの伝搬を幾何学的にモデル化する手法

に大別される。このうち、前者の波動音響学的手法はさらに、以下に大別される。

- 周波数領域解法：ある周波数における定常解を求める手法
- 時間領域解法：過渡的な音響伝搬現象の進行を時々刻々、アニメーションのコマのように時間を進めながら追う手法

前者の代表例としては境界要素法 (boundary element method, BEM) および有限要素法 (finite element method, FEM)、後者の代表例としては時間領域有限差分法 (finite-difference time-domain method, FDTD 法) があげられる。これら 3 手法はいずれも 1990 年代前半までにはおおむね基礎が築かれた、いわば古典的手法といえる。しかしながらそれで終わることなく、今世紀に入っても今日に至るまで、大規模問題解析のための手法開発^{1),2)} や、既存

表 1.1 おもな音響数値シミュレーション手法の一覧
(太字は本書で扱う手法)

		古典的手法	比較的新しい手法
波動音響学的手法	周波数領域解法	有限要素法 境界要素法	高速多重極境界要素法
	時間領域解法	時間領域有限差分法	時間領域有限要素法 CIP 法
幾何音響学的手法		虚像法 音線法	

の解法を組み合わせ、特定の問題により即した手法の開発^{3),4)}が続けられている。

一方、後者の幾何音響学的手法は、1960年代後半から導入が図られてきた、音響数値シミュレーション手法としては最も歴史の長い手法群といえる⁵⁾。

以上の議論から、現在知られている音響数値シミュレーション手法を整理して示すと、表1.1のとおりであり、バリエーションに富んだ手法群となっている（表中の太字は本書で扱う手法である）。1.2節では、これら各種手法のうち20世紀中に完成された比較的古典的な手法を、1.3節ではそれ以降の比較的新しい手法を概説する。

1.2 古典的手法

1.2.1 波動音響学的手法

波動音響学的手法は、計算力学分野全般において適用される汎用的な離散化手法によって後述の波動方程式を離散化し、音の波動的な振る舞いを予測する手法である。前述のとおり、前世紀中におおむね完成された古典的な数値シミュレーション手法の代表として、周波数領域解法では有限要素法および境界要素法、時間領域解法では時間領域有限差分法があげられる。いずれもシミュレーションを行う対象となる音場領域、ないしはその境界面を小空間または面に分割して解くのが特徴であり、分割された領域または面のそれぞれは要素またはセルといい、それらの集合はメッシュと呼ばれる（図1.1）。

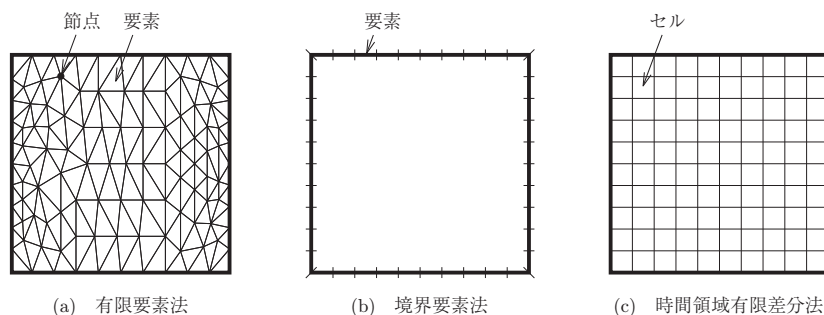


図 1.1 正方形音場のメッシュ例

これら古典的手法はいずれも汎用性が高く、適切な境界条件と組み合わせることで、幅広い問題に適用可能である。計算力学全般において確立された理論的基盤の適用、原理的な精度の高さ、および汎用性の高さといった点から、これらの波動音響学的手法、および1.3節以降に述べるこれらをルーツとする各種発展手法の適用事例は大幅に増加しつつある⁶⁾。

(1) **有限要素法** 有限要素法では、本来、無限の自由度（問題規模）を有する対象音場の空間全体を、図 1.1(a) に示したように有限個の要素に分割し、要素ごとに定義された節点において音場を表現する。系の振る舞いは、質量行列および剛性行列によって記述され、もっぱら周波数領域における表現が用いられる。解析領域全体を要素分割する必要があるため、自由度は大きくなりがちであるが、系の振る舞いを記述する行列は疎行列であるため、解法の工夫によって演算負荷の低減が可能である⁷⁾。開領域の扱いには無限要素導入などの工夫が必要であるため、閉空間への適用が主となる⁸⁾。

有限要素法は、最も汎用的な計算工学的手法のひとつであるゆえ、メッシュ生成などの前処理（プリプロセッシング）および可視化などの後処理（ポストプロセッシング）ソフトウェアが充実しており、その展開も連成問題や設計変数の最適化への適用⁹⁾ など多様である。

(2) **境界要素法** 境界要素法では、波動方程式を対象領域空間の境界面における積分方程式に変換して、おもに周波数領域で解く。図 1.1(b) に示したように、対象空間の境界面のみメッシュ分割すればよいため、メッシュ分割が比較的容易であること、開領域問題のための特別な境界条件が不要であることが特徴といえる¹⁰⁾。境界面のみメッシュ分割を行うため、空間全体をメッシュ分割する有限要素法などと比較して自由度は大幅に低減されるが、伝搬系を表すマトリクスが密行列となるため行列計算の演算量が多く、演算負荷上は他手法と比較して必ずしも有利ではない。

ただし、1.3.2 項に述べる高速多重極境界要素法が近年になって導入されたことにより、演算負荷上の問題が大幅に低減された。それによって一気に実用性の高い手法となりつつある。

(3) **時間領域有限差分法** 時間領域解法とは、音場の初期状態からの過渡的な現象の推移を時間発展的に解く手法であり、解析結果の可視化、また、音響数値シミュレーション独特の解析結果提示手法である可聴化に優れる。時間領域有限差分法は、時間領域解法の代表的なものであり、音響伝搬媒質に関する連続の式、および運動方程式を、図 1.1(c) に示したような直交メッシュ上で、時々刻々少しずつ時間を進めながら解く手法である。静止媒質を前提とするため、両式ともに簡単な形式であり、コーディングが容易である。さらに、音圧および粒子速度を、格子上および時間軸上で交互に配置するスタガード格子を利用し、時間方向を陽的に積分するため、有限要素法や境界要素法で必要な音響伝搬系を表すマトリクスの構成および求解が不要である。そのため、波動音響学的手法としては、比較的少ない計算機資源で解析が可能である。

一方、基本的に直交メッシュを用いてシミュレーションを行うため、不整形室などの複雑形状の解析に工夫を要すること、波長当りの必要格子数が他の手法と比較して多めとなる（1 波長に 10～20 セルが必要）のが、短所といえる。

1.2.2 幾何音響学的手法

幾何音響学的手法の主要なものとしては、音源から放射された多数の音線の軌跡を追跡する音線法、境界面における音の反射によって生成される音源の虚像を順次求める虚像法などがあげられる。両者の図式的表現については、後出第7章の図7.6を参照されたい。

波動性に根ざす現象の再現は（特にそれらを考慮した手法を導入しない限り）不可能であり、それゆえ厳然たる理論的背景を有するとはいえないとされる¹¹⁾。しかしながら、少ない計算機資源によって、大規模空間における音響伝搬をシミュレート可能であり、ヨーロッパでは、商用の幾何音響シミュレーションソフトウェアを設計ツールとして用いた建築音響設計が盛んである¹²⁾。

前項で述べたように歴史の長い手法であることから、純然たる幾何音響学の計算手法論としての研究は、もはや今日的でないといえる。しかしながら、前処理段階における幾何的複雑度の自動制御による周波数別の拡散性考慮¹³⁾、幾何的要素自動低減の研究¹⁴⁾など、シミュレーションを支援する周辺技術との接点において、いまだに研究が続いている。

(1) 音線法 音線法は、音源から多数の音響エネルギー粒子を全方向に放出し、時々刻々粒子を進行させながら、壁面での反射履歴およびエネルギー減衰を追跡する手法である。虚像法と比較すると、反射音の到来方向、到達時間が必ずしも正確でない¹²⁾。しかしながら、計算負荷が壁面数と音線数の積に対して比例関係であるため、虚像法と比較して長時間の音場計算が容易であり、さまざまな問題に対する汎用性が高い。とりわけ、残響減衰過程のシミュレーションに有効とされている¹⁵⁾。

(2) 虚像法 虚像法は閉空間内の音源に対し、空間を囲む壁面を鏡面と見なして虚像の位置を多重反射にわたって求めていく手法であり、実音源と虚音源群からの音の和により、空間内の音の強さが求められる。虚像法は音線法と比較して、反射音の到来方向、到達時間などが正確に求まるため、建築音響設計における初期反射音構造やエコー障害の検討に有効とされている。反面、反射次数の増加に伴って計算負荷が指数関数的に増大するため、高次の虚音源追跡は困難である¹²⁾。

1.3 比較的新しい手法

以上に見たような汎用の音響数値シミュレーション技術に関する研究は、前世紀におおむね完成された感がある。今世紀に入ってからシミュレーション技術開発は、おおむね

1. これら古典的手法の大規模解析への特化
2. 特定のアプリケーションのための手法開発

の2点を大きな柱としているように思われる。以下ではこのうち、前者について解説する。

波動音響学に基づいた汎用手法に共通する欠点として、自由度の2~3乗の計算負荷が要求され、それゆえ問題が大規模となるほど膨大な計算機資源が必要となることがあげられる。表1.2は、有限要素法および境界要素法における必要計算機資源を、解析対象の代表寸法 L および波数 k 、有限要素法および境界要素法における解析自由度 $N_f \cdot N_b$ の関係で表している⁵⁾。有限要素法で直接解法を用いた場合、計算時間は自由度の2乗、メモリに関しては自由度に比例したオーダーの所要量となる。解析対象の代表寸法との比較で見れば、空間全体を要素分割するため代表寸法のそれぞれ6乗、3乗のオーダーとなる。一方、境界要素法で直接解法を用いた場合、それぞれ自由度の3乗、2乗のオーダーであり、代表寸法に対しては、境界面のみ要素分割するためそれぞれ6乗、4乗のオーダーとなる。このように、大規模になるほど急速に増大する、必要な計算機資源の問題に対して、いくつかの汎用手法が近年新たに提案されている。

表 1.2 有限要素法・境界要素法の演算負荷⁵⁾

手法	自由度	演算量	メモリ使用量
有限要素法	直接解法	$O(N_f^2) \sim O((kL)^6)$	$O(N_f) \sim O((kL)^3)$
	反復解法	$O(N_f) \sim O((kL)^3)$	
境界要素法	直接解法	$O(N_b^3) \sim O((kL)^6)$	$O(N_b^2) \sim O((kL)^4)$
	反復解法	$O(N_b^2) \sim O((kL)^4)$	
	高速多重極境界要素法	$O(N_b) \sim O((kL)^2)$	$O(N_b) \sim O((kL)^2)$

1.3.1 時間領域有限要素法

有限要素法による音場解析は、振動問題との連成、モード解析との親和性から周波数領域における適用が主であったが、近年は時間応答を算出可能な時間領域有限要素法 (time-domain finite element method, TD-FEM) が注目されている。その時間領域有限要素法による大規模音場解析を実現すべく、効率的な反復解法の導入が行われている^{16),17)}。表 1.2 にも示したように、有限要素法における反復解法は直接解法と比較して、演算量を自由度に比例するオーダーまで低減が可能であるが、さらに、反復解法のなかでも効率的かつ自由度に対する反復回数の安定した手法を使用することで、演算量の自由度に対する良好な比例係数が得られることを示している。また、大規模計算に不可欠である並列計算の導入についても、128 プロセッサまでの良好な並列スケールリングを報告している²⁾。

1.3.2 高速多重極境界要素法

有限要素法では、解析対象空間全体の要素分割が必要なのに対し、境界要素法による解析では、領域境界面の要素分割のみでよい。したがって、大規模複雑形状となるほど有限要素法に対するメッシュ生成の負荷が小さくなるのが、境界要素法を実務へ適用するうえでの大きな利点のひとつとされる。しかしながら、計算負荷については表 1.2 に示したように反復解法を導入しても有限要素法より不利であった。それに対して高速多重極境界要素法¹⁾では、演算量およびメモリ使用量ともにおおむね自由度に比例するオーダーまでの低減を図っており¹⁸⁾、境界要素法の大規模問題への適用性を大幅に改善している。

1.3.3 CIP 法

従来の FDTD 法による時間領域解析では、安定条件による制約から、きわめて小さな時間ステップを取る必要があり、長時間応答の計算においては大きな計算負荷となっていた。それに対して各種の改善手法が提案されており、CIP (constrained interpolation profile) 法^{19),20)}もそのひとつである。物理量とともにその空間微分値を陽に計算することで、時間進展に伴う数値分散 (位相誤差) の蓄積を大幅に抑えているのが特徴である。また、メモリ使用量に対する演算量が比較的多く、計算のほとんどが単純な積和演算に帰着するため、近年の計算機ハードウェアに見られる高速ベクトル積和演算機構との親和性が高いのが特徴である²¹⁾。ただし、境界においても空間微分を考慮する必要があり、FDTD 法と比較すると境界条件の取扱いはやや複雑である。

1.4 本書の概要

以上のように、現代の音響数値シミュレーション手法は実にバリエーションに富んでいる。これらのうち本書では、古典的かつ最も主要な波動音響学的手法である有限要素法、境界要素法、時間領域有限差分法の 3 手法、および汎用性の高い幾何音響学的手法として音線法、さらに、比較的新しいが、今後の発展が期待される波動音響学的手法として CIP 法を取り上げる。

これら各手法に対して、まずはじめに、最低限の数学的議論から、プログラミングに即した数式である離散化式の導出を行う。なお、各手法の詳細な理論的背景については、他書に譲る。例えば、文献¹⁵⁾を参照されたい。

ついで、離散化式をプログラムへと変換する過程であるコーディングを、コーディング例を示しながら詳細に解説する。さらに例題 1 (コーディングの基礎) として、手計算でも追試可能な最小限の問題を設定し、途中計算の数値も示しながらシミュレーション手法の流れを

索引

【あ】			
圧力	10	可聴化	3
後処理	3, 7	Galerkin 法	32
アニメーション	117	関数型	18
安定条件	6	完全反射性境界	61, 86
		【き】	
【い】		幾何音響学 (的手法)	1, 4, 6, 166
位相誤差	6	気体媒質	7
移流方程式	126, 140	逆問題	166
インタプリタ	7, 14	吸音性境界	61, 86
インデント	15	吸音率	168
インピーダンス境界	33, 101	境界条件	2, 61, 73, 87
		境界積分方程式	55, 59
【う】		境界要素	60, 73
運動方程式	3, 7, 9, 102, 125	境界要素のサイズ	61, 87
		境界要素法	1, 3, 6, 30, 54
【え】		鏡像法	169
エコー障害	4	鏡面反射	168
エコータイムパターン	178, 191	虚音源	4, 170
エルミート補間	127	局所座標	64, 67
		局所作用	61
【お】		虚像法	4, 169
音粒子	166	【く】	
重み関数	33	空間微分値	124, 128
重み付き残差法	32, 57	繰返し	21
音圧	3, 7, 125	【け】	
音響アドミタンス	61, 73, 74, 87	計算時間	146
音響インピーダンス	61, 73, 168, 173	形状関数	64
音響数値シミュレーション手法	1, 6	【こ】	
音線	166, 191	後進差分	102
音線法	4, 6, 166, 169	剛性行列	3
音速	10, 126	高速多重極境界要素法	3, 6, 56
音場	2	高速ベクトル積和演算機構	6
		コーディング	6
【か】		コマンド	14
外部吸収境界	125	コマンド プロンプト	14
開領域問題	3	コメント	15
外力	10	固有音響インピーダンス	61
ガウシアンパルス	105	コンテナ型	19
仮想音源	170	コンパイラ言語	7
		コンパイル	7
		【さ】	
		サーフェスプロット	117
		最適化	3
		差分	102
		残響減衰過程	4
		散乱係数	169
		【し】	
		シェル	14
		時間領域解法	1, 3, 122, 136
		時間領域有限差分法	1, 3, 6, 100
		時間領域有限要素法	5
		試験関数	34
		辞書型	19
		実音源	4
		室情報ファイル	173
		質量行列	3
		質量保存式	7
		集合型	19
		自由度	3
		周波数領域解法	1
		条件分岐	20
		初期音圧	139
		初期反射音構造	4
		振動境界	61
		【す】	
		数値散逸	125
		数値振動	101
		数値積分	30, 65, 69, 72
		数値分散	6, 123
		スクリプト	14
		スタガード格子	3, 101
		【せ】	
		制御文	20
		整数型	18
		積分方程式	3
		積和演算	6
		節点	3, 33
		セル	2

線形補間	128, 134	倍精度実数型	23		
前進差分	102	配列	19, 22		
	【そ】	波数	5	無限要素	3
疎行圧縮行列	29	パッケージ	17		
疎行列	3, 29, 42	波動音響学 (的手法)	1, 6		【め】
疎行列オブジェクト	29	波動方程式	2, 7, 10, 32	メッシュ	2
	【た】	パルス性信号	123	メモ帳	16
ターミナル	14, 211, 212	反射境界	157		【も】
体積弾性率	8, 102, 125	反射率	168	モード解析	5
代表寸法	5	反復解法	5, 63	文字コード	16, 21
多次元配列	25		【ひ】	モジュール	17
多倍長整数型	18	微分	102	文字列型	18
タプル型	19, 22, 108		【ふ】		【ゆ】
	【ち】	ファイル出力	145	有限要素法	1, 3, 5, 6, 32
中心差分	102	ブール型	18		【よ】
直接解法	5, 63	複素数型	18	要素	2, 3, 33
直線と平面の交点	186	浮動小数点型	18		【り】
直交格子	123	部分配列	25	離散化式	6
	【て】	プリプロセッシング	3	離散化手法	2
テキストエディタ	14		【へ】	リスト型	19, 22, 23
デルタ関数	58	平面の方程式	185	粒子速度	3, 7, 8, 125
点と直線の距離	186	並列計算	5		【る】
伝搬経路	166	ベクトル化	75	ルンゲ-クッタ法	103
	【と】		【ほ】		【れ】
特異積分	66, 69, 78	法線方向微分	57, 64, 68	レイトレーシング	166
特性インピーダンス	61, 126	ポストプロセッシング	3	連結リスト	29
特性曲線法	122, 125		【ま】	連成問題	3
特性法	122	前処理	3	連想配列	19
	【に】	マトリクス方程式	38	連続の式	3, 7, 102, 125
任意次元配列	27		【み】	連立方程式のソルバ	63, 80
	【は】	密行列	3		
媒質	7	密度	7, 125		

【A】		【B】		cd コマンド	210, 211
arange() 関数	23, 24	bar() 関数	183	CIP 法	6, 122
array クラス	22	BEM	1	clock() 関数	17, 146
array() 関数	22	bool	18	colorbar() 関数	92
ASCII コード	16		【C】	complex	18
Axes3D クラス	118, 182	C 型 CIP 法	128, 132	compressed sparse row	29
		C 言語	7, 14	constrained interpolation profile	6
				contourf() 関数	92

co-located grid 122
 CP932 16
 cross() 関数 186
 CSR 形式 29
 csr_matrix クラス 29

[D]

def 文 16
 direct factorization solver 30
 dot() 関数 22, 81
 dsolve サブパッケージ 30

[E]

elif 文 20
 else 文 20

[F]

FDTD 法 1, 6, 124
 FDTD(2,4) 法 124
 FEM 1
 Fermat の法則 166
 float 18
 float64 クラス 23
 for 文 20, 21
 Fortran 言語 7, 14
 function 18

[G]

Gauss 積分法 30
 Gauss-Legendre 法 65
 getLines() メソッド 87
 getNodes() メソッド 87
 Gmsh 45, 85, 211, 212
 gmsh モジュール 87
 GMsh2D クラス 46, 87
 GMsh3D クラス 46
 Green 関数 57, 74
 Green の定理 34, 57

[H]

Hammer の公式 69
 Hankel 関数 57, 66, 74
 Helmholtz 方程式 56
 Hermite 補間 127, 130, 131

[I]

if 文 20
 import 文 17
 int 18
 integrate サブパッケージ 30

[L]

Lagrange 補間 134
 legend() 関数 184
 LIL 形式 29
 lil_matrix クラス 29
 linalg サブパッケージ 22, 28, 30, 72, 80

linalg.norm() 関数 27, 28
 linalg.solve() 関数 27, 28
 linear algebra 30
 linked list 29
 linspace サブパッケージ 22
 linspace() 関数 23, 183
 Linux 14, 212
 loadGeo() メソッド 87
 long 18

[M]

M 型 CIP 法 128, 134
 Mac OS 14, 210
 MATLAB 13, 47
 matplotlib 7, 152, 208, 209
 matplotlib パッケージ 47, 91, 111, 182
 mgrid() 関数 27, 91
 MM-MOC 法 122
 multi-moments 124

[N]

Netlib 63, 66
 norm() 関数 186
 NumPy 7, 13, 22, 208, 209
 numpy パッケージ 22
 NumPy 表記 119

[O]

OpenAcoustics 208, 209, 211

[P]

pass 文 20
 PATH 209
 python.exe 209
 pi 定数 27, 73
 plot_surface() 関数 118
 plot() 関数 92
 points per wavelength 123
 PolyCollection クラス 47
 PPW 123
 print 文 21
 pylab.cm モジュール 92

Python 7, 13, 208, 209
 python.exe 209

[Q]

quadrature() 関数 30, 72, 78

[R]

return 文 16

[S]

savefig() 関数 98
 scatter() 関数 203
 SciPy 7, 13, 28, 208, 209
 scipy パッケージ 28, 42
 scipy.sparse サブパッケージ 29
 scipy.sparse.linalg.dsolve サブパッケージ 30
 setup.py 210, 213
 shape 属性 24, 80
 show() 関数 93
 sin() 関数 22
 size 属性 24
 sleep() 関数 17
 sparse matrix solver 30
 special サブパッケージ 28, 30, 72
 spsolve() 関数 30
 Staggered grid 101
 staggered grid 123
 str 18
 subplot() 関数 183
 Subversion 209

[T]

TD-FEM 5
 time パッケージ 17
 title() 関数 92
 tocsr() 関数 29

[U]

Ubuntu 212
 unicode 18
 Unicode 体系 21
 Unicode 文字列 21
 Unicode 文字列型 18
 UTF-8 16, 21

[V]

VTK 52

【W】		【Y】		【記号, 数字】	
while 文	20	Yee	100	,	25
Windows	14, 208-210	ylabel() 関数	92	:	25
				;	209
				[]	29
【X】		【Z】		1 次元配列	23, 27
xlabel() 関数	92, 184	zeros() 関数	22, 23, 108	1 次元リスト	22
xlim() 関数	184			2 次元配列	23, 27, 29

— 著者略歴 —

大嶋 拓也 (おおしま たくや)

- 1995年 東京大学工学部建築学科卒業
- 1998年 東京大学大学院工学系研究科修士課程修了 (建築学専攻)
- 2000年 東京大学大学院工学系研究科博士課程中退 (建築学専攻)
- 2000年 新潟大学助教
現在に至る
- 2005年 博士 (環境学) 東京大学

石塚 崇 (いしづか たかし)

- 1999年 九州芸術工科大学芸術工学部音響設計学科卒業
- 2001年 九州芸術工科大学大学院芸術工学研究科博士前期課程修了 (情報伝達専攻)
- 2004年 九州芸術工科大学大学院芸術工学研究科博士後期課程修了 (芸術工学専攻), 博士 (芸術工学)
- 2005年 清水建設株式会社技術研究所勤務
現在に至る

大久保 寛 (おおくぼ かん)

- 1999年 東北大学工学部電気工学科卒業
- 2001年 東北大学大学院工学研究科博士前期課程修了 (電気・通信工学専攻)
- 2001年 秋田県立大学助手・助教
- 2004年 東北大学大学院工学研究科博士後期課程修了 (電子工学専攻), 博士 (工学)
- 2007年 首都大学東京助教
- 2009年 首都大学東京准教授
現在に至る

鈴木 久晴 (すずき ひさはる)

- 2002年 九州芸術工科大学芸術工学部音響設計学科卒業
- 2004年 九州芸術工科大学大学院 芸術工学研究科 博士前期課程修了 (芸術工学専攻)
- 2007年 九州大学大学院 芸術工学府 博士後期課程修了 (芸術工学専攻), 博士 (芸術工学)
- 2007年 九州大学COE学術研究員
- 2008年 日本エヴィクサー株式会社研究開発部部长
現在に至る

星 和磨 (ほし かずま)

- 1999年 日本大学理工学部建築学科卒業
- 2001年 日本大学大学院理工学研究科博士前期課程修了 (建築学専攻)
- 2001年 株式会社アセント勤務
- 2002年 日本大学短期大学部建設学科副手
- 2008年 日本大学大学院理工学研究科博士後期課程修了 (建築学専攻), 博士 (工学)
- 2008年 日本大学短期大学部建設学科助手
現在に至る

はじめての音響数値シミュレーション プログラミングガイド
Beginners' programming guide to numerical acoustic simulation

© 一般社団法人 日本建築学会 2012

2012年11月30日 初版第1刷発行

検印省略

編 者 一般社団法人 日本建築学会
発 行 者 株式会社 コロナ社
代 表 者 牛来真也
印 刷 所 三美印刷株式会社

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社
CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-00838-8 (新宅) (製本：愛千製本所)

Printed in Japan



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられております。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めておりません。

落丁・乱丁本はお取替えいたします