音響解析ソフトウェア Advance/FrontNoiseの時間領域解析機能およびその並列処理性能

松原　聖*　　末光　啓二**

Time domain acoustic analysis by Advance/FrontNoise

Kiyoshi Matsubara* and Keiji Suemitsu**

音響解析ソフトウェア Advance/FrontNoise は、これまで周波数領域での解析機能（以下、周波数ソルバ）のみについて開発を実施し、ユーザーに提供してきた[1]。しかし、時間領域での音響解析機能（以下、時間領域ソルバ）についてもいくつかのニーズが出てきた。そこで、当社では新しいアルゴリズムによる時間領域ソルバーを開発し、その第1バージョンをリリースしたのでここに報告する。本稿では、時間領域ソルバーの計算原理とともに処理時間・並列計算等の性能について述べる。特に、並列化においては高い並列化効率を得たことを報告する。

Key word:音響解析、時間領域、大規模解析、並列化、MLSM、FDTD、有限要素法、形状関数

1. はじめに

音響解析ソフトウェア Advance/FrontNoise では、周波数領域での解析機能（周波数領域ソルバ）のみを提供してきた。しかし、音源から発生する音の波面を見たい等の理由により、時間領域での解析機能（時間領域ソルバ）でのニーズが出てきた。これまでに利用してきた周波数ソルバーでは、周波数毎に方程式を陰的に解いていくために、特に大規模問題では非常に大きな処理時間と大きな記憶容量を必要としていた。また、場合によっては収束しにくい問題もあり、そのような場合には結果が得られないこともあった。このような状況から時間領域に期待されるニーズに対しては、周波数ソルバーでの陰的な処理はオーバースペックとなっていた。

これに対して、時間領域の問題に対しては陽的な手法を利用した解法が適用可能であるため、そのような手法を利用した時間領域ソルバーでは時間進行に伴い何らかの結果が得られるという利点がある。さらに、万が一計算が発散した場合にも、タイムステップを細かくして解析することで多少処理時間がかかるものの何らかの結果を得ることができることが保証されている。このような時間領域ソルバーの利用方法により、音の波面を見るということが可能である。これが時間領域ソルバーの利点である。ただし、ひとつ注意が必要なこととして、ある周波数の音の伝播について同じ精度を得るために必要なメッシュのサイズは周波数領域と時間領域のソルバーでは同一であることを挙げておく。

また、原理的に、時間領域のソルバーを用いて得られた準定常状態に達した音場から、周波数領域のソルバーと同一の結果を得ることが可能である。しかし、時間領域のソルバーを用いて準定常状態になるまでには、非常に多くのタイムステップを計算する必要があり、また、大きな処理時間が必要である。特に、共鳴を含む場合にはその計算は困難となる。従って、時間領域のソルバーを用いて、準定常状態を解析することは現実的には不可能であると考えていい。また、その定常状態になったという判定は困難な場合が多い。

*アドバンスソフト株式会社　代表取締役社長
President, AdvanceSoft Corporation
**アドバンスソフト株式会社　第1事業部
1st Computational Science and Engineering Group, AdvanceSoft Corporation

以上のことから、共鳴等を含む定常的な音をシミュレーションする目的には周波数領域のソルバーが適しており、音源からの音の伝播の波面等をシミュレーションする目的には時間領域のソルバーが適している。すなわち、この 2 つのソルバーは必要に応じて使い分けが必要で、その使い分けで音響解析ソフトウェア Advance/FrontNoise の用途が拡がるものと考えている。

2. 時間領域解析機能

時間領域音響ソルバーである音響解析ソフトウェア Advance/FrontNoise/ TD は、有限要素法を用いた時間領域ソルバーである。本ソフトウェアでは、四面体 1 次要素から構成されるメッシュで与えられた解析領域に対して、音源の位置とその時系列データを入力として与え、メッシュの節点における時間領域における解析領域内の音響伝播（音圧、粒子速度、音圧レベル）を計算することが可能である。ある時刻での音圧等を解析領域全体で出力すること、および、あらかじめ与えた観測点での音圧等の時系列変化を出力として得ることが可能である。

時間領域ソルバーAdvance/FrontNoise/TD では、時間領域の音響基礎方程式を解く。音源としては、メッシュの節点に対して点音源を与えることができる。また、メッシュの要素の面に対して、反射や無反射の条件を含むインピーンダンスの境界条件を与えることが可能である。

本ソフトウェアで利用する離散化においては、解析条件が柔軟に設定可能な有限要素法を適用している。ここでは、どんな複雑形状に対しても容易にメッシュを作成できるように、有限要素法においては、形状適合性の高い四面体 1 次要素を採用している。四面体要素を採用していることにより、ユーザーによるメッシュ作成の自由度が大幅に向上している。有限要素法で離散化したのちに、自動領域分割による並列化を行う。並列化は MPI を用いて行っている。解析結果は、粒子速度、音圧、音圧レベルを出力できる。また、解析結果については、いくつかの周辺ツールにより編集し、音響特性を求めることができる。

以上で述べたように Advance/FrontNoise/TD の特長は、①大規模解析が可能、②低コストのソフトウェア、③メンテナンス体制である。このうち、大規模解析では、1 億要素・2000 万節点程度の解析実績がある。また、Advance/FrontNoise /TD ではプリポストを合わせて提供することも可能であるし、同時に、騒音・音響解析の最小限度必要なコア部分のみを合理的な価格で提供している。また、複数 CPU での稼動については同一料金を設定している。さらに、自社開発ソフトのため十分なサポート体制がとれ、特定のニーズにカスタマイズが可能である。

3. ソフトウェア概要

3.1. 機能一覧

音響解析ソフトウェア Advance/FrontNoise/ TD の機能一覧を下表に示す。

表 1　Advance/FrontNoise/TD の標準機能

項目		機能
基礎方程式		音圧と粒子速度の波の方程式を時系列で解く
解析領域		内部領域、および外部領域
物性値等		場で一定とする
境界条件	点音源	節点に時系列でエネルギーを設定可能
	インピーダンス	壁境界に音響インピーダンスを設定可能
	透過境界	面（外部境界）に ρc 境界を設定可能
	拡散反射境界	拡散反射境界部分に拡散反射条件を設定可能
数値解法	離散化	有限要素法
	要素	四面体 1 次要素
	時間積分	陽解法
	並列計算	自動領域分割で MPI で並列化
解析結果		音圧等を指定された出力間隔でファイル出力 定点の音圧等の時系列データをファイル出力 音圧等をある区間で平均化して出力

本ソフトウェアでは、外部領域・内部領域にかかわらず、音圧と粒子速度の波の方程式を時系列で解く。音の伝搬する媒体の物性値は全領域で一定とする。境界条件としては、点音源、壁にインピーダンスを与える条件、開口部等を模擬する無反射境界、拡散反射境界である。数値解法としては有限要素法を陽解法で解く。また、自動領域分割で MPI による並列化を行っている。ユーザーにはコア数制限なしの並列計算機能を提供する。

ここで、時間領域ソルバーは第 1 バージョンであることから、周波数領域ソルバーは時間領域ソルバーよりも多機能となっている。時間領域ソルバー不足している機能については、原理的に時間領域ソルバーに導入できる機能が多く、今後のユーザーのニーズにしたがって、不足する機能を開発する予定である。

3.2. ソフトウェア構成

Advance/FrontNoise/TD と関連するファイルの関係を下図に示す。

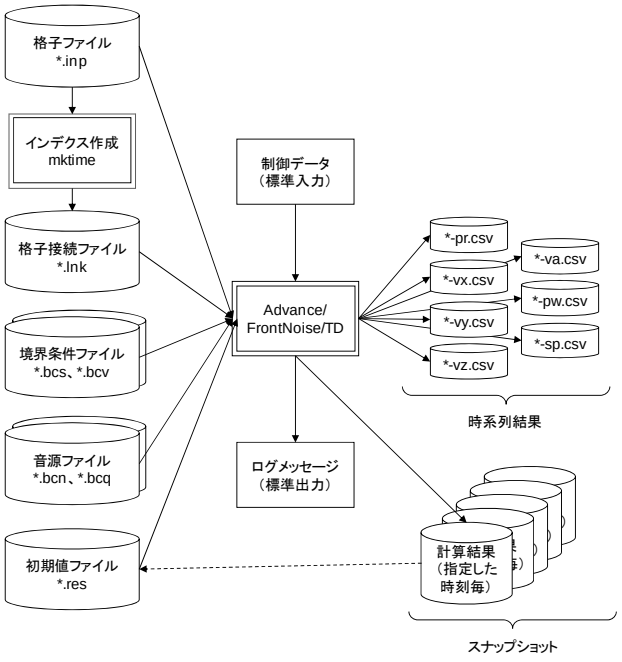


図 1　Advance/FrontNoise/TD
ソフトウェア構成

制御データファイルでは、計算モデル名、媒体の音速、密度、解析開始時間、解析終了時間、解析時間ステップ幅、出力間隔を指定する。各入力ファイルは計算モデル名を元に Advance/FrontNoise/TD のルールに従い自動的に決定される。

格子ファイルは、格子点座標、要素節点情報が UCD 形式で記述された格子ファイルである。境界条件ファイルでは、格子ファイルの情報を参照して、境界条件を与える面（群）を定義する。音源ファイルでは、節点に対して、音源の時系列変化を与える。

また、初期値を初期値ファイルとして与えることが可能である。初期値ファイルは計算結果ファイルと同じ形式であり、結果ファイルのファイル名を変更して、初期値ファイルとして利用することが可能である。

4. 時間領域における音響解析の定式化

4.1. 定式化の特長

本ソフトウェアの特長はその数値計算アルゴリズムにある。計算原理については、これまで差分法に適用されていた手法を有限要素法に適用した。これまで、そのような手法は論文レベルで存在してはいたものの、処理時間が遅い等の理由で広く利用されていなかった。今回、いくつかの工夫を行い、有限要素法に適用する手法を開発して、時間領域での音響シミュレーションを有限要素法で行うことを可能にした。また、有限要素法については、従来から数値計算のソフトウェアに適用されてきた並列化で高速化する手法を適用することで、かなり高い並列性能を得た。

以上のことから、本ソフトウェアにおいては、有限要素法を利用することで形状適合性の高いメッシュを適用できるとともに、差分法で利用されている手法を有限要素法に適用することで高速に処理を行うことを可能とした。

すなわち、従来から利用されてきた差分法ソルバーと比較すると、有限要素法を利用した本ソフトウェアは

①形状適合性が非常に高い。
②処理速度は同等の性能である

という 2 つの特徴を持つ手法であるといえる。

4.2. 基礎方程式

基礎方程式は次の通りである。

$$\rho\frac{du}{dt}+grad(P)=0 \qquad (1)$$

$$\frac{dP}{dt}+\kappa\cdot div(u)=0 \qquad (2)$$

ここで、ρ は音の伝播媒体の密度であり、κ は体積弾性率である。体積弾性率は、音速 c を利用して

$$\kappa=\rho c^2 \qquad (3)$$

と表すことができる。また、u はベクトルであるため、成分毎に記述すると

$$\rho\frac{du_x}{dt}+\frac{\partial P}{\partial x}=0 \qquad (4)$$

$$\rho\frac{du_y}{dt}+\frac{\partial P}{\partial y}=0 \qquad (5)$$

$$\rho\frac{du_z}{dt}+\frac{\partial P}{\partial z}=0 \qquad (6)$$

$$\frac{dP}{dt}+\kappa\left(\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right)=0 \qquad (7)$$

となる。空気では、室温では、

$$\rho=1.24\left[kg/m^3\right] \qquad (8)$$

$$c=3.43\times10^2\left[m/\sec\right] \qquad (9)$$

$$\kappa=1.41\times10^5\left[kg/m\sec^2\right] \qquad (10)$$

程度の値となる。

4.3. 境界条件

境界条件については、

- インピーダンスを与える
- 完全反射
- 無反射境界（吸収境界）

の 3 通りとする。

物理的に与えられるインピーダンス Z は、

$$Z=\frac{P}{V} \qquad (11)$$

である。壁に対して normal 方向の成分が有効であるため、n を壁での法線ベクトルとして、

$$Z=\frac{P(x,t)}{n\cdot V(x,t)} \qquad (12)$$

が成り立つ。この関係式を満たすように、決定する。完全反射の場合には、インピーダンスは無限大であるため、

$$n\cdot V(x,t)=0 \qquad (13)$$

と設定する。また、無反射境界については、周波数領域の解析と同じく

$$Z=\rho c \qquad (14)$$

と与えることではうまく表現できないため、通常の FDTD 法で利用されている方法を採用した。

4.4. 離散化の手法

離散化の手法では、FDTD 法の考え方を利用する。FDTD 法は、1966 年に K.Yee が考案したことから開始された差分法の手法である。この手法では、速度を圧力の定義点をスタッガードに配置し、さらに、時間方向にも速度と圧力をスタッガードに配置する。この配置により、クーラン条件を満たす程度のタイムステップに対して、電磁界の解析を陽解法で安定的に解くことができるようになった。その後、FDTD 法は弾性体の振動や波動の解析をはじめとして音響解析にも利用されるようになった（[2][3]等）。しかし、差分法では、形状への適合性に非常に厳しい条件があり、その適用範囲は限定せざるを得なかった。

本ソフトウェアでは、差分法と同様に大規模な計算を行うこと、および、形状への適合性を柔軟なものにも適用可能なソフトウェアとすることを目的として開発を行ってきた。従って、差分法のような直交の構造格子ではなく、非構造格子を利用する。また、さらに、複雑な形状への適合を目指すため、六面体要素ではなく、四面体要素での有限要素法を利用する。FDTD 法に沿った離散化の仮定では、有限体積法よりも自然な形で微分が離散化できるため、有限要素法を適用した。

しかし、FDTD 法は、差分法を対象として考案されたアルゴリズムであり、有限要素法の定式化に持っていくためには、いくつかの工夫を行う必要がある。四面体 1 次要素は、最も差分法の概念

からは離れており、この要素で離散化および検証を行うことができれば、FDTD 法を有限要素法に適用する手法の正しさが検証できる。従って、本ソフトウェアでは、基本的に四面体 1 次要素を対象とする。ここで行った手法上の特長は下記の通りである。

① 四面体の節点で速度を定義し、四面体の重心で圧力を定義する。

② （四面体の節点で定義された）速度の微分を圧力の定義点（四面体要素中心）で求めるために、有限要素法の形状関数を利用する。

③ （四面体の要素中心で定義された）圧力の微分を速度の定義点（四面体の節点）で求めるために、MLSM（Mean Least Square Method）を利用する。

なお、時間方向の物理量の定義に関しては、スタッガードに配置し、従来の FDTD 法が利用可能である。以下では、このアルゴリズムについて説明する。

4.5. 物理量の定義位置

本ソフトウェアでは、速度と圧力の定義点をスタッガードに配置し、さらに、時間方向にも速度と圧力をスタッガードに配置する。

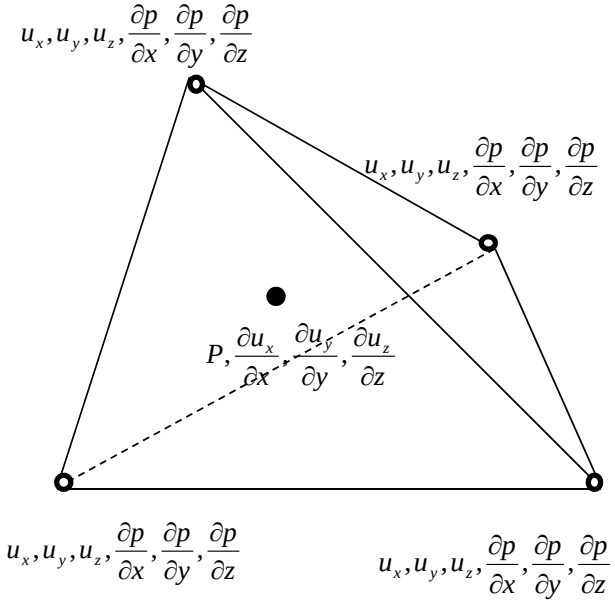


図 2　格子点における物理量の配置

4.6. 速度の微分の求め方

速度は 3 成分あるが、ここでは、X 方向の速度についてのみ述べる。これを代表して u と記述して式を展開する。まず、速度の微分については、通常の有限要素法と同じである。四面体 1 次要素を対象としているため、速度の微分は要素内で一定の値となる。

$$u(x,y,z)=\sum_{i=1,4}N_i(x,y,z)u_i \qquad (15)$$

であり、ここで求めるべき量は、

$$N_i(x,y,z)=a_ix+b_iy+c_iz+d_i \qquad (16)$$

の 16 個の数値（a_i,b_i,c_i,d_i の 4 変数に対して i=1,4）を求める必要がある。ここでこれらの形状関数の満たすべき式は、

$$N_i(x_j,y_j,z_j)=u_i\delta_{ij} \qquad (17)$$

の 16 個の関係式であり、これを解くことで、16 個の数値を決定できる。具体的には、ひと組の a_i,b_i,c_i,d_i を

$$\begin{pmatrix}x_1&y_1&z_1&1\\x_2&y_2&z_2&1\\x_3&y_3&z_3&1\\x_4&y_4&z_4&1\end{pmatrix}\begin{pmatrix}a_k\\b_k\\c_k\\d_k\end{pmatrix}=\begin{pmatrix}\delta_{1k}\\\delta_{2k}\\\delta_{3k}\\\delta_{4k}\end{pmatrix} \qquad (18)$$

として解くことができる。これで、a_i,b_i,c_i,d_i を求めることができた。具体的には、速度の微分を

$$\frac{\partial u_x}{\partial x}=\sum_{i=1,4}a_iu_i \qquad (19)$$

$$\frac{\partial u_y}{\partial y}=\sum_{i=1,4}b_iu_{y_i} \qquad (20)$$

$$\frac{\partial u_z}{\partial z}=\sum_{i=1,4}c_iu_{z_i} \qquad (21)$$

として求める。プログラム内では、$a_i,b_i,c_i(i=1,2,3,4)$ を動作の最初に計算しておき、それを保持しておく。

4.7. 圧力の微分の求め方

ここで考慮する領域内に着目した節点近傍にある点を利用して

$$P(x,y,z)=\sum_{i=1,n} M_i(x,y,z)P_i \qquad (22)$$

とできるような

$$M_i(x,y,z)=a_i x+b_i y+c_i z+d_i \qquad (23)$$

を見つける。

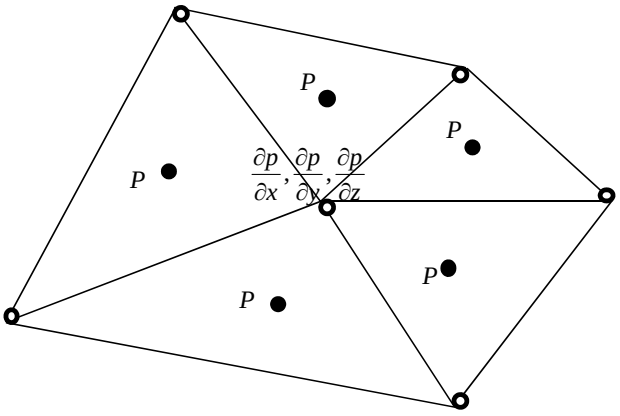


図 3　圧力の微分を求める手法

ここで、未知数の数は、$4n$ である。また、n は、近傍に存在する点の個数である。具体的には、着目した節点を持つ要素の集合であり、四面体要素で分割された物体の内部の点であれば、n は通常 10 数個の点となる。この関数に課せられる条件は、

$$M_i(x_j,y_j,z_j)=\delta_{ij} \text{、} i,j=1,\cdots,n \qquad (24)$$

である。この条件は、n^2 個となる。未知数の数と条件の式の数が異なるため、$n=4$ の場合にのみこの式を解くことができ、$n<4$ の場合には未知数は未定となるし、$n>4$ の場合には過剰条件となる。上記で述べたように、一般的には n は通常 10 数個の点であるため、過剰条件となる。従って、ここで、MLSM を利用する。すなわち、

$$F=\sum_{i=1,n}\sum_{j=1,n}\left\{M_i(x_j,y_j,z_j)-\delta_{ij}\right\}^2 \qquad (25)$$

が最小になるように、a_i,b_i,c_i,d_i を決定する。通常の手順に従い、次のように計算できる。まず、

$$F(a_i,b_i,c_i,d_i;i=1,\cdots,4)=\sum_{i=1,n}\sum_{j=1,n}\left\{a_i x_j+b_i y_j+c_i z_j+d_i-\delta_{ij}\right\}^2 \qquad (26)$$

ここで、i,j の添え字に注意する必要がある。この式が最小になるため、

$$\frac{\partial F}{\partial a_i}=\sum_{j=1,n} x_j\left\{a_i x_j+b_i y_j+c_i z_j+d_i-\delta_{ij}\right\}=0 \qquad (27)$$

$$\frac{\partial F}{\partial b_i}=\sum_{j=1,n} y_j\left\{a_i x_j+b_i y_j+c_i z_j+d_i-\delta_{ij}\right\}=0 \qquad (28)$$

$$\frac{\partial F}{\partial c_i}=\sum_{j=1,n} z_j\left\{a_i x_j+b_i y_j+c_i z_j+d_i-\delta_{ij}\right\}=0 \qquad (29)$$

$$\frac{\partial F}{\partial d_i}=\sum_{j=1,n}\left\{a_i x_j+b_i y_j+c_i z_j+d_i-\delta_{ij}\right\}=0 \qquad (30)$$

である。すなわち、

$$a_i\sum_{j=1,n} x_j^{\,2}+b_i\sum_{j=1,n} x_j y_j+c_i\sum_{j=1,n} x_j z_j+d_i\sum_{j=1,n} x_j=x_i \qquad (31)$$

$$a_i\sum_{j=1,n} x_j y_j+b_i\sum_{j=1,n} y_j^{\,2}+c_i\sum_{j=1,n} y_j z_j+d_i\sum_{j=1,n} y_j=y_i \qquad (32)$$

$$a_i\sum_{j=1,n} x_j z_j+b_i\sum_{j=1,n} y_j z_j+c_i\sum_{j=1,n} z_j^{\,2}+nd_i=z_i \qquad (33)$$

$$a_i\sum_{j=1,n} x_j+b_i\sum_{j=1,n} y_j+c_i\sum_{j=1,n} z_j+d_i\sum_{j=1,n} 1=1 \qquad (34)$$

を満たす。従って

$$\begin{pmatrix} \sum x_j^{\,2} & \sum x_j y_j & \sum x_j z_j & \sum x_j \\ \sum x_j y_j & \sum y_j^{\,2} & \sum z_j x_j & \sum y_j \\ \sum x_j z_j & \sum z_j x_j & \sum z_j^{\,2} & \sum z_j \\ \sum x_j & \sum y_j & \sum z_j & n \end{pmatrix}\begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix}=\begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

、$i,j=1,\cdots,4$ (35)

を節点毎に解くことになる。ここまでで、

a_i, b_i, c_i, d_i を求めることができた。最終的に求める微分量は、

$$\frac{\partial P}{\partial x}(x,y,z)=\sum_{i=1,n}\frac{\partial M_i}{\partial x}(x,y,z)P_i \qquad (36)$$

$$\frac{\partial P}{\partial y}(x,y,z)=\sum_{i=1,n}\frac{\partial M_i}{\partial y}(x,y,z)P_i \qquad (37)$$

$$\frac{\partial P}{\partial z}(x,y,z)=\sum_{i=1,n}\frac{\partial M_i}{\partial z}(x,y,z)P_i \qquad (38)$$

であり、

$$\frac{\partial M_i}{\partial x}(x,y,z)=a_i \qquad (39)$$

$$\frac{\partial M_i}{\partial y}(x,y,z)=b_i \qquad (40)$$

$$\frac{\partial M_i}{\partial z}(x,y,z)=c_i \qquad (41)$$

であるから、

$$\frac{\partial P}{\partial x}=\sum_{i=1,n}a_iP_i \qquad (42)$$

$$\frac{\partial P}{\partial y}=\sum_{i=1,n}b_iP_i \qquad (43)$$

$$\frac{\partial P}{\partial z}=\sum_{i=1,n}c_iP_i \qquad (44)$$

として求める。プログラムでは、、
$a_i, b_i, c_i (i=1,\cdots,n)$ を動作の最初に計算しておき、それを保持しておく。

プログラムに関する注意を 2 つほど述べる。ここで求める量は、d_i を除き、平行移動に関して不変な量である。従って、プログラムで計算する場合には、精度の面から、最終的に求める節点を原点として計算すべきである。また、

$$\frac{\partial P}{\partial x}=\sum_{i=1,n}a_iP_i \qquad (45)$$

において、$P(x)=x$ とすると、

$$\frac{\partial P}{\partial x}=1 \qquad (46)$$

である事実を利用して、

$$\sum_{i=1,n}a_ix_i=1 \qquad (47)$$

であるかどうかを確認して、プログラムの部品をチェックすることができる。同様に、

$$\sum_{i=1,n}b_iy_i=1、\ \sum_{i=1,n}c_iz_i=1 \qquad (48)$$

である。

5. 使用例

5.1. 作業フロー

Advance/FrontNoise を利用した音響解析のフローについて下記に示す。解析領域の形状の定義から開始し、その領域に関して四面体 1 次の格子を作成する。形状を作成する部分は市販またはフリーの CAD ソフトを利用する。また、メッシュ作成には、Advance/REVOCAP for FrontNoise を利用するか、または、形状作成と同様に市販のソフトまたはフリーソフトを利用する。

メッシュを作成後、音響解析のデータ準備に移る。格子ファイルおよび境界条件ファイルはここまでの操作で完成したので、主として音源および制御データを作成する。音源のデータについては、本ソフトウェアでは節点に音源を与える。座標を与えるとその位置にある節点を見つけるコマンドを用意（コマンド nfind）しており、この情報を利用して節点を指定する。また、音波のデータについては、その節点に対してテーブルで格納されたデータを参照するようにする。制御データについては、数行の内容を記述するのみである。

ここまでの準備のもと、コマンド「mktime」で時間領域の音響解析を実施するためのインデクス等を作成する。これは、解析を実行するにあたり必ず必要であるため、忘れないようにする。ただし、格子ファイルの変更がない限り、このインデクスは変更の必要がないため、解析実行前に必ず本コマンドを実行するのではなく、格子変更時にのみ忘れないようにする。

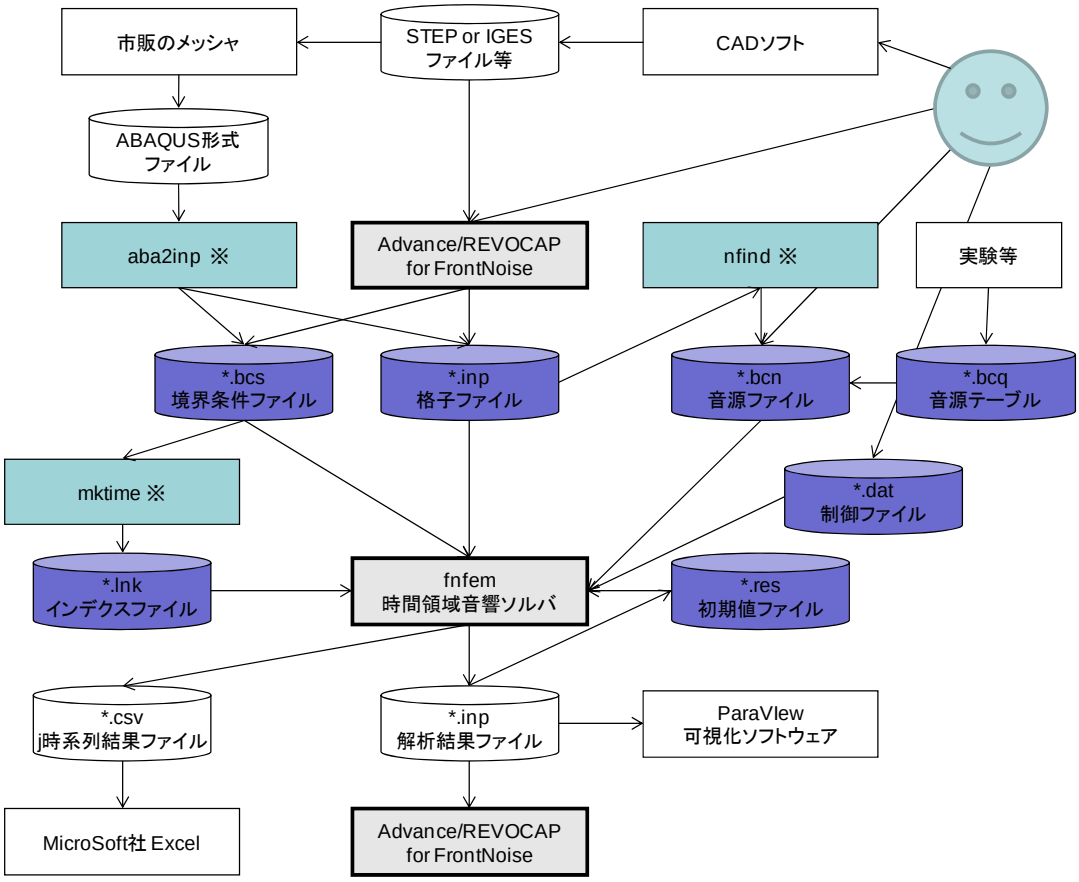


図 4　Advance/FrontNoise を利用した音響解析の作業フロー

5.2. 入力データの作成

5.2.1. 形状作成

市販の CAD ソフト等を利用して、3 次元形状の CAD データを作成する。作成したのちに、step 形式、iges 形式等でファイルに出力する。すでに、CAD データがある場合にはこの作業手順をスキップする。ただし、すべての CAD からすぐに格子を作成できるわけではない。必要に応じて CAD データを調整する必要がある。

本節の使用例では、工場の内部騒音の時系列解析を行うことを例にとって、本ソフトウェアの使用方法について説明する。下記の領域の大きさは、40m×30m×60m の大きさであり、工場等の建屋を想定していくつかの形状を配置していく。これらは仮想的な形状であり、斜め形状および曲線部のある形状を取り扱うために作成したデータである。有限要素法による特徴のあるメッシュを作成するためにこのような例とした。

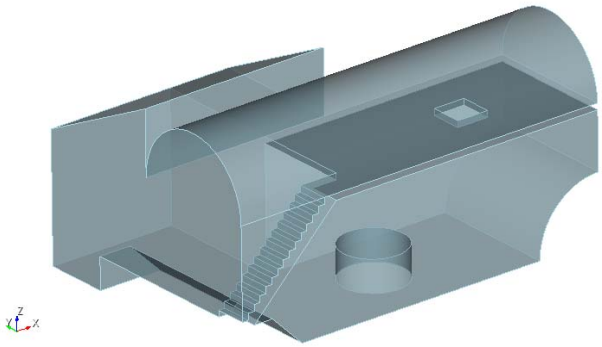


図 5　使用例における形状データ

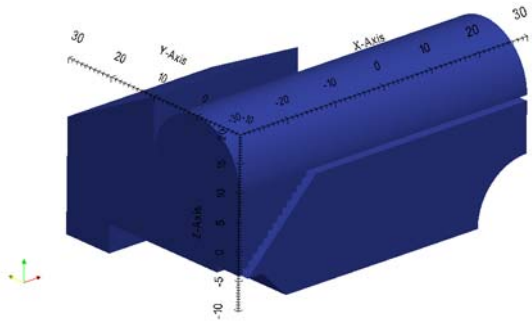


図 6　解析形状の大きさ

5.2.2. 格子の作成（必須）

(1) REVOCAP を利用する場合

CAD データを読み込み、四面体 1 次メッシュを作成する。下記はメッシュ作成の画面である。詳細の利用方法は、Advance/REVOCAP のドキュメントを参照のこと。

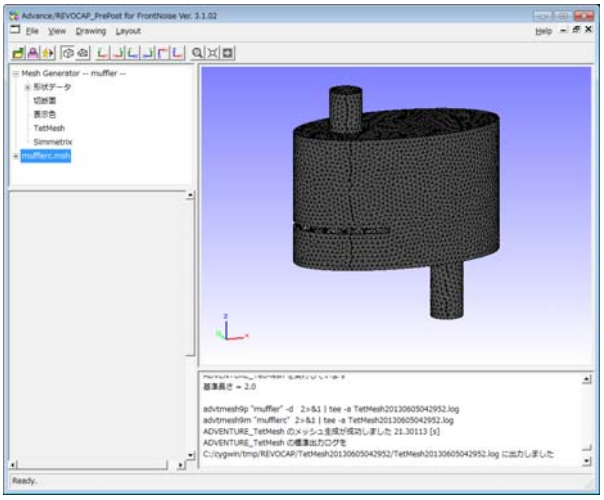


図 7　Advance/REVOCAP の画面

(2) 市販のメッシャを利用する場合

Advance/REVOCAP と同様に、CAD データを読み込み、四面体 1 次メッシュを作成する。つぎに、ABAQUS 形式でメッシュをファイル出力する。最後にツール aba2inp で Advance/FrontNoise 用のファイルに変換する。この場合には、*.inp ファイルおよび*.bcs ファイルのみが必要である。ここでは、「nfind」コマンドで得た音源から 2m おきの節点で出力した。その波形を示す。それぞれ時間遅れで波が到達していることが分かる。

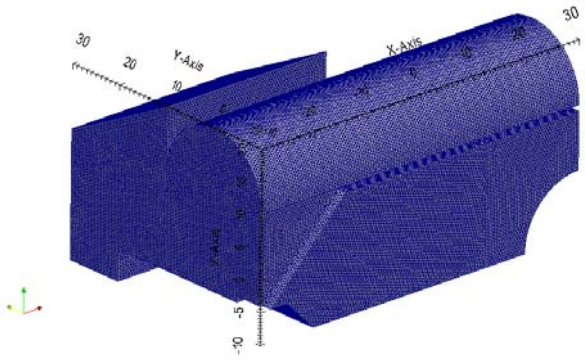


図 8　メッシュ図

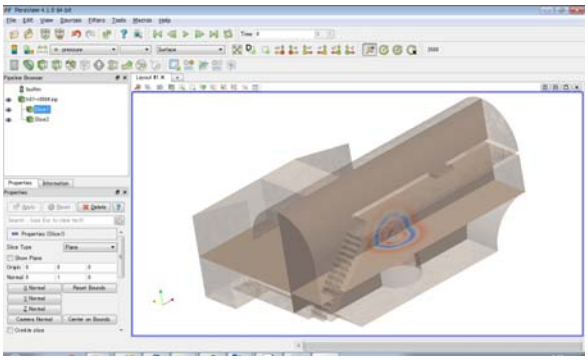


図 9　出力画面（ParaView を利用）

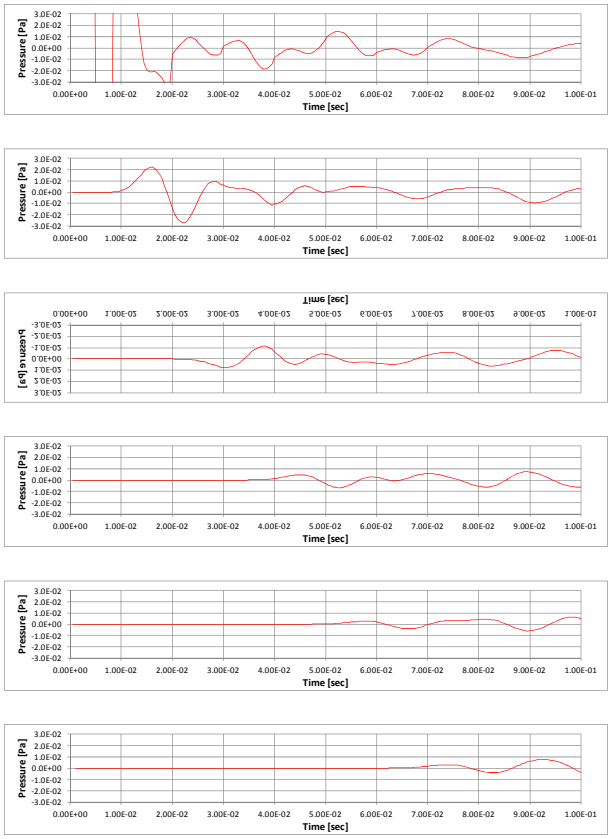


図 10　観測点での時系列グラフ

5.3. タイムステップの制限について

5.3.1. 解析の安定性

タイムステップの目安としては、この例では、格子幅 1m の場合には Δt= 1.0e-3[sec]程度で安定な解析が可能である。格子幅が 1/2 になればタイムステップも 1/2 にしなければならない。これは、クーラン数からくる発散しないための制約事項であり、解析精度からの制約条件ではない。

5.3.2. 解析の精度

また、解析したい波長からも格子幅は制限を受ける。十分な精度を得るためには、1 つの波の中

に 8 程度の格子点が必要である。

5.4. データサイズと処理時間・使用メモリ量

5.4.1. 実測値

ここでは、6 節の使用例で利用したデータ（ケース h01）を用いた。このデータを 1 回 refine したデータ（ケース h02）、および、2 回 refine したデータ（ケース h03）を利用した。いずれも計算サーバーでの実行であり、そのデータ準備の時間は含まれていない。

表 2　利用したデータ（その 1）

	節点数	要素数	リンク数
h01	320, 470	1, 785, 277	7, 141, 088
h02	2, 474, 445	14, 282, 176	57, 128, 704
h03	19, 423, 998	114, 257, 408	457, 029, 632

表 3　利用したデータ（その 2）

データ	ファイルサイズ	タイムステップ
h01	114MB	0.500msec
h02	1.01GB	0.250msec
h03	8.08GB	0.125msec

表 4　各データの前処理時間
（コマンド mktime）

データ	処理時間
h01	1 分以内
h02	16 分程度
h03	110 分程度

※ ディスクへの書き込み時間に依存する。

表 5　各データの処理時間（コマンド fntime）

データ	ファイル入力	前処理	時間進行/20steps	使用メモリ
h01	4.97sec	1.55sec	2.4sec	300MB
h02	33.0sec	9.5sec	18.9sec	3GB
h03	263sec	73.4sec	148.0sec	27GB

5.4.2. 処理時間等の予測

本ソフトウェアは、解析の前処理コマンド mktime については、処理時間はデータサイズ n の n*log(n)に比例する。また、解析本体 fntime の処理時間は、n に比例する。また、使用メモリについては、いずれのコマンドもデータサイズ n に比例する。

表 6　データ規模によるファイルサイズ

項目	小規模	中規模	大規模	超大規模
節点数	20 万	200 万	2000 万	1 億
要素数	100 万	1000 万	1 億	5 億
データサイズ	0.1GB	1GB	10GB	100GB

表 7　データ規模による処理時間（その 1）

項目	小規模	中規模
必要なメモリサイズ	0.3GB	3GB
前処理時間	1 分	10 分
100 ステップの解析	10 秒	2 分
1000 ステップの解析	2 分	20 分
1 万ステップの解析	20 分	2 時間
10 万ステップの解析	2 時間	1 日
結果ファイルサイズ	0.2GB/step	2GB/step

表 8　データ規模による処理時間（その 2）

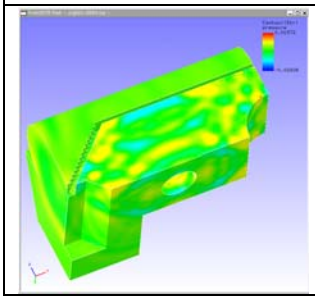
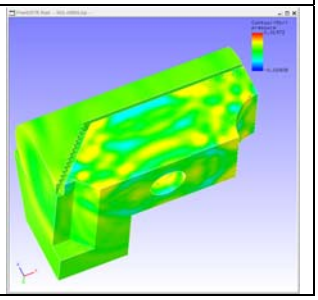
項目	大規模	超大規模
必要なメモリサイズ	27GB	270GB
前処理時間	2 時間	10 時間
100 ステップの解析	20 分	1 時間
1000 ステップの解析	2 時間	10 時間
1 万ステップの解析	1 日	5 日
10 万ステップの解析	10 日	50 日
結果ファイルサイズ	20GB/step	200GB/step

6. 並列機能の検証

6.1. 並列化結果

本ソフトウェアの並列化は、高速な処理が可能であることおよび汎用的な計算機で動作させるために、領域分割を利用して、MPI を用いて並列化を行った。前節で示したデータを用いて、本ソフトウェアの検証を実施した。逐次処理について、オリジナルソフトウェアとの解析結果の比較を以下に示す。数値的にも両者は完全に一致した。また、並列数による解析結果の比較を以下に示す。問題なく並列化できていることが確認できた。

表 9　並列化処理前後での比較

並列化前	並列化後
	

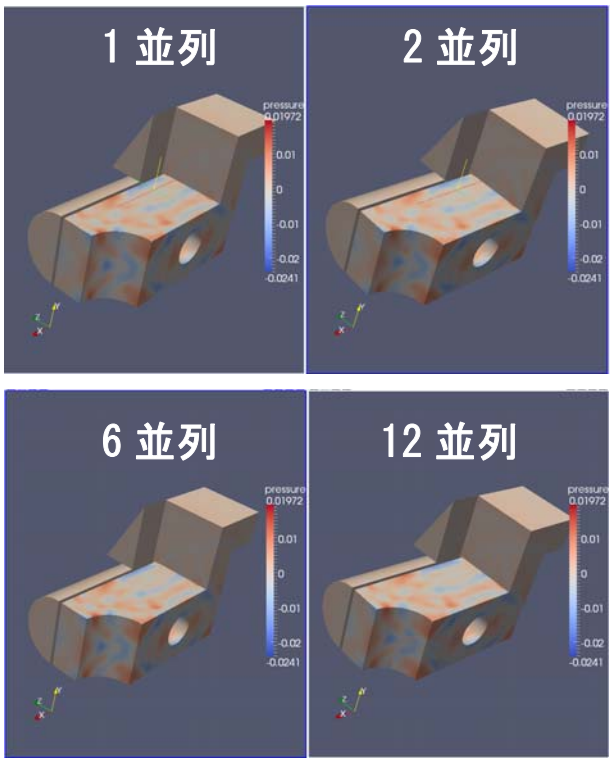


図 11　1 並列から 12 並列までの結果比較

6.2. 並列性能の計測

これらのデータで並列性能を計測した。時間積分の演算処理は良好な並列性能を示している。

表 10　処理時間

並列数	処理時間（秒）			
	全体	前処理	時間積分	出力
1	234.1	8.3	169.7	56.1
2	140.6	4.4	99.5	36.7
4	101.3	2.7	42.7	55.9
6	109.2	2.0	26.2	81.0
12	145.7	1.1	12.0	132.7

表 11　並列化率

並列数	増速率	
	前処理	時間積分
1	1.0	1.0
2	1.9	1.7
4	3.1	4.0
6	4.2	6.5
12	7.5	14.1

7. まとめ

本ソフトウェアは、従来から差分法に利用されてきた手法を有限要素法に適用し、有限要素法の形状適合性の利点も生かした新しい計算手法によるソルバーである。その手法と計算事例、および、その性能を紹介した。

当社の時間領域の音響ソルバーは、まだ緒に就いた段階であり、今後、必要な機能の開発を進めていく予定である。また、時間領域ソルバーと周波数領域ソルバーの使い分けについては、ユーザーのニーズをもとに検討しながら必要な機能追加を行い、今後ノウハウを蓄積していきたい。

参考文献

[1] 松原 聖, 桑原 匠史, “音響解析ソフトウェア Advance/FrontNoise の現状”, アドバンスシミュレーション Vol.15, 2013.5

[2] 村上桂一,青山剛史, “FDTD 法による音響透過損失の数値解析”, 第 40 回流体力学講演会／航空宇宙数値シミュレーション技術シンポジウム 2008 論文集

[3] 朝倉巧,坂本 慎一, “FDTD 法による音響振動連成解析を用いた遮音性能のシミュレーション”, 生産研究, Vol. 61, No.4, P.793-796 (2009)

※技術情報誌アドバンスシミュレーションは、アドバンスソフト株式会社 ホームページのシミュレーション図書館から、PDF ファイルがダウンロードできます。（ダウンロードしていただくには、アドバンス/シミュレーションフォーラム会員登録が必要です。）