JAERI-Data/Code
96-031

並列計算機による多原理統合型CFDシミュレーションの概念設計

1996年11月

大西亮一・太田高志・木村俊哉

日本原子力研究所
Japan Atomic Energy Research Institute

本レポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-11 茨城県那珂郡東海村）あて，お申し越しください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, 319-11, Japan.

© Japan Atomic Energy Research Institute, 1996

編集兼発行　日本原子力研究所
印　　刷　いばらき印刷㈱

JAERI-Data/Code 96-031

並列計算機による多原理統合型ＣＦＤシミュレーションの概念設計

日本原子力研究所計算科学技術推進センター
大西　亮一・太田　高志・木村　俊哉

(1996年10月１日受理)

航空機の統合シミュレーションを行うための空力弾性連成解析コードの設計を行った。並列計算機を用いてEuler/Navier-Stokes式を基礎式とした流体解析コードを、航空機主翼の有限要素モデルに連成させて解く手法を考案した。

近代航空機の開発では空力、構造、制御システム、運用性など多くの設計原理の最適化が重要であり、米国の航空宇宙産業を中心に、高性能計算機を使用した多原理最適設計の為の統合シミュレーションの研究が進められている。

本報告書では、流体解析と構造解析を動的変形格子を用いてMPMD(Multiple Program Multiple Data)方式により並列的に統合する方法を示している。流体解析では、構造物の弾性変形によりダイナミックに変化する流れ場の３次元解析を行う。これにより得られた圧力データを用いて構造解析を行い構造変形を求め、再度流体解析へ入力する。流体格子と構造要素の間で圧力と変位を相互に交換しながら、流体と構造の計算を繰り返す。これにより、空力と構造変形の相互作用を考慮したシミュレーションを行う事が出来る。

本報告書では、上記の処理を行うための各機能の仕様を報告している。

また、将来制御システムと統合し、空力弾性変形を伴う航空機の運動シミュレーションを行い、航空機統合シミュレーションのためのツールへと展開する案も示している。

日本原子力研究所：〒153 東京都目黒区中目黒町２－２－５４

JAERI-Data/Code 96-031

A Conceptual Design of Multidisciplinary-Integrated C.F.D. Simulation on Parallel Computers

Ryoichi OHNISHI, Takashi OHTA and Toshiya KIMURA

Center for Promotion of Computational Science and Engineering
Japan Atomic Energy Research Institute
Nakameguro, Meguro-ku, Tokyo

(Received October 1, 1996)

A design of a parallel aeroelastic code for aircraft integrated simulations is conducted. The method for integrating aerodynamics and structural dynamics software on parallel computers is devised by using the Euler/Navier-Stokes equations coupled with wing-box finite element structures.

A synthesis of modern aircraft requires the optimizations of aerodynamics, structures, controls, operabilities, or other design disciplines, and the R&D efforts to implement Multidisciplinary Design Optimization environments using high performance computers are made especially among the U.S. aerospace industries.

This report describes a Multiple Program Multiple Data(MPMD) parallelization of aerodynamics and structural dynamics codes with a dynamic deformation grid. A three-dimensional computation of a flowfield with dynamic deformation caused by a structural deformation is performed, and a pressure data calculated is used for a computation of the structural deformation which is imput again to a fluid dynamics code. This process is repeated exchanging the computed data of pressures and deformations between flowfield grids and structural elements. It enables to simulate the structure movements which take into account of the interaction of fluid and structure.

The conceptual design for achieving the aforementioned various functions is reported. Also the future extensions to incorporate control systems, which enable to simulate a realistic aircraft configuration to be a major tool for Aircraft Integrated Simulation, are investigated.

Keywords: CFD, Aerodynamics, Structural Dynamics, Coupled Simulation, Grid Generation, FDM, FEM, MDO, Design Optimization, Aircraft Design, Parallel Computer, MPMD, Parallel Processing, Super Computing

目　　次

1. 序　論 …… 1
1.1 背　景 …… 1
1.2 研究概要 …… 2
参考文献 …… 3
2. 流体格子と構造要素 …… 4
2.1 概　要 …… 4
2.2 形状データの生成 …… 5
2.3 流体格子の生成 …… 8
2.4 構造要素の生成 …… 10
2.5 格子と要素間のデータ変換 …… 11
2.6 流体格子の動的変形 …… 14
2.7 並列処理による格子と要素の生成 …… 15
参考文献 …… 16
3. 流体解析 …… 17
3.1 基礎方程式 …… 17
3.2 計算手法 …… 20
3.3 並列化手法 …… 22
3.4 ＭＰＭＤ並列化による移動格子における格子形成と流体計算の例 …… 25
参考文献 …… 29
4. 構造解析 …… 30
4.1 概　要 …… 31
4.2 計算手順 …… 31
4.3 係数マトリクスの作成 …… 32
4.4 直接時間積分による解法 …… 34
4.5 複合材の適用 …… 34
4.6 空力加熱による剛性低下の考慮 …… 35
4.7 並列処理の適用 …… 36
参考文献 …… 38
5. 連成解析システムの構成 …… 39
5.1 流体、構造の連成解析 …… 39
5.2 時間の同期 …… 40
5.3 並列化手法 …… 41
5.4 データ転送（インターフェース） …… 43

5.5 結果の可視化 …… 44
参考文献 …… 45
6. 制御システムとの統合概念 …… 46
参考文献 …… 49
7. 今後の進め方 …… 50
謝 辞 …… 50
APPENDIX A …… 51
APPENDIX B …… 53

Contents

1. Introduction 1
1.1 Background 1
1.2 Overview of the Research 2
References 3
2. Fluid Grid and Structural Element 4
2.1 Overview 4
2.2 Geometric Data Generation 5
2.3 Fluid Grid Generation 8
2.4 Structural Element Generation 10
2.5 Data Exchange between Grid and Element 11
2.6 Dynamic Deformation of Fluid Grid 14
2.7 Parallelization of Grid and Element Generation 15
References 16
3. Fluid Dynamics Analysis 17
3.1 Basic Equations 17
3.2 Numerical Methods 20
3.3 Parallelization 22
3.4 Example of MPMD Parallel Calculation in a Moving Grid 25
References 29
4. Structural Dynamics Analysis 30
4.1 Overview 31
4.2 Process of a Computation 31
4.3 Coefficient Matrix Generation 32
4.4 Solution of Direct Integration 34
4.5 Application of Composite Materials 34
4.6 Stiffness Decrease by Aerodynamic Heating 35
4.7 Application of Parallel Processing 36
References 38
5. Design of the Coupled Simulation System 39
5.1 Coupled Analysis of Fluid-structural Dynamics 39
5.2 Synchronization of Time 40
5.3 Parallelization Method 41
5.4 Data Transfer of Coupled System 43

5.5 Visualization 44
References 45
6. Concept on an Integration with Control System 46
References 49
7. Developmental Works 50
Acknowledgements 50
Appendix A 51
Appendix B 53

1. 序　　論

1.1 背景

計算機性能の高度化に伴い、大規模計算を統合し複数力学系の相互作用を考慮した、より完全なシミュレーションを行う研究が進められる様になった。特に、航空分野では空力弾性などでこの種の問題を扱う機会が多く、流体解析と構造解析を直接連成させた解析コードなどの開発が活発になされ、近年は、並列計算機を用いて制御や性能計算などと、より多くの統合を行い設計の同時最適化を図る研究、多原理設計最適化 (Multidisciplinary Design Optimization) の研究が進められている[1)2)]。統合シミュレーションは、機体の外部構造、内部構造、制御システム、推進など多数の要素の計算機モデルを作り、機体コンフィギュレーションや飛行条件などを変更した結果を、即時にシミュレートするものである。この様なシミュレーションは、同型機の開発例が少なく実験データの入手が困難な革新航空機の開発に重要な手段となる。

以上の背景から、一般産業分野にも応用が広い流体解析と構造解析の並列処理による連成を行い、統合シミュレーションの実現性について研究を行う。米国では 80 年代に NASA、空軍、航空機メーカーが共同で設計最適化の研究を行い、ASTROS(Automated STRuctural Optimization System) を始め、各種の空力弾性コードの研究開発を行っている[3)4)5)]。ASTROS は、空力と構造ソルバーに NASA の USSAERO と MSC/NASTRAN を組み合わせたもので、ダグラスやノースロップなどの機体メーカーはこの成果を基礎として最適設計ツールを開発し、超音速旅客機や短距離垂直離着陸 (STOVL) 機の概念設計に適用している。これらの研究は IBM3090 など従来の逐次処理計算機を用いて行われたが、90 年代には Intel Paragon や IBM SP2 などの並列計算機による空力弾性連成ソルバーや多原理最適設計の研究が行われ、次世代超音速旅客機の開発に適用する試みが NASA で行われる様になった。Computational Aero Science Project や HiSAIR(High Speed Airframe Integration Research) プロジェクトは、その様な分野の代表的なプロジェクトで、空力弾性解析、フラッタ計算、安定制御計算のコードを並列計算機あるいは Cray などの高速計算機上で統合し、超音速旅客機の空力／構造／制御の最適化を図る研究を進めている[6)]。また、各種のコード、形状データや格子データのデータベース、可視化ライブラリ等を複合計算機環境上に構築するためのフレームワークの研究も進められている。

日本でも次世代超音速旅客機開発計画で、航空宇宙技術研究所に於いて機体エンジン統合空力設計や逆問題 CFD 空力設計などの研究が進められ、ボルテックスフラップなど新式の高揚力装置の空力特性、先進制御システム、空力加熱対策、超音速航行エンジン、環境対策など各種の革新技術のシミュレーションを行う事が計画されている[7)]。

1.2 研究概要

航空機のシミュレーションには空力や構造以外に制御、推進、熱、高揚力装置や降着装置などの機器の作動といった要素が関連する。実際の運用性能も考慮すると、ソニックブームや騒音などの環境性能、軍用航空機でのレーダー断面積や脆弱面積の計算など、更に多くの要素が加わって来る。これらを全て統合的に模擬する事は極めて難しいが基本的な性能は空力、構造、制御によって見る事が出来、この三つの統合により流れ場の挙動、構造変形や応力状態、機体の総合的な運動性能を統一的に見る事が出来る。この様子を Fig.1.1 に示す。

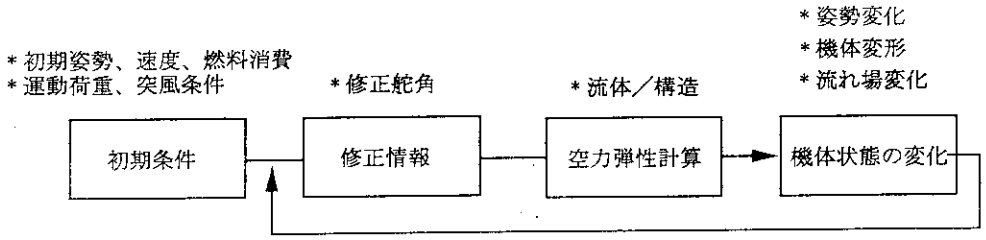


Fig.1.1 統合のイメージ

これは、航空機の位置、速度、姿勢、燃料消費などを初期値として与え、空力弾性計算を行いその結果として飛行状態の変化が得られ、それを修正情報として舵角を変更し、再び空力弾性計算を行うことにより機体の制御を行う流れを示している。研究開発の手順としては、制御系を含まない空力と構造だけの連成ソルバー開発を先に進める。理由は、航空機のシミュレーションでは空力と構造は最も基本となる部分であるため、及び流体力学と構造力学の連成は航空分野以外の一般分野にも広く応用が出来るため、である。また、一度に全機での解析を行わず主翼だけの解析を行った後、同じ手法を用いて胴体、尾翼を合わせた解析を行う。これは、準備する形状データや格子データ、計算処理時間などを軽くし、方式の正当性を早く確認するためである。

本研究の主な課題は次の通りである。

(1) 変形物体の流体解析の研究

構造変形を伴う 3 次元物体周りの流れ場計算について研究する。処理の高速化のために流体解析ソルバーの並列化についても研究する。

(2) 構造解析での並列処理の研究

動的に変化する荷重による有限要素構造解析において、高速化のための並列処理を研究する。

(3) 動的変形格子生成技術の研究

構造変形や可動する部分構造を持つ物体の流体計算を可能とするため、動的に変形する 3 次

元格子の生成方法について研究する。

(4) 連成計算のための並列統合の研究

ソルバー間でのデータのインタフェース、安定条件が異なる解析の間での解析時間の同期や負荷分散などについて研究する。

(5) 制御システム組み込みの研究

制御計算を組み入れた連成の可能性について検討する。構造変形や空力舵面の操作と制御則との連成を図り、姿勢角変化などを計算する方式を検討する。

なお、報告書には第二章で流体解析と構造解析の為の格子と要素の生成、構造変形に合わせた流体格子の動的変形、及び流体解析と構造解析との間での圧力データや変形量の受け渡しを、第三章及び第四章で並列処理による流体解析と構造解析を、第五章で連成計算の並列化の方針を、第六章及び第七章で制御システムとの統合概念と今後の開発の進め方を述べている。

参考文献

1) Sobieszczanski-Sobieski,J. and Chopra,I., "Multidisciplinary Optimization of Aeronautical Systems" ,J. of Aircraft,Vol. 27,NO. 12,1990,pp. 977-978

2) Sobieszczanski-Sobieski,J., "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design:Recent Advances and Results", J. of Aircraft,Vol. 27,NO.12,1990,pp. 993-1001

3) Neil,D.J.,Johnson,E.H., and Canfield,R., "ASTROS-A Multidisciplinary Automated Structural Design Tool", J. of Aircraft,Vol. 27,NO. 12,1990,pp.1021-1027

4) Dodd,A.J.,Kadrinka,K.E.,Loikkanen,M.J.,Rommel,B.A.,Sikes,G.D.,Strong,R.C.,and Tzong,T.J.,"Aeroelastic Design Optimization Program" ,J. of Aircraft,Vol. 27, NO. 12,1990, pp. 1028-1036

5) Miura,H. and Neil,D.J.,"Applications to Fixed-Wing Aircraft and Spacecraft/Structural Optimization: Status and Promise" ,Progress in Astronautics and Aeronautics, Vol. 150, AIAA,1992, pp. 705-742

6) Coen,P.G.,Sobieszczanski-Sobieski,J.,and Dollyhigh,S.M., "Preliminary Results from the High-Speed Airframe Integration Research Project", AIAA-92-1004,Irvine,California,February 1992.

7) 科学技術庁研究開発局 小型高速実験機飛行実験計画研究会,"小型実験機による次世代超音速機技術の研究の推進", 平成8年2月

2. 流体格子と構造要素

Nomenclature

A_e	要素面積
$d_1, ..., d_8$	節点と辺の垂直距離
D_1, D_n	始点と終点での格子点の間隔
f_1, f_2, f_3, f_4	Coon's スプラインのブレンディング関数
g_1, g_2, g_3, g_4	格子点と辺の垂直距離
$\boldsymbol{L_1, L_2, L_3, L_4}$	節点荷重ベクトル
L_1, L_2, L_3, L_4	節点荷重の大きさ
n	全格子数
NJ	J 方向の格子数
P	格子点圧力
S_1, S_2, S_3, S_4	要素の各辺
u_j,v_j,w_j	x,y,z 方向への格子点移動速度
$\boldsymbol{V_1, V_2, V_3, V_4}$	要素節点の座標

2.1 概要

流体と構造の連成解析を行うためには、流体格子と構造要素を同じ形状データから生成し両者のインタフェースをとらなければならない。本章では、流体格子と構造要素の元となる形状データの生成、格子と要素データの生成、格子と要素間のデータ交換、流れ場の動的変形について方式案を述べる。なお、1.2 で述べた様にここでは主翼だけを対象とした方式を示すが、胴体や尾翼もほぼ同じ方式が適用出来、最終的には第 6 章で述べる様に各構造のブロックをオーバラップさせ全機形態での解析を行う事を考慮している。

流体力学と構造力学の連成計算には第 5 章で述べる様に、両方の方程式を連立させて解く手法[1)2)]と、それぞれ独立した計算法により解き、互いの計算結果を交換して連成計算を進める手法[3)4)]がある。解析コードの変更や並列処理化が容易である事を考慮し後者の方式、即ち流体解析はナビアストークス式で、構造解析は非線形動解析式を有限要素法により解く。流体格子には構造格子を、構造要素には四辺形シェル要素を用いる。格子と要素は構造物表面 (ここでは主翼表面) で接し、空力荷重と構造変形量をソルバー間で相互に交換しながら空力弾性の連成計算を進めていく。Fig.2.1 に示す様、同一の形状データをもとに変形前の初期格子と要素を生成し流体計算を実行する。次に、流体計算から得た空力データを要素の節点荷重に変換し、構造計算を行い変形量を求める。この変形量から変形後の新たな構造要素のデータを作り出すと同時に、構造変形に合わせ流体格子を変形し、変形後の流体格子の座標とメトリックスを計算し、再度流体計算に入力する。この様な過程を繰り返す事により連成計算を行う。流体計算と構造計算はそれぞれ独立し

た実行が可能であるため、第5章に述べる様にMPMD(Multiple Program Multiple Data)方式による並列化を行う事を考慮している。

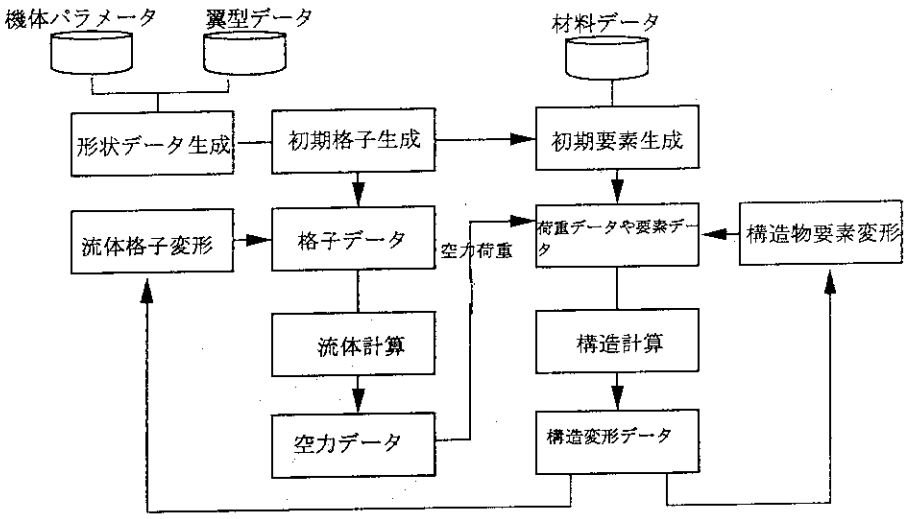


Fig.2.1 処理の流れ

2.2 形状データの生成

主翼の形状データは外板パネル、前後桁、桁間リブ、縦通材を多面体データで作ったもので、Fig.2.2の形状寸法をパラメータ化したデータ(Fig.2.3)から生成する。外形形状は100節点の基準翼型データに線形変換を行って生成される10-20の断面から作る(Fig.2.4)。内部構造は外形からの断面切り出しにより作る(Fig.2.5)。これらの処理は、全てプログラム内で自動的に行う事を考えている。翼型データは各種のものの使用を考慮し別ファイルとする。後述の様に、構造要素は形状データの多面体をそのまま用いるため、形状データと同時に有限要素データの生成も行う。Fig.2.3に600席クラスの遷音速旅客機の主翼を定義したパラメータの例を、これより生成された形状をFig.2.4に示す。以上の処理の概略手順を次に示す。

(1) パラメータの入力

構造パラメータと翼型データをファイルより読む。

(2) 外部構造の生成

外形寸法から平面形の生成、翼型の補間による中間断面の生成、平面形への中間断面の配置を行い主翼の3次元形状を生成する。基準翼型のスケーリング、ブレンド、回転により、中間断面の形状と捩じり角をスムージングする(Fig.2.4)。

(3) 内部構造の生成

前後桁、桁間リブ、縦通材を生成する。 桁、リブ、縦通材の位置に切断平面を置き、切断平面と外板の交差線を求め、これらの交差線から桁、リブ、縦通材の多面体を作る(Fig.2.5)。

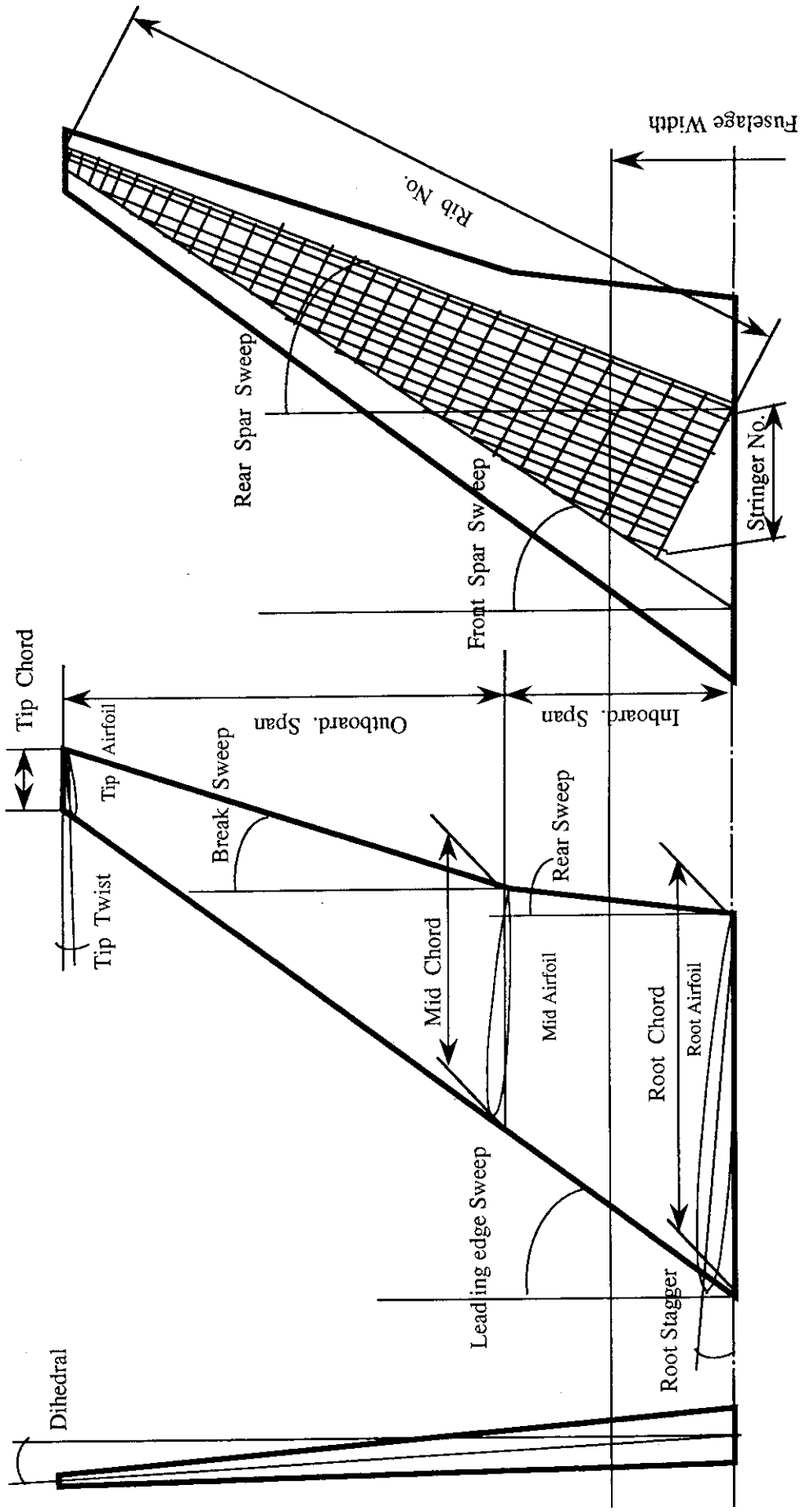


Fig.2.2 パラメータに定義される寸法

```
# WING DESIGN PARAMETERS.
# Canard Subsonic Ultra High Capacity Aircraft DR-1
#
# Aircraft type
# Section profiles at root, mean, and tip.
# Root, Mid, & tip chord.(ft)
# Whole , inboard,& outboard span.(ft)
# Fuselage width.(ft) , tip twist angle (degree)
# Root stagger, leading edge sweep, & dihedral angle.(degree)
#
uhca1
sc20714 sc20714 64a010
50.0 32.0 13.0
105.0 40.0 65.0
22.0 -2.0
2.0 35.0 6.0
```

Fig.2.3 主翼のパラメータ定義

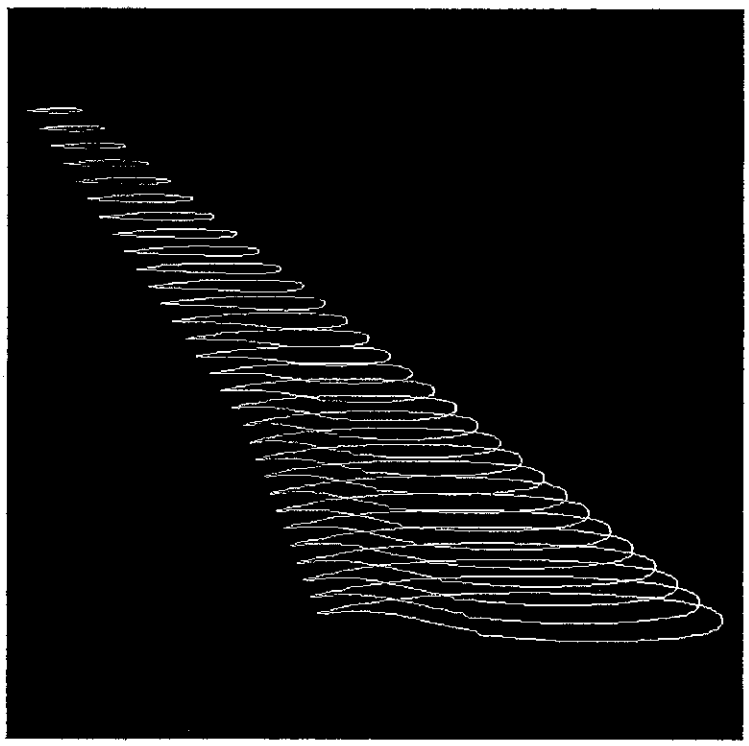


Fig.2.4 翼型断面の生成による外形の形成

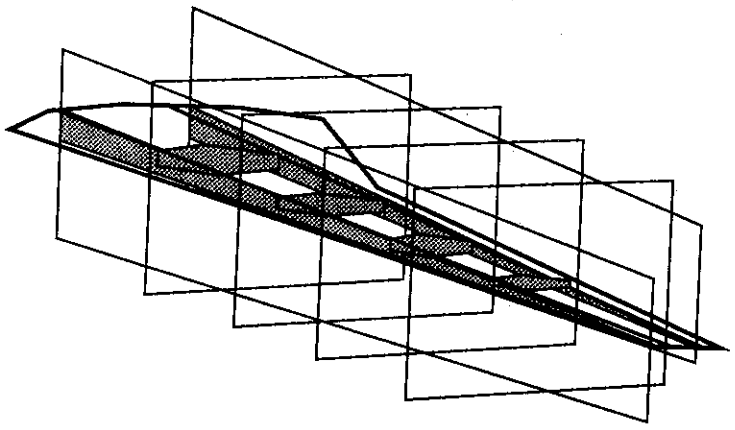


Fig.2.5 内部構造の生成

2.3 流体格子の生成

流体格子の生成では transfinite 法などの代数型格子により初期格子を生成し、楕円型偏微分式や放物線型偏微分式によりスムージングを行う手法が一般的であるが[6]、ここでは代数型格子にスプライン曲線を用いた格子生成を行う。本方式は偏微分型格子と較べて計算処理が簡略である事、後述の様に周囲格子点の情報を使わずに独立して格子点生成が行えるため、領域分割による並列化が容易である事である。特に、本研究で用いる流体格子は動的に変形されその都度スムージング処理が行われるため、差分式の収束計算を伴う偏微分型格子生成法では全体の処理速度を大きく落とす事が予想される。本方式では一回の代数計算でスムージングを行うため、処理速度の改善に大きく貢献出来ると思われる。

格子はコード断面でCタイプ、スパン断面でHタイプのトポロジーを持つ構造格子とし、主翼外板上に表面格子を発生させ、表面格子と法線ベクトル、及び外周境界上の格子点を用いてスプライン曲線を張り中間格子点を得る。この過程を次に示す。

(1) コード、フィールド、スパンの各方向に格子軸 (I,J,K) を Fig.2.6 の様に設定する。

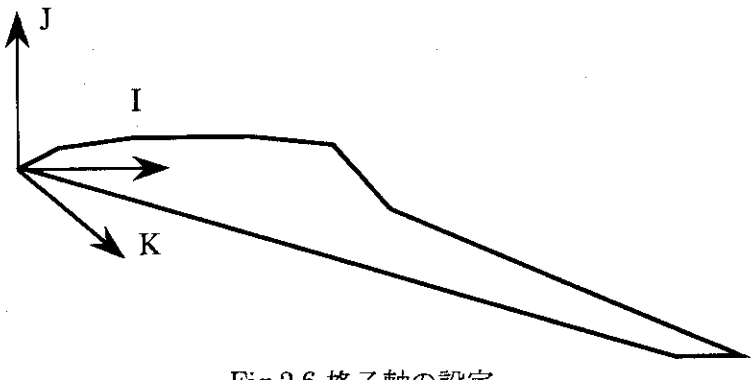


Fig.2.6 格子軸の設定

(2) K 軸方向の分割を行いコード断面を切り出す。分割位置は翼端部で格子集中が起きる様に決める。 コード断面の切り出しは内部構造と同様に、切断面定義により行う。

(3) コード断面上に表面格子を生成する。前後縁部に格子集中が起きる様に制御する。

(4) 表面格子の法線を求め、法線方向の接線ベクトルを持つスプライン曲線を生成し[5)]、曲線上に中間格子点を生成する。この時 (5) に示す方法で、表面付近に格子集中が起きる様に格子生成の制御を行う。

(5) 格子集中の制御

翼周りの流れには前縁部の高圧領域、後縁や翼端部の渦流発生域、表面の境界層付近など圧力変化が大きな領域が存在するため、この部分の格子密度を上げて精密な解析を行う事が一般になされる。ここではコード、フィールド、スパンの各方向に分割を行う際、スプライン補間によって格子分割を行う方法を示す。Fig.2.7 はコード方向での分割の概念を示している。図中破線で示される、主翼表面と後流域での格子点の x 座標値を式 (2.1) により求める。式 (2.1) は分割開始点 X_1 と終点 X_n、始点と終点でのギャップ D_1、D_nを用いてギャップ分布がスムーズになる様な中間分割を与えるスプライン補間式である[5)]。格子の集中度は D_1 と D_nで決まる。即ち、D_1が大なれば進行方向に向かって密に、小さければ粗に中間格子点の位置が計算される。D_1と D_nの値は補間区間の長さに比例する量として式 (2.1) に示す様計算する。係数 d_1 と d_nは、実際の格子生成状況を見ながら適当な値を設定する。フィールド及びスパン方向に対しても、同様の手順で中間格子点を求める。

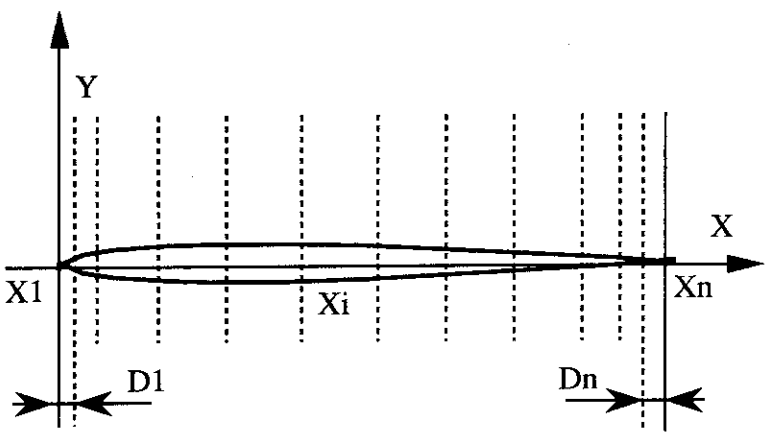


Fig.2.7 格子集中の制御

$$
\begin{aligned}
X_i &= f_1X_1 + f_2X_n + f_3D_1 + f_4D_n \\
f_1 &= 2t^3 - 3t^2 + 1 \\
f_2 &= -2t^3 + 3t^2 \\
f_3 &= t^3 - 2t^2 + t \\
f_4 &= t^3 - t^2 \\
D_1 &= d_1(X_n - X_1) \\
D_n &= d_n(X_n - X_1) \\
t &= \frac{i-1}{n}, \ (i = 1, 2, ..., n+1)
\end{aligned} \tag{2.1}
$$

ここで、nは補間区間内の全格子点数である。

2.4 構造要素の生成

有限要素法による構造解析を行うため縦通材を梁要素で、その他の部分を20自由度四辺形シェル要素で構成される要素データを生成する。 要素の構成は形状データの四辺形多面体の面構成と同じとする。このため、2.2でその作成方法を説明した形状データをもとに、Table1に示す要素データを生成する。要素データは材料特性、節点拘束条件、荷重条件、要素構成、節点座標をTable1の形式で記述したものである。材料物性のデータは材料ファイル(構造材の物性データを記録したもの。)より材質名をキーとして取り込む。材料ファイルには材料(等方性材料)のE(縦弾性係数)、G(横弾性係数)、ν(ポアソン比)、ρ(密度)が記述される。複合材では二方向のE、G、νのデータが与えられる。

ここでは、通常行われる要素分割の処理を簡略化するため、構造要素の構成と形状データの多面体構成を同じとしている。これにより要素データを自動的に生成する事が出来るが、大変形解析など構造解析を詳細に行う必要がある場合は、要素分割を部分的に詳細化する、などの処理が必要となる。現時点では主翼の弾性振動などの小変形を見る事を目標とし、計算結果に高い精度を求めていないが、将来、座屈など構造破壊に至る大変形解析も行う事になれば、要素分割を行う処理を新たに追加する必要がある。

Table 1 構造の要素データ形式

構造名	STRUCTURE		名称				
材料物性(等方材)	PROP1	名称	材料名		板厚		
材料物性(複合材)	PROP2	名称	対象／非対象		積層数	一層当厚さ	
		開始層	終了層		材料名	繊維方向角	
		⋮					
拘束条件	CONST		拘束ID	拘束自由度のリスト			
	⋮						
荷重条件	LOAD		荷重ID	X成分値	Y成分値	Z成分値	
	⋮						
要素データ	QUAD		要素ID	N1	N2	N3	N4
	⋮						
要素データ	BAR		要素ID	N1	N2		
	⋮						
節点データ	GRID		節点ID	X	Y	Z	拘束ID 荷重ID
	⋮						

2.5 格子と要素間のデータ交換

連成計算では流体格子と構造要素の間で荷重と変位の伝達を行う必要がある 。このため、格子と要素が持つ計算値に対して次の様な変換を行い、各ソルバーに入力可能な形式とする。

(1) 格子点圧力から要素の節点荷重への変換

流体格子上の圧力を構造側の節点荷重に変換する。変換は表面格子点での圧力を 、格子点を含む当該要素の各節点に配分する事により行う。要素と格子の対応付けは、初期格子生成の時点で作成する対応リストを参照しながら行う。ここで、流体解析により得られた一格子点上の圧力から、当該要素の節点荷重を求める方法を Fig.2.8 に示す。Fig.2.8 は節点荷重による各辺の周りのモーメントの釣り合いを示しており、これより辺 S_1周りのモーメントの釣り合いは $PA_e g_1 = L_3 d_7 + L_4 d_4$で示される。ここで L_3, L_4は当該要素の節点 V_3, V_4に配分された荷重である。同様にして辺 S_2に対し荷重 L_1, L_4の、辺 S_3に対し荷重 L_1, L_2の、辺 S_4に対し荷重 L_2, L_3の式が得られ、これら 4 つの式を解いて節点荷重 L_1, L_2, L_3, L_4を得る事が出来る。この様にして得た節点荷重の大きさとそのベクトル表現をそれぞれ式 (2.2) と式 (2.3) に示す。式 (2.3) に於いてベクトル方向は逆法線方向とする。以上の計算を当該要素内の全格子点の圧力データに対して行い、得られる節点荷重の和を求める。この様にして得た荷重ベクトルを Table1 の LOAD 形式の荷重データに変換し、構造解析への入力データとする。

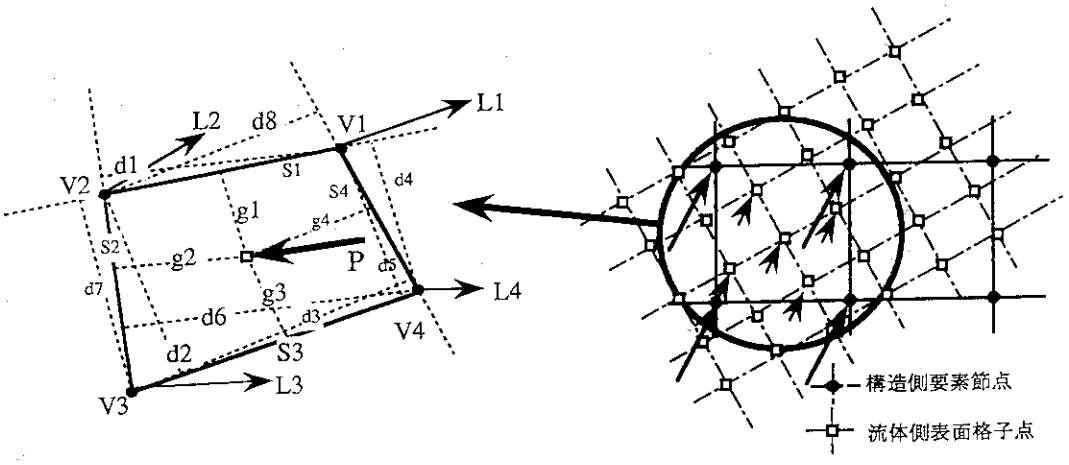


Fig.2.8 格子点圧力から節点荷重への変換

$$
\begin{aligned}
L_1 &= \frac{(g_4d_2 - g_3d_8)d_7d_6 - (g_1d_6 + g_2d_4)d_3d_2}{(d_1d_2d_3d_4 - d_5d_6d_7d_8)}P \\
L_2 &= \frac{(g_1d_3 - g_4d_7)d_6d_5 - (g_2d_5 - g_3d_1)d_4d_3}{(d_1d_2d_3d_4 - d_5d_6d_7d_8)}P \\
L_3 &= \frac{(g_3d_8 + g_4d_2)d_1d_4 - (g_1d_6 + g_2d_4)d_5d_8}{(d_1d_2d_3d_4 - d_5d_6d_7d_8)}P \\
L_4 &= \frac{(g_1d_3 - g_4d_7)d_2d_1 - (g_2d_5 - g_3d_1)d_8d_7}{(d_1d_2d_3d_4 - d_5d_6d_7d_8)}P
\end{aligned}
\tag{2.2}
$$

$$
\begin{aligned}
\boldsymbol{L_1} &= -L_1((\boldsymbol{V_4} - \boldsymbol{V_1}) \times (\boldsymbol{V_2} - \boldsymbol{V_1}))/|(\boldsymbol{V_4} - \boldsymbol{V_1}) \times (\boldsymbol{V_2} - \boldsymbol{V_1})| \\
\boldsymbol{L_2} &= -L_2((\boldsymbol{V_3} - \boldsymbol{V_2}) \times (\boldsymbol{V_1} - \boldsymbol{V_2}))/|(\boldsymbol{V_3} - \boldsymbol{V_2}) \times (\boldsymbol{V_1} - \boldsymbol{V_2})| \\
\boldsymbol{L_3} &= -L_3((\boldsymbol{V_4} - \boldsymbol{V_3}) \times (\boldsymbol{V_2} - \boldsymbol{V_3}))/|(\boldsymbol{V_4} - \boldsymbol{V_3}) \times (\boldsymbol{V_2} - \boldsymbol{V_3})| \\
\boldsymbol{L_4} &= -L_4((\boldsymbol{V_1} - \boldsymbol{V_4}) \times (\boldsymbol{V_3} - \boldsymbol{V_4}))/|(\boldsymbol{V_1} - \boldsymbol{V_4}) \times (\boldsymbol{V_3} - \boldsymbol{V_4})|
\end{aligned}
\tag{2.3}
$$

(2) 構造変形から流体格子変形への変換

構造側の変形をもとに流れ場の変形量を求める。Fig.2.9は幾何学補間により構造物の節点の変位から、当該要素の中の表面格子点の移動量を求める方法を示している。流体格子点から変形前要素の各辺に下ろした垂直点の位置を辺の長さとの比で示し、変形後要素の各辺上での位置を求め、対する辺同志の辺上の点を結ぶ直線の交点として変形後の格子点位置を求める。

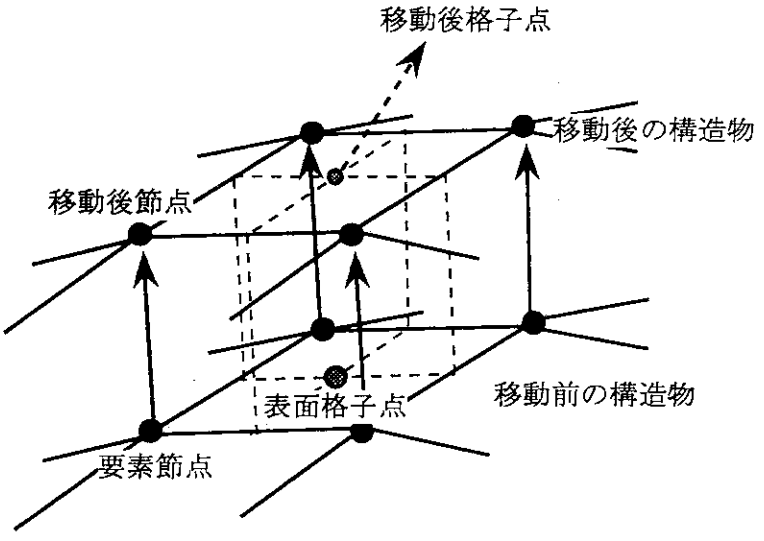


Fig.2.9 表面格子点の移動量計算

2.6 流体格子の動的変形

構造物の表面格子の移動量と速度から流れ場に置かれた流体格子の移動量と速度を求める。流体格子の移動量と速度は、Fig.2.10 と式 (2.4) に示す様、表面格子上で最大、外周格子上でゼロとなる様な線形補間により求める[7]。外周格子は流れ場の外部領域、即ち構造物の影響が及ばない十分遠方に設定された一様流との境界、を規定する格子点群の事である。

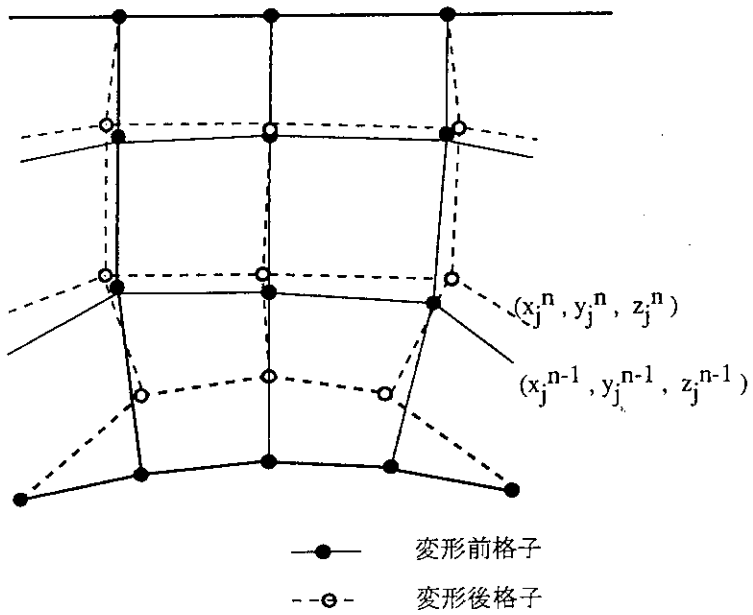


Fig.2.10 全体の流れ場格子の変形

$$
\begin{aligned}
x_j^n &= x_j^{n-1} + \Delta x_1^n \left(\frac{NJ-j}{NJ-1}\right) \\
y_j^n &= y_j^{n-1} + \Delta y_1^n \left(\frac{NJ-j}{NJ-1}\right) \\
z_j^n &= z_j^{n-1} + \Delta z_1^n \left(\frac{NJ-j}{NJ-1}\right) \\
u_j^n &= \frac{\Delta x_1^n}{\Delta t} \left(\frac{NJ-j}{NJ-1}\right) \\
v_j^n &= \frac{\Delta y_1^n}{\Delta t} \left(\frac{NJ-j}{NJ-1}\right) \\
w_j^n &= \frac{\Delta z_1^n}{\Delta t} \left(\frac{NJ-j}{NJ-1}\right)
\end{aligned}
\tag{2.4}
$$

2.7 並列処理による格子と要素の生成

形状データの生成、流体格子の生成、要素データ生成、格子と要素間でのデータ交換、流れ場の変形に対する並列処理方式を以下に述べる。

(1) 形状データ生成に於ける並列化

主翼形状データの生成では、基準翼型データの線形変換により10-20の中間断面を作り、四辺形多面体データを生成する。総面数は最大でも5000面程度と考えており、この程度の多面体データの生成を最初に一回行うだけであれば、あえて並列化の必要は無いと考える。しかし、より大量の面数とする場合にはスパン方向に領域分割を行い並列化する。基準翼型に対し線形変換を行い主翼形状を生成する過程では隣接断面の情報は不要であるため、並列化は容易である。

(2) 流体格子生成に於ける並列化

3軸方向のインデックス分割により、3次元的な領域分割による並列処理が可能である。偏微分型格子の様に周囲格子点の情報を必要としないため、PE間での通信が無く並列化は容易である。

(3) 構造要素データの生成に於ける並列化

構造要素は形状データの多面体と同一分割であるため、形状データで行う様にスパン方向での領域分割による並列化を行う。この処理は形状データの生成と同時に行う事が可能である。

(4) 格子と要素間のデータ交換に於ける並列化

流体計算と構造計算を行う都度、(要素数 × 表面格子数) 回のデータ交換が行われる。従って、全体の処理速度に与える影響は大と予測され、後述の流れ場の変形と合わせ、並列化の効果が期待出来る部分である。ここでは、要素をグループ化し各PEに割り当て並列処理を行う。空力から構造への荷重データの交換を行う際は、隣接要素と共有する節点同士で荷重のベクトル合成が必要であるが、これは、全処理が終了した段階で行う。構造から空力への変形データの交換を行う過程では、隣接要素の情報は不要である。

(5) 流れ場格子の変形に於ける並列化

この部分も構造計算を行う都度、全流体格子について座標とメトリックスの更新を行うため、流体格子の生成と同様に3次元領域分割による並列化の効果が大きいと期待出来る。

参考文献

1) Liu, M. and Gorman, D.G., "Symmetrization of Equations of Motion for Coupled Systems Subject to Fluid-Structural Interactions", Communications in Numerical Methods in Engineering, Vol.11,831-838 (1995)

2) Belanger, F. et.al., "Dynamics of Coaxial Cylinders in Laminar Annular Flow By Simultaneous Integration of the Navier-Stokes and Structural Equations", Journal of Fluids and Structures, Vol.8, 747-770 (1994)

3) Guruswamy, G.P. and Byun, C.,
"Fluid-Structural Interactions Using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures",AIAA-93-3087,Orlando,Florida,July 1993.

4) Byun,C. and Guruswamy,G.P. ,
"Wing-Body Aeroelasticity Using Finite-Difference Fluid/Finite Element Structural Equations on Parallel Computers", AIAA-94-1487-CP (1994)

5) 研野和人,"自動設計法", 現代自動制御双書 3, コロナ社, 昭 44

6) Hoffmann,K.A. and Chiang,S.T., "Computational Fluid Dynamics for Engineers-Volume1" ,Engineering Education System,1993.

7) Coen,P.G.,Sobieszczanski-Sobieski,J.,and Dollyhigh,S.M.,
"Preliminary Results from the High-Speed Airframe Integration Research Project",AIAA-92-1004,Irvine,California,February 1992.

3. 流体解析

本研究では、構造物体の変形とその物体の周囲もしくは内部を流れる流体の流れ場の解析を、物体との物理的相互作用を考慮しながら行なう。すなわち、飛行機の主翼を例として考えると、飛行機が運動状態にあるとき、翼の周りには空気が高速で流れる。翼はその周りを流れる空気の流れによって抵抗力や揚力といった力を受ける。翼は現実にはそれ自体完全な剛体ではないので、流れ場から受ける力により翼全体の変形や振動が発生する。翼の変形や振動はその周りの空気の流れに影響を与え、それはまた翼に与える空力にも影響する。本研究では、このような物体の変形についての構造解析と流れ場に関する流体解析をお互いに連成させて行なう。

本節では、このような流体解析のための数値流体力学（CFD:Computational Fluid Dynamics）手法についての概略を述べる。

3.1 基礎方程式

本研究で扱う流体は主に気体（空気）であるので、気体の運動を記述する基礎方程式として、3次元 Navier-Stokes(NS) 方程式を採用する。

固定したデカルト (Cartesian) 座標系 (x, y, z) における3次元非定常NS 方程式は以下のように表される。

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}-\frac{\partial E_v}{\partial x}-\frac{\partial F_v}{\partial y}-\frac{\partial G_v}{\partial z}=0 \tag{3.1}$$

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ e\end{pmatrix}, E=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ (e+p)u\end{pmatrix}, F=\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ (e+p)v\end{pmatrix}, G=\begin{pmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ (e+p)w\end{pmatrix}$$

$$E_v=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{yx}\\ \tau_{zx}\\ \tau_{xx}u+\tau_{yx}v\\ +\tau_{zx}w+k\frac{\partial T}{\partial x}\end{pmatrix}, F_v=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{zy}\\ \tau_{xy}u+\tau_{yy}v\\ +\tau_{zy}w+k\frac{\partial T}{\partial y}\end{pmatrix}, G_v=\begin{pmatrix}0\\ \tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\\ \tau_{xz}u+\tau_{yz}v\\ +\tau_{zz}w+k\frac{\partial T}{\partial z}\end{pmatrix}$$

$$p=(\gamma-1)\left[e-\frac{1}{2}\rho(u^2+v^2+w^2)\right]$$

ここで、ρ、p、e、T は密度、圧力、単位体積当たりのエネルギー、そして温度を表す。u、v、w は x、y、z 各方向の速度、γと k は比熱比及び熱伝導係数である。また、τは粘性応力（後述、添字は各成分）を表す。

実際の流れを解く場合には、翼の回りやエンジン内部のように複雑な境界を伴った流れ場の場合が多い。一般にこのような流れ場を解く場合には物体の形状に適合した座標系を設定し、座標変換を用いて単純な立方体（２次元では正方形）格子上で数値計算を行なう。このような座標系を物体適合座標 (body-fitted coordinate system) もしくは一般座標 (generalized coordinate system) と言う。一般座標での計算はヤコビアン (Jacobian) とメトリック (metrics) を用いて (3.1) 式をデカルト座標 (x, y, z) から一般座標 (ξ, η, ζ) へ変換することによって行なう。

本研究においても、一般的な流れ場を扱うので一般座標系による計算を行なう。また、本研究では物体の形状が時間的に変化する、すなわち格子が非定常に変形する場合の流れ場を求めることになる。そのためメトリックに時間項を考慮する。

格子の変形 (移動) を考慮した一般座標系 (ξ, η, ζ) における３次元非定常 NS 方程式は以下のように表される。

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \zeta} - \frac{\partial \hat{E}_v}{\partial \xi} - \frac{\partial \hat{F}_v}{\partial \eta} - \frac{\partial \hat{G}_v}{\partial \zeta} = 0 \tag{3.2}$$

$$\hat{Q} = \frac{1}{J}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix}$$

$$\hat{E} = \frac{1}{J}\begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U - \xi_t p \end{pmatrix}, \hat{F} = \frac{1}{J}\begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V - \eta_t p \end{pmatrix}, \hat{G} = \frac{1}{J}\begin{pmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W - \zeta_t p \end{pmatrix}$$

$$\hat{E}_v = \frac{1}{J}\begin{pmatrix} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} + \xi_z \tau_{xz} \\ \xi_x \tau_{yx} + \xi_y \tau_{yy} + \xi_z \tau_{yz} \\ \xi_x \tau_{zx} + \xi_y \tau_{zy} + \xi_z \tau_{zz} \\ \xi_x \sigma_x + \xi_y \sigma_y + \xi_z \sigma_z \end{pmatrix}, \hat{F}_v = \frac{1}{J}\begin{pmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} + \eta_z \tau_{xz} \\ \eta_x \tau_{yx} + \eta_y \tau_{yy} + \eta_z \tau_{yz} \\ \eta_x \tau_{zx} + \eta_y \tau_{zy} + \eta_z \tau_{zz} \\ \eta_x \sigma_x + \eta_y \sigma_y + \eta_z \sigma_z \end{pmatrix}$$

$$\hat{G}_v = \frac{1}{J}\begin{pmatrix} 0 \\ \zeta_x \tau_{xx} + \zeta_y \tau_{xy} + \zeta_z \tau_{xz} \\ \zeta_x \tau_{yx} + \zeta_y \tau_{yy} + \zeta_z \tau_{yz} \\ \zeta_x \tau_{zx} + \zeta_y \tau_{zy} + \zeta_z \tau_{zz} \\ \zeta_x \sigma_x + \zeta_y \sigma_y + \zeta_z \sigma_z \end{pmatrix}$$

ここで、J は Jacobian と呼ばれ、

$$J \equiv \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix}$$

と表される。ξ_x、η_x、...はメトリック (metrics) で、

$$\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta), \quad \eta_x = J(y_\zeta z_\xi - y_\xi z_\zeta), \ldots$$

のようにに表される。ここで、各添字は添字による微分を表す。
U、V、W は反変速度で次のように表される、

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w, \quad V = \eta_t + \eta_x u + \eta_y v + \eta_z w, \quad W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w$$

ξ_t、η_t、ζ_tは、

$$\xi_t = -x_t\xi_x - y_t\xi_y - z_t\xi_z, \quad \eta_t = -x_t\eta_x - y_t\eta_y - z_t\eta_z, \quad \zeta_t = -x_t\zeta_x - y_t\zeta_y - z_t\zeta_z$$

のように表され、U、V、W は格子点の移動速度を用いると、

$$U = (u - x_t)\xi_x + (v - y_t)\xi_y + (w - z_t)\xi_z$$

$$V = (u - x_t)\eta_x + (v - y_t)\eta_y + (w - z_t)\eta_z$$

$$W = (u - x_t)\zeta_x + (v - y_t)\zeta_y + (w - z_t)\zeta_z$$

と表される。

τは粘性応力成分で、以下のようになる。

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right), \tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x}\right), \tau_{zz} = \frac{2}{3}\mu\left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right), \quad \tau_{zx} = \tau_{xz} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)$$

$$\sigma_x = \tau_{xx}u + \tau_{xy}v + \tau_{xz}w + k\frac{\partial T}{\partial x}$$

$$\sigma_y = \tau_{yx}u + \tau_{yy}v + \tau_{yz}w + k\frac{\partial T}{\partial y}$$

$$\sigma_z = \tau_{zx}u + \tau_{zy}v + \tau_{zz}w + k\frac{\partial T}{\partial z}$$

kはプラントル係数P_rと定圧比熱C_pを用いて、$k = \frac{C_p\mu}{P_r}$と表される。

上式を代表的長さ、速度、時間等を用いて無次元化し、時間、空間に関して離散化し、数値計算を行なう。

レイノルズ数が大きい時、実際の流れは乱流的となり、乱流の影響を考慮しなければならない。乱流を解く方法としては様々な手法が考えられているが、(3.2) 式を直接解く方法 (DNS 法) と (3.2) 式を時間平均化し平均化によって新たに生ずるレイノルズ応力をモデル化して解く方法が代表的である。DNS 法においては、十分に乱流を捉えるためには膨大なグリッド数が必要となり、現在のスーパーコンピューターを用いても高いレイノルズ数流れを解くのは困難である。後者の方法においては、$\mu \to \mu_l + \mu_t$、$\mu/P_r \to \mu_l/P_{rl} + \mu_t/P_{rt}$(添字$l$、$t$はそれぞれ層流、乱流を表す) とすることで、形式的には (3.2) と変わらなくなるので一般にはこの方法が用いられる。乱流粘性係数μ_tを求めるためのモデルとして、代数モデル[1)2)]、一方程式モデル[3)]、二方程式モデル[4)5)]、等様々なモデルが考えられている。

本研究では、構造力学との連成計算を並列計算機上で行なうための手法開発が主な目的であるので、流体の数値計算においては乱流の影響は無視し、粘性項も省略する。一般に NS 方程式の計算では、粘性項の計算に 2-3 割の計算負荷が掛かり、物体表面で境界層を分解するために非常に細かい ($\sim 0.1/\sqrt{R_e}, R_e$:レイノルズ数) 格子間隔を取らなければならないために時間刻みが小さくなり、さらに計算負荷が大きくなる。

従って、本研究の開発段階では (3.2) 式から粘性項を省いたオイラー方程式を基礎方程式とする。解法としてはオイラー方程式も NS 方程式も粘性項の有無だけで本質的な差は無い。

3.2 計算手法

(3.2) 式の数値解法としては、Chakravarthy and Osher[6)]の 3 次精度の TVD(Total Variation Diminishing) 法を用いて差分化を行なう。時間積分は 2 次または 4 次精度の Runge-Kutta 法を用いる。TVD 法は数値振動を起こすことなくシャープに衝撃波を捉えられるとして近年一般に用いられている手法であり、Chakravarthy and Osher の TVD 法は 2 次精度風上差分と 2 次精度中心差分を線形に結合し、流束制限関数を用いて TVD 化したもので、パラメーターの取り方で精度の異なるいくつかのスキームが扱えるが、本計算では主に 3 次精度で計算を行なう。

この手法を用いた計算例を以下に示す。本研究では前述のように最終的には 3 次元の NS 方程式 (開発段階ではオイラー方程式) を用いるが、以下の計算例では 2 次元の NS 方程式をこの解法を用いて、NACA0012 と呼ばれる翼型の周りの流れを計算したものである (Fig.3.1)。流入マッハ数は 0.8 で、左図は翼型に適合するように張られた C 型計算格子である。この計算例では、NS 方程式を解いているので翼面の近傍及び翼端部分、そして後流部分に格子間隔を小さくし格子を集

中させている。右図は翼回りの等マッハ線図である。レイノルズ数は約40万である。この計算例においては翼型は固定しており、計算の並列化も行なっていない。

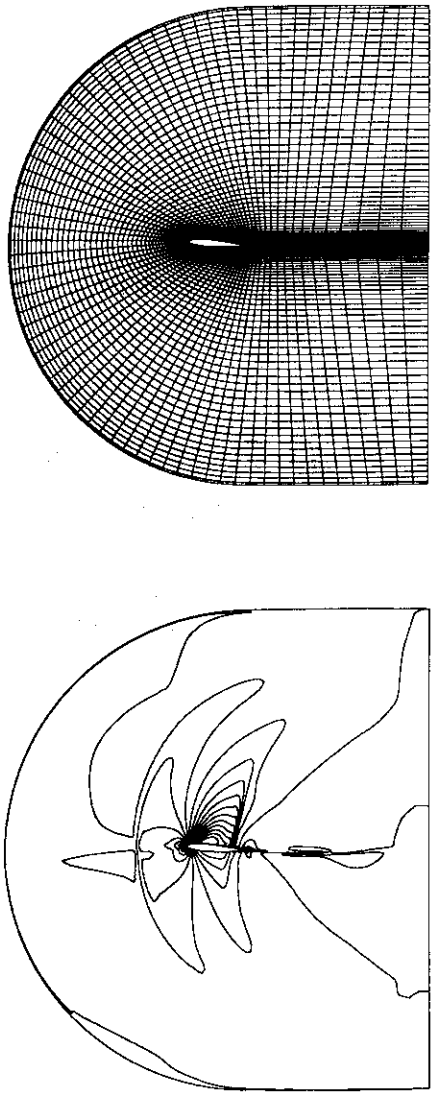


Fig.3.1 NACA0012翼回りの2次元格子図及び等マッハ図

初期条件は問題に応じて適宜設定を行なうが、Fig.3.1の計算例では初期に計算領域全体に一様な密度、圧力、そして流入速度を与えた状態から突然計算を開始する (impulsive start)。また、境界条件については物体表面に関してはNS方程式の場合には、物体表面上で速度が零、表面に垂直な温度勾配が零となるような条件（断熱、滑べり無し条件）を与え、オイラー方程式の場合は、物体表面で表面に垂直な速度成分が零、平行な成分が1内点と同じ値を持たせる条件（滑べり壁条件）を与える。計算例では上流の境界には一様流の流入条件、下流の境界には零次外挿による流出条件を設定している。

流体コードにおける計算の基本的な流れは、Fig.3.2にフローチャートとして表しているように、最初に各パラメーターの設定と初期条件の計算を行ない、格子形成プログラムによって作られた格子データをデータファイルから入力した後、時間ステップによるループに入る。各時間ステップでは、安定性条件（クーラン条件）に基づいた時間刻みを計算し、境界条件を設定する。そして、各格子点上における数値流束を計算し、各物理量の更新を行なう。設定された時刻もしくは収束判定により時間ステップを終了し結果を出力して計算を終了する。

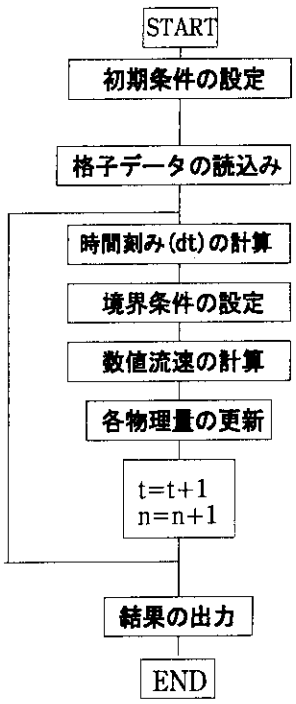


Fig.3.2 流体計算の基本的な流れ

3.3 並列化手法

並列化の方法としては、領域分割法を採用する。２節において説明した手法によって形成した流体用の格子（流体格子）をξ、η、もしくはζの方向に、内部の格子点数がほぼ均等になるように分割する。Fig.3.3は、Fig.3.1で示した２次元格子をξ方向（翼型に平行な方向）に４分割、η方向（放射状方向）に３分割した例を示している。各領域を各プロセッサに割り当てSPMD(Single Program Multiple Data)方式で並列に計算を行なう。すなわち、各プロセッサには同一の流体コードが搭載され、割り当てられた領域内の流れの計算だけを担当する。従って、プロセッサ１台での計算に対して理論的にはメモリ、計算時間共にプロセッサ台数分の１に節約できる。格子１点に対する演算量は計算領域のどの点においても同じであるので、各プロセッサに割り当てる格子点数がほぼ均等であればプロセッサ間の負荷分散はほぼ均等になる。

各方向での分割方法については、問題に応じて適当に設定し、分割数については後述する構造計算や格子計算との間の負荷分散に応じて定める。

計算には差分法を用いているので、各格子点上の物理量を計算するために隣接する上流もしくは下流の１点または２点の格子点上のデータを必要とする。従って、各領域の境界点では隣の領域のデータが必要となるために、各領域の境界から１点もしくは２点内側のデータ（２次元ならば帯状、３次元ならば面状）を隣接するプロセッサどうしで交換しなければならない。また、時間刻みはグローバルタイムステップを採用するので、各領域内においてクーラン条件を満たすような時間刻みを計算し、それらの中から全領域での最小値を求め、それを全領域に再分配しなければならない。そのため領域間での通信（データの交換）が必要となる。

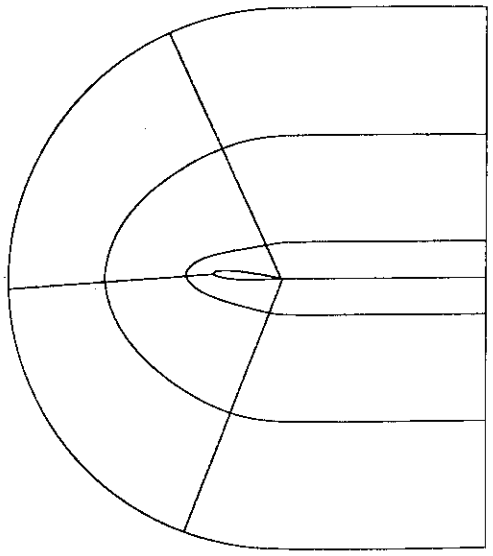


Fig.3.3 領域分割の例

領域間通信にはメッセイジパッシングという方法を用い、本計算においては MPI[7]もしくは PVM[8]ライブラリーを使用する。

MPI を用いて並列化された２次元の流体コードの一部分を以下に例として示す。[1] 行で設定する (nng,nne) が計算領域全体の格子点数で、[2] 行における ncpu1、ncpu2 は i、j 方向での分割数である。その分割数に応じて各プロセッサ（各領域）の持つ i、j 方向の格子点数が [3] 行の (ng,ne) で定義されている。従って、各プロセッサの持つ配列の大きさは (ng,ne) となる（[4] 行）。

```
c
      parameter(nng=100,nne=100) ..................................... [1]
      parameter(ncpu1=2,ncpu2=2,ncpu=ncpu1*ncpu2) ..................... [2]
      parameter(ng=(nng+2)/ncpu1+1,ne=(nne+2)/ncpu2+1) ................ [3]
c
```

```
      dimension q(ng,ne),qo(ng,ne),fo(ng,ne) ..................................[4]
c
      iprocs(1)=ncpu1
      iprocs(2)=ncpu2
      bound(1) =.false.
      bound(2) =.false.
c
      call mpi_init(ierr) ..................................................[5]
      call mpi_comm_size(mpi_comm_world,nprocs,ierr) .......................[6]
      call mpi_comm_rank(mpi_comm_world,myrank,ierr) .......................[7]
c
      call mpi_cart_create(mpi_comm_world,2,iprocs,bound, .false.,icomm,ierr) ....[8]
      call mpi_cart_coords(icomm,myrank,2,mycoord,ierr) ....................[9]
      call para_range(1,nng,iprocs(1),mycoord(1),ista,iend) ................[10]
      call para_range(1,nne,iprocs(2),mycoord(2),jsta,jend) ................[11]
c
      do 100 j=jsta,jend
        jj=j-jsta+1
        do 100 i=ista,iend
          ii=i-ista+1
          voliv=1.0d0/vol(ii,jj)
          q(ii,jj,1)=qo(ii,jj,1)-dt*(fo(ii,jj,1)-fo(ii-1,jj,1))*voliv
  100 continue
c
```

[5]、[6]、[7]行でMPIの初期化と各プロセッサにランクを割り当てている。[8]行目でプロセス座標テーブルを作成し、[9]行目で各ランクに対しテーブル上での座標を与えている。[10]、[11]行では各ランクが、全領域の中で受け持つ領域を表すi座標とj座標それぞれの範囲を与えられる。

次のDOループは、流体の計算の中で最も頻繁に現れる典型的な2重ループである。インデックスi、jは全領域内におけるインデックスで、ii、jjは各分割領域内でのインデックスを表す。iの範囲ista、iend、jの範囲jsta、jendは各ランクごとに[6]、[7]で与えられているが、ii、jjは各ランク内で1から始まるインデックスに対応する。

このようにして、同じプログラムを用いて計算領域の異なる領域を各プロセッサに担当させることが可能になる。

3.4 MPMD 並列化による移動格子における格子形成と流体計算の例

前節で示したように、格子が固定されている場合は、別の格子形成プログラムによって格子を形成し、格子データを流体の計算の最初の段階で一度読み込むだけでいいが、格子が非定常に変形をする場合、格子が移動するために格子点の座標、メトリックの値、ヤコビアン、等は時々刻々変化する。そのため、各時間ステップにおいて格子の再形成を行ない、それらの値を更新する必要がある。

本研究においては、このような流体格子の変形は流体格子に接する物体の変形に起因するものであり、物体の構造計算との連成計算を MPMD 方式で並列化して行なう（第５章参照）。構造計算との連成により、格子計算は流体計算プログラムに組み込まれ、構造計算プログラムと流体計算プログラムが MPMD 形式で並列化されることになるが、本節では MPMD 方式の一例として、物体の構造計算を含まない場合について、流体プログラムと格子形成プログラムの MPMD 方式による並列化について、迎角が単振動変化する翼周りの流れ場を計算する場合を例として簡単に説明する。

物体の構造計算を含まない場合、格子の変形と流体計算を逐次処理として考えると、格子計算プログラムを流体プログラム中に組み込み各時間ステップ毎に格子形成を行なわせることになる。すなわち、格子を形成し格子データを計算した後、この格子上での流れを計算する。そして、次の時間ステップにおいて再び格子を形成し格子データを計算して、流れ場を計算する。このような、格子形成と流体計算を各時間ステップで交互に逐次的に処理をすることで、移動格子上での流れ場の計算が行なえる。

しかしながら、この場合、基本的に格子形成と流体計算は独立しているので、それぞれを並列に処理することが可能である。すなわち、ある時間ステップで与えられた格子上での流れ場を計算している間に、次のステップでの格子の計算を行なうことが可能である。並列計算機上の別のプロセッサ上で格子形成プログラムを流体プログラムと連係して動かして、格子データを流体プログラムの各時間ステップ毎に転送することによって、流体計算と格子形成の並列計算が可能になる。

このような、全く異なるプログラムがデータの転送を行ない同期をとりながら異なるプロセッサ上で並列に動く形式を MPMD (Multiple Program Multiple Data) 方式と呼ぶ。それによって、計算負荷をそれぞれ異なるプロセッサに分散させることが可能になり計算の高速化やメモリーの節約が計れる。さらに、格子計算プログラムと流体計算プログラムが独立しているのでそれぞれのプログラムの改変やデバッグも容易となる。また、それぞれのプログラムはそれ自身並列化され複数のプロセッサ上で SPMD 方式で動くことになる。

格子形成と流体計算を独立に並列して動かす場合の計算の基本的な流れを Fig.3.4 にフローチャートとして表す。この場合、初期条件として、前節の計算例のように固定した格子上で既に収束し定常解として得られた翼の回りの流れ場を初期データとして読み込み、その状態から計算を始める。またその状態の格子データも同様に読み込み、その後、時間ステップのループに入る。各時間ス

テップでは、まず時間刻みを計算した後、時刻と時間刻みを格子形成プログラムに転送し、流れ場の計算を行なう。

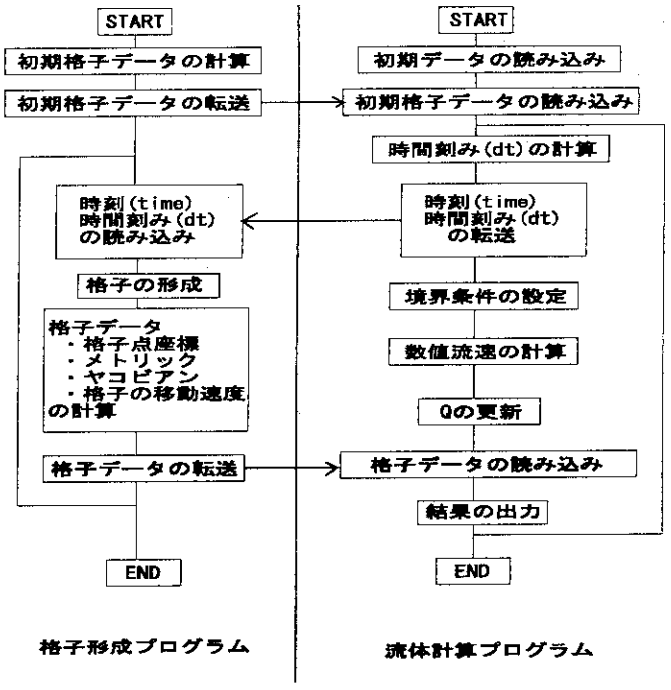


Fig.3.4 格子及び流体プログラムの並列計算の例

格子形成プログラムは時刻と時間刻み dt を受けとると、その時刻から dt 経た後の翼の迎角を計算しその翼型に適合した格子を形成し、その格子に対する格子データを計算する。すなわち、流体プログラムが時刻 t の流れ場を計算している間に、時刻 t+dt の格子データを計算する。

流体プログラムは、流れ場の計算が終了した時点で、格子プログラムから次の時間ステップで用いる格子データを受けとり、次のステップへ進む。その繰り返しとなる。このように、流れ場の計算と格子の形成の計算が同時期に、別のプロセッサ上で行なわれることになる。格子形成と流体計算を同一のプログラム内で行なう場合、格子の計算の後、流れ場の計算を行なわなければならず（もしくはその逆)、以上のような並列化をすることによって、計算の高速化が期待される。

格子形成プログラムと流体プログラムはそれぞれ並列化され複数のプロセッサで計算されるが、1ステップでの計算時間が均等になるように、それぞれの並列度（プロセッサ台数）を調節することで効率の良い計算が行なえる。

MPI を用いて、MPMD 方式で格子形成プログラムと流体計算プログラムを並列化するためには、使用する全プロセッサを2つのグループに分けそれぞれのプログラムに割り当てる、それぞれのプロセッサグループにコミュニケータを定義する（例えば、格子プログラムに icomm_grid、流体プログラムに icomm_cfd)。それぞれのグループ内では定義されたコミュニケータを用いて、前述のような SPMD 方式で並列化を行なう。グループ間の通信、すなわち格子計算プログラムと流体

計算プログラムの間の通信は、全プロセッサを定義するコミュニケータ（通常 mpi_comm_world）を用いて行なうことが出来る（Fig.3.5 参照）。以下に、基本的なプログラムの一部を示す。

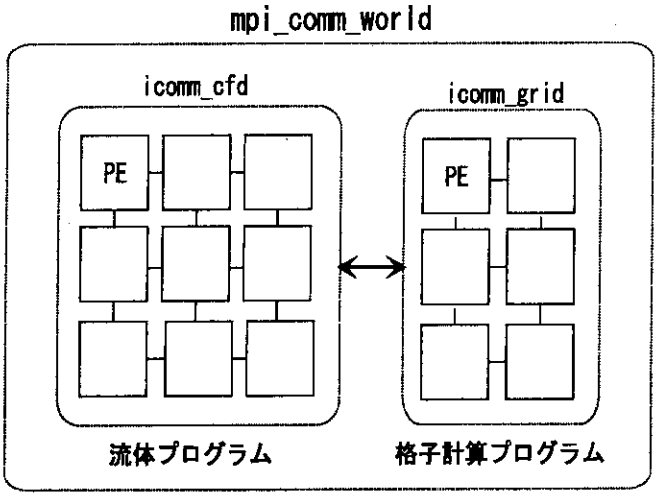


Fig.3.5 プロセッサグループの割り当て

```
      call mpi_init(ierr) ..............................................[A1]
      call mpi_comm_size(mpi_comm_world,nprocs,ierr) ...................[A2]
      call mpi_comm_rank(mpi_comm_world,myrank,ierr) ...................[A3]
c
      icolor=1
      key =myrank
      call mpi_comm_split(mpi_comm_world,icolor,key,icomm_cfd,ierr) ....[A4]
      call mpi_comm_size(icomm_cfd,newprocs,ierr) ......................[A5]
      call mpi_comm_rank(icomm_cfd,newrank,ierr) .......................[A6]
c
      call mpi_send(time,1,mpi_double_precision,nprocs-1,1, mpi_comm_world,ierr) ....[A7]
      call mpi_send(dt,1,mpi_double_precision,nprocs-1,1, mpi_comm_world,ierr) .....[A8]

      call mpi_init(ierr) ..............................................[B1]
      call mpi_comm_size(mpi_comm_world,nprocs,ierr) ...................[B2]
      call mpi_comm_rank(mpi_comm_world,myrank,ierr) ...................[B3]
c
      icolor=2
```

```
      key =myrank
      call mpi_comm_split(mpi_comm_world,icolor,key, icomm_grid,ierr) ..............[B4]
      call mpi_comm_size(icomm_grid,newprocs,ierr) ..................................[B5]
      call mpi_comm_rank(icomm_grid,newrank,ierr) ...................................[B6]
c
      call mpi_recv(time,1,mpi_double_precision,0,1, mpi_comm_world,istatus,ierr) .....[B7]
      call mpi_recv(dt,1,mpi_double_precision,0,1, mpi_comm_world,istatus,ierr) .......[B8]
```

Aで示された行番号を持つプログラムが流体計算、Bで示された行番号を持つプログラムが格子計算とする。[A1]、[A2]、[A3]でMPIの初期化、全プロセッサ数、そして全プロセッサ中での自分のランクを得る。このランクはグループ間の通信に用いられる。[A4]で全プロッセッサをグループに分け、ここではicomm_cfdというコミュニーケータで定義している。[A5]、[A6]で、icomm_cfdのコミュニケータを持つ新しいグループのプロセッサ数及びグループ内でのランクを得ている。このランクは、グループ内での通信に用いられる。[A7]、[A8]で、時刻timeと時間刻みdtを格子形成のプログラムに転送する。この時、コミュニケータは全体を表すmpi_comm_world、宛先のランクは全体の中でのランク（この例ではnprocs-1）を与える。流体プログラム中では、icomm_cfdというコミュニケータとグループ内のランクnewrank、グループのプロセッサ数newprocsを用いることで前述のSPMD方式のプログラムと同じになる。

同様に、格子形成のプログラムにおいても、[B1]–[B3]でMPIの初期化、全プロセッサ数、全体の中でのランクを得、[B4]–[B6]で、グループのコミュニケータ（ここではicomm_grid）の定義とグループ内でのプロセッサ数、ランクを得ている。[B7]、[B8]で、流体プログラムから転送された、時刻time、時間刻みdtを受けとる。

以上のようにすることによって、格子計算プログラムと流体計算プログラムをMPMD方式で、並列化しデータのやりとりが可能になる。それぞれのプログラムは、それぞれのグループ内で前の節で述べたようにSPMD方式で並列化され実行される。

参考文献

1) Cebeci,T. and Smith,A.M.O.,1974, in Analysis of Turbulent Boundary Layers, New York: Academic Press

2) Baldwin,B. and Lomax,H.,1978, "Thin layer approximation and algebraic model for separated turbulent flow",AIAA paper 78-0257

3) Launder,B.E. and Spalding,B.,1972, in Mathmatical Models of Turbulence, New York:Academic Press

4) Jones,W.,P. and Launder,B.,E.,1972, "The prediction of laminarization with a two-equation model of turbulence",International Developments in Heat Transfer, **15**,303-314

5) Peter,V.L.,Rodi,W., and Scheurer,G.,1985, "Turbulence models for near-wall and low-Reynolds number flows: a review",AIAA Journal **23**,1308-1319

6) Chakravarthy,S.,R. and Osher,S.,1985, "A new class of high accuracy TVD schemes for hyperbolic conservation laws",AIAA paper 85-0363

7) Gropp,W.,Lusk,E., and Skjellum,A.,1994, in Using MPI:Portable Parallel Programming with the Message-Passing Interface, The MIT Press

8) Geist,A.,Beguelin,A.,Dongarra,J.,Jiang,W.,Manchek,R., and Sunderam,V.,1994, in PVM:Parallel Virtual Machine - A Users' Guide and Turorial for Networked Parallel Computing, The MIT Press

4. 構造解析

Nomenclature

A_e	要素面積
$[B]$	歪み変位マトリックス
$[C]$	全体減衰マトリックス
$[c]$	要素減衰マトリックス
$[D]$	歪み応力マトリックス
e	要素
$[F]$	荷重ベクトル
g	グローバル
H_1, H_2	ガウス数値積分の重み係数
$[I]$	単位マトリックス
$[K]$	全体剛性マトリックス
$[k]$	要素剛性マトリックス
$\boldsymbol{L}$	荷重ベクトル、または要素座標
$[M]$	全体質量マトリックス
$[m]$	要素質量マトリックス
$[N]$	形状関数マトリックス
S_{panel}	燃料タンク下面に相当するパネル面積
$[T]$	座標変換マトリックス
t	板厚
u	節点の変位
V_f	燃料容量
α,β	Rayleigh 減衰パラメータ
ρ	材料密度
ρ_f	燃料密度

4.1 概要

前章で述べた様に、航空機は空気力を受け機体に振動や変形を生じながら飛行する。一般に航空機のCFDでは機体は剛体として計算される事が多いが、弾性体として扱う事により変形を伴った現実に近い解析を行う事が期待出来る。この様なモデルを通称Elastic Aircraftと呼ぶが、機体の変形は空力だけでなく燃料消費による慣性重量の変化、旋回などの際の運動荷重の増大、空力加熱による剛性低下などからも大きな影響を受ける。また、今後適用率が増大すると見込まれる複合材の適用では、積層パターンなどを変える事により変形状態が大きく変化する。

前章では変形する機体の空力計算法について述べたが、本章では空力計算への入力となる構造変形を計算する方法を述べる。特に、長距離高速飛行を行う超音速旅客機を意識し燃料重量、運動荷重、空力加熱、及び複合材構造の効果を考慮した案を示す。

航空機の空力弾性計算は一般の構造解析での動解析問題と同様に解かれる。即ち、式(4.1)をuについて解き空力荷重による構造変形を計算する。

$$[\boldsymbol{M}]\,\ddot{u} + [\boldsymbol{C}]\,\dot{u} + [\boldsymbol{K}]\,u \;=\; [\boldsymbol{F}] \tag{4.1}$$

ここで、$[\boldsymbol{M}]$は質量マトリックス、$[\boldsymbol{C}]$は減衰マトリックス、$[\boldsymbol{K}]$は剛性マトリックス、$[\boldsymbol{F}]$は流体ソルバーにより計算した空力荷重ベクトルである。式(4.1)を解く手法にはモード重畳法(Mode Superposition)と直接積分による解法がある。モード重畳法は左辺式の固有振動数成分の重ね合わせにより任意時刻の変位を求める近似解法で、マトリックス計算を行わないため大自由度の場合でも計算時間の低減が図れる反面、複雑な構造や高次モード形状の高精度の計算が困難である[1)]。直接積分法は複雑構造に対しても厳密解を与えるが、大自由度の場合マトリックス計算に多大な処理時間を要する。 空力弾性連成計算コードには、NASAのENSAERO[1) 2)]やダグラス社のADOP[3)]など、直接積分法のNewmark β法や後述の線形加速度法[4)]などとモード重畳法を併用しているものがある。ここでは複雑構造への適用や、将来大変形解析を行う可能性を考慮し、直接積分による求解を前提に話を進める。

4.2 計算手順

要素データ、拘束条件、流体解析プログラムが計算した空力データを用いて係数マトリックスを作り式(4.1)を解く。前回計算値を初期条件とし、指定された時刻の変形量を求め、流体解析プログラムに渡す。Fig.4.1は、入力データから要素マトリックスを作り、全体マトリックスを合成し、節点変位と速度を求める過程を示している。

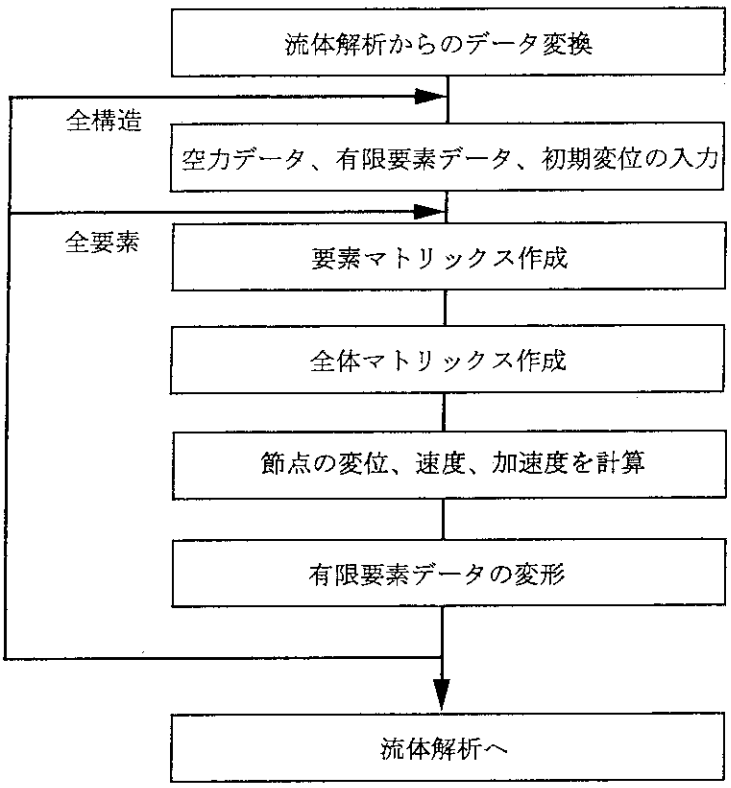


Fig.4.1 計算手順

4.3 係数マトリクスの作成

有限要素データの材質、拘束条件、荷重条件、要素、節点などのデータから、式(4.1)で用いる係数マトリックスを作成する[4)5)6)]。通常、要素毎にマトリックスを作り、全体座標系に変換後、全自由度分のマトリックスを合成するという過程で行う。各マトリックスの内容などについてはAppendixに記載する。

質量マトリックスは構造自体の重量に加え、燃料重量も考慮して作る。大型旅客機の主翼には自重の30 - 40 %に相当する燃料が搭載されており、燃料重量が構造変形に与える影響は非常に大きい。これを考慮するため、燃料タンクとして使われる主翼中央ボックス構造の各節点に、式(4.2)の集中質量(Lumped Mass)を割り当てる。式(4.2)は燃料重量を加えた質量を要素節点の三成分に配分している。隣接要素の質量は全体マトリックス作成の際、要素マトリックスを足し込む過程で加算される。

式(4.3)は剛性マトリックスを作る数値積分式である。ここで、$[\boldsymbol{B}]$は歪みと変位マトリックス、$[\boldsymbol{D}]$は歪みと応力マトリックスである。$[\boldsymbol{D}]$を作る際に材料物性を変更する事によって各種構造材に対応させる事が出来る。ここでは金属材(等方性材料)、複合材(多層異方性材料)、空力加熱による剛性低下を考慮した方式をそれぞれ Appendix -A-、4.5 と Appendix -B-、及び 4.6 に示している。

減衰マトリックス(式(4.4))は、結合のゆるみなどによる構造物内部でのエネルギー散逸を考慮したものである。αとβは Rayleigh 減衰パラメータである。

式(4.1)の右辺荷重ベクトルは空力荷重から作られるもので、式(4.5)に示す様荷重を受ける節点(ここでは第 i 節点)の並進 3 自由度成分に設定する。荷重は面垂直方向にかかる圧力とし、モーメント成分は 0. とする。

これらの要素マトリックスを合成し全自由度のマトリックスを作る。要素マトリックスは要素座標系で定義された部分マトリックスであるため、全体座標系に変換し全体マトリックスに足し込んでいく。要素座標系から全体座標系への変換は、次の座標変換マトリックス$[T]$(式(4.6))により行う。なお、$[T]$の内容は Appendix -A-に示している。

$$[\boldsymbol{m}] = \frac{1}{4}\left(A_e t \rho_s + \frac{V_f \rho_f}{N_e}\right)[\boldsymbol{I}] \tag{4.2}$$

$$[\boldsymbol{k}] = det\,|\boldsymbol{J}| \sum_{i=1}^{4}\sum_{j=1}^{4} H_1 H_2 [\boldsymbol{B}]^T [\boldsymbol{D}] [\boldsymbol{B}] \tag{4.3}$$

$$[\boldsymbol{c}] = \alpha [\boldsymbol{m}] + \beta [\boldsymbol{k}] \tag{4.4}$$

$$[\boldsymbol{F}] = \begin{bmatrix} \vdots \\ L_{ix} \\ L_{iy} \\ L_{iz} \\ 0. \\ 0. \\ \vdots \end{bmatrix} \tag{4.5}$$

$$[K]_g = [T]^T [K]_e [T] \tag{4.6}$$

4.4 直接時間積分による解法

係数マトリックスの作成を行ったら次に運動方程式を解き変位計算を行う。
式(4.1)を時間積分により解く方法には中心差分法、線形加速度法(Linear Acceleration Method)、Wilson法、Crank-Nicolson法、Newmark法などがあるが、いずれも前回(あるいは前々回)計算値を初期値として変位、速度、加速度を求めるものである。例として、式(4.7)に線形加速度法により時刻t_nでの変位u_n、速度$\dot{u}_n$、及び加速度$\ddot{u}_n$を初期値として、時刻t_{n+1}でのそれぞれの値を求める手順を示す[6)]。

$$\left([K]+\tfrac{3}{\Delta t_n}[C]+\tfrac{6}{\Delta t_n^2}[M]\right)\{u\}_{n+1}=$$

$$\{F\}_{n+1}+[M]\left(2\{\ddot{u}\}_n+\frac{6}{\Delta t_n}\{\dot{u}\}_n+\frac{6}{\Delta t_n^2}\{u\}_n\right)+[C]\left(\frac{\Delta t_n}{2}\{\ddot{u}\}_n+2\{\dot{u}\}_n+\frac{3}{\Delta t_n}\{u\}_n\right) \quad (4.7)$$

$$\begin{aligned}\{\dot{u}\}_{n+1} &= \frac{3}{\Delta t_n}\left(\{u\}_{n+1}-\{u\}_n\right)-2\{\dot{u}\}_n-\frac{\Delta t_n}{2}\{\ddot{u}\}_n\\ \{\ddot{u}\}_{n+1} &= \frac{6}{\Delta t_n^2}\left(\{u\}_{n+1}-\{u\}_n\right)-\frac{6}{\Delta t_n}\{\dot{u}\}_n-2\{\ddot{u}\}_n\\ t_{n+1} &= t_n+\Delta t_n\end{aligned}$$

ここで、重力や運動荷重による変形を考慮するため、重力加速度と運動荷重倍数(航空機の機動により生じる加速度で通常Gと呼ばれる。)のベクトル和を$\ddot{u}$の初期値に設定する。運動荷重倍数と方向ベクトルは燃料残量と供に計算条件として外部より与える事とする。

4.5 複合材の適用

航空機用構造材として、炭素繊維をエポキシ樹脂で固めたグラファイト／エポキシ複合材等が使われる様になり、軍用機では構造重量の20%以上の適用がなされたものがある。また、次世代超音速旅客機では重量軽減のために機体のほぼ全域に対しハニカムスキン複合材の適用が検討されている[9)]。航空機に対しては繊維方向が異なる複数の素材を積層して用いる。積層パターンの変更により強度方向を変える事が出来るため、適用部位の応力方向に合わせて特定方向の強度を高くする空力弾性テーラリング(Aeroelastic Tayloring)が試みられている。NASAのX-29前進翼実験機は主翼に0度、90度、+45度、-45度のグラファイト／エポキシ複合材を使い、空力弾性テーラリングを適用している[7)]。

この様な構造を考慮しFig.4.2に示す積層モデルを設定する。図は全体がn層から成りθ_kの方向角パターンを持つ、板厚tの複合材を示している。複合材の材質データには二方向の縦／横弾性係数及びポアソン比、全積層数、各層の繊維角が指定される。以下に複合材のデータセット形式を示す。

/材質名/構造名/密度/積層パターン/積層数/一層厚/開始層/終了層/方向角/Ex/Ey/Gxy/ν_{xy}/ν_{yx}

この様な複合材モデルの[*D*]マトリックスを Appendix -B-に示している[8)]。

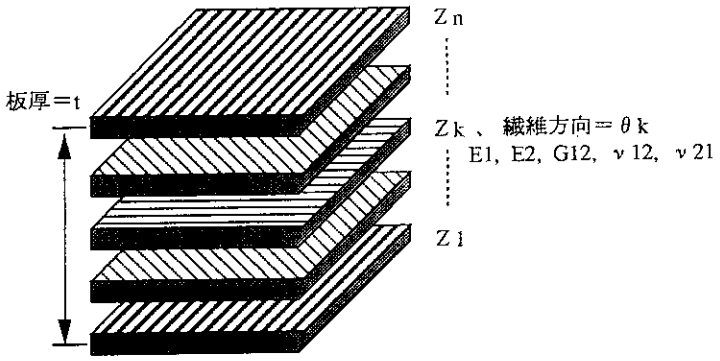


Fig.4.2 多層複合材のモデル

4.6 空力加熱による剛性低下の考慮

現在、米国で研究が進められている次世代超音速旅客機はマッハ 2.5-3.0 の巡航速度が計画されており、この時の空力加熱は最高約 400 度 C に達する。このため機首や主翼前縁などの高温部には、従来のアルミ合金に替わりチタン合金やカーボン/ポリイミド材などの耐熱性が高い新複合材の使用が検討されている[9)]。

空力加熱による剛性低下を考慮するには縦弾性係数 E、横弾性係数 G、ポアソン比νの温度特性データが必要である。現時点ではこの様なデータは入手してないが、 Fig.4.3 の様に [*D*]マトリックスの作成に際し、温度依存データの 取り込みが可能な様にプログラムを作っておく事は難しい事ではない。温度は流体解析側で圧力データと一緒に計算出来る為、空力データと供に持って来る事が出来る。

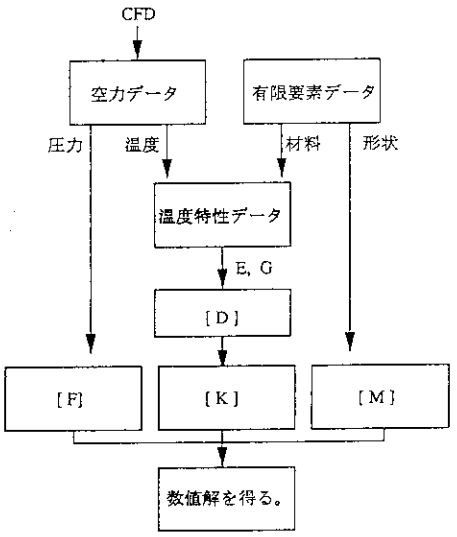


Fig.4.3 温度依存を考慮した剛性マトリックス作成

4.7 並列処理の適用

構造解析では一節点当り5自由度のデータが作られ、単純な見積りでは(5 × 全節点数)のサイズのマトリックスを数値積分を行って作成し、解くという処理を行う。しかもこれをある時間ステップで繰り返すため高速処理は必須である。ここでは、解析領域毎に要素をグループ化し並列処理を行う案を示す。

前章で述べた様に並列化は、流体解析とはMPMD方式により行うが構造解析の中での処理はSPMD(Single Program Multiple Data)方式により行う。各PEには要素グループ毎の計算領域が割り当てられる。流体格子とのインタフェースプログラムとは同じ領域分割とする。並列処理の流れは、Fig.4.4に示す様、四段階で行う事が考えられる。

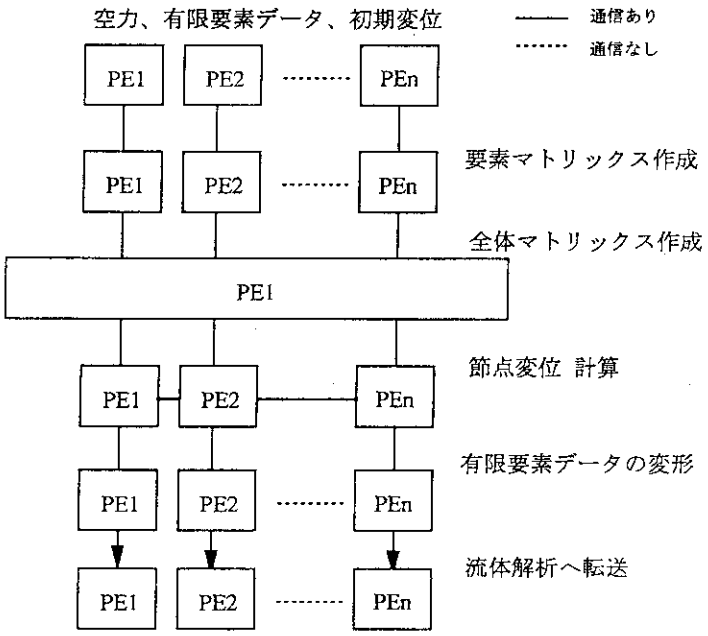


Fig.4.4 構造解析の並列処理の流れ

(1) 入力段階

流体/構造間でのデータ交換プログラムから領域毎に空力データを受け取る。この過程では領域間でのデータ交換は不要で、並列処理に於いてPE間の通信は発生しない。

(2) 要素マトリックス作成段階

有限要素データと空力データを用いて要素マトリックスを作る。要素マトリックスの生成は各要素毎に独立しているため、ここでもPE間の通信を行わない並列化が可能である。

(3) 全体マトリックス作成段階

要素マトリックスを合成し全体マトリックスを合成する過程は単一CPUで行うため、並列処理は行わない。

(4) 直接積分による変位計算の段階

前過程で得られた全体マトリックスを再度領域分割し、領域毎に直接積分計算を行い変位や速度を求める。この過程は有限要素法の各過程の中で最も計算時間を要する部分で、並列化の効果が期待出来る部分である。ここでは、変位境界条件による反復領域分割法を用いた並列化が有効と考えられる[10]。反復領域分割法では領域境界の節点に強制変位を初期値として与え、領域毎に有限要素法計算を行う。領域境界に生じる反力がつり合うまで、変位を更新しながら計算を繰り返す。このため、各領域間で変位と反力の受け渡しが必要になり、PE間での通信が発生する。

(5) 有限要素データ更新段階

計算結果をもとに変形後の有限要素データを生成する。これは次の計算ステップの初期値として用いるため記憶される。領域毎に独立した処理が可能であるためPE間の通信は発生しない。

(6) 流体側への変形データの受け渡しの段階

構造物節点の変形量をもとに流れ場側の表面格子点の移動量を計算し、流体解析側へ渡す。単純なデータ変換を行うだけであるため、計算領域の間でのデータ交換は不要でPE間の通信は発生しない。

参考文献

1) MacMurdy,D.E.,Guruswamy,G.P. and Kapania,R.K., "Static Aeroelastic Analysis of Wings Using Euler/Navier-Stokes Equations Coupled with Improved Wing-Box Finite Element Structures",AIAA-94-1587-CP,1993,pp. 2146-2158.

2) Byun,C. and Guruswamy,G.P.,
"Wing-Body Aeroelasticity Using Finite-Difference Fluid/Finite-Element Structural Equations on Parallel Computers",AIAA-94-1487-CP,1994,pp. 1356-1365.

3) Dodd,A.J.,Kadrinka,K.E.,Loikkanen,M.J.,Rommel,B.A.,Sikes,G.D.,Strong,R.C.,and Tzong,T.J.,"Aeroelastic Design Optim ization Program",J. of Aircraft,Vol. 27, NO. 12,1990,pp. 1028-1036.

4) 矢川元基, 吉村忍,"有限要素法", 計算力学と CAE シリーズ 1, 培風館,1994,pp. 171-179.

5) 鷲津久一郎, 宮本　博, 山田嘉昭, 山本善之, 川井忠彦," 有限要素法ハンドブック (I. 基礎編)", 培風館,1982.

6) I.M. スミス著, 戸川隼人訳," 有限要素法のプログラミング", ワイリージャパン,1984.

7) Miura,H. and Neil,D.J.,"Structural Optimization: Status and Promise/Applications to Fixed-Wing Aircraft and Spacecraft",Progress in Astronautics and Aeronautics, Vol. 150, AIAA,1992,pp. 705-742.

8) Hoskin,B.C. and Baker,A.A.,"Composite Materials for Aircraft Structures", AIAA Education Series,1986.

9) Grande,D.L., "HSCT Materials & Structures Program Introduction" ,Boeing Commercial Airplane Group, Seattle, Washington, June 20-23, 1994.

10) 矢川元基, 曽根田直樹,"パラレルコンピューティング", 計算力学と CAE シリーズ 7, 培風館

5. 連成解析システムの構成

5.1 流体、構造の連成解析

本研究のように、複数のプロセス(流体、構造、格子生成)を連成させる場合に、それぞれのプロセスをどのように組み合わせて実行するかについてはいくつかの手法が考えられる。まず、流体と構造解析に関しては両者の方程式系を連立させる形で一斉に解いてしまうもの[1)2)]と、それに対しそれぞれの解析は独立したものとして解き、お互いのデータをもう一方の計算条件の一部として渡し合う事により連成をすすめるやり方が考えられる[3)4)5)]。後者のアプローチをとる場合にも、各プロセスの実行の仕方としては逐次的に行なうものと並列に行なう構成を選ぶことが出来る。すなわち、前者は一つのプロセスを実行しているときにはそれだけを行ない、それが終了してから次のプロセスを行なうというように計算を進めて行く、それに対し後者は複数のプロセスを同時に実行させようとするものである(Fig.5.1)。

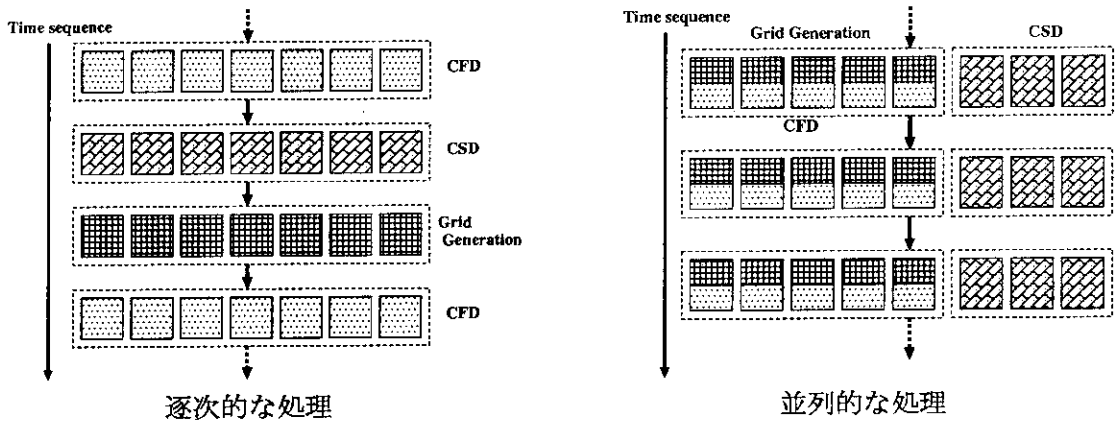


Fig.5.1 複数プロセスの処理法

本研究では分散メモリ型並列計算環境上で開発することを念頭に置き、それぞれのプロセスを独立したものとして扱い、また並列に実行するやり方を採用する。この方法を採る理由、利点としては次のことがあげられる。

1. それぞれのプロセスを独立したプログラムとして構成出来る。すなわち手法などを他と関係なしに選ぶことが出来る。これはそれぞれの手法を改良したり、異なる手法を用いたりということが簡単に行なえるので、システムとしてより高い柔軟性を持たせる事が可能になる。
2. 並列環境において逐次的に複数のプロセスを実行させる場合、全てのプロセスを同じ数のノードに分割することになる。プロセスによる計算量(負荷)が大幅に異なる場合、このアプローチでは最適な並列化効率が得られない事があるのが予想される。これに対し、並列的に構成することによりそれぞれのプロセスにおける並列化も独立に行なうことが出来る。

ここで二つ目の利点としてそれぞれの並列化を独立に出来るように書いたが、後の章で述べる

ようにそれぞれの１時間ステップ毎の計算時間が異なることによる並列化効率の悪化が起こるので、その意味では独立とは言えない。それぞれのプロセスの計算時間のバランスがとれるように互いの分割数を決定することになる。

具体的に本研究における計算の進行、各プロセス間の連携について見て行きたい。対象とする計算領域は例えば Fig.5.2 のようなものである。すなわち、構造解析が対象とする翼形状があり、その周囲に翼周りの空力を解析するための流体解析用の格子を配置している。

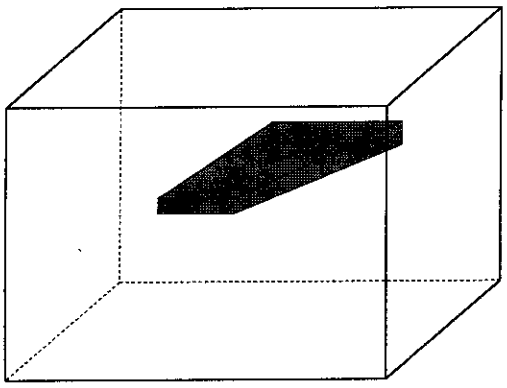


Fig.5.2 計算領域概念図

構造解析の計算格子は翼型を構成しており、その表面が流体格子の境界の一部となっている。流体の計算格子は翼型の表面と外部境界からなる領域内において生成される。お互いの計算結果は従って共通に扱うことになる翼の表面を介して相互に受け渡される。流体解析からは翼表面の圧力値が構造解析の表面荷重として与えられ、構造解析からは翼形状の変形による格子境界の座標値が新たな流体の計算格子生成の情報として与えられる。加えて格子点の移動速度も流体解析プログラムで利用される。逐次的な計算の流れは Fig.5.3 のようになる。

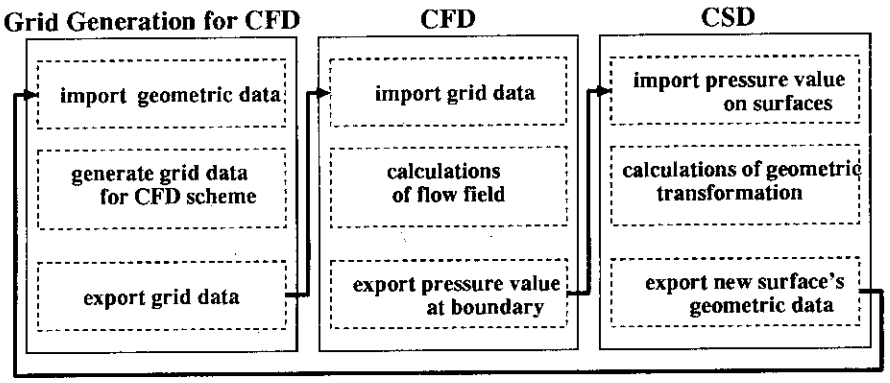


Fig.5.3 計算の流れ

5.2 時間の同期

異なる複数の解析プログラムを連成させるときには、各解析計算の物理時間の進行を同期させなければならない。これには２つの側面があって、ひとつには各解析プログラムで扱う時間の物理

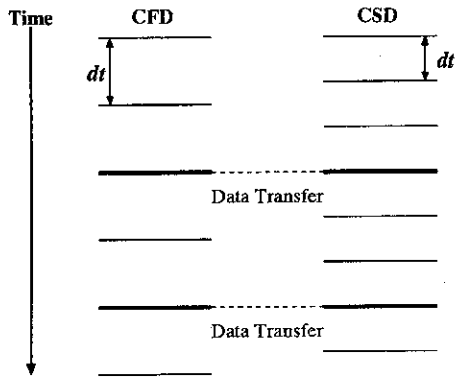


Fig.5.4 時間刻みの設定

的な単位を合わせなくてはならないこと、もうひとつはそれを満たした上で各プログラムのデータを交換するのを同じ計算の経過時間毎において行なうということである。物理的な単位を合わせるということはそれぞれのプログラムで扱っている物理系で異なる単位系を採用している場合には、時間について考えるとき同一の単位に修正するということである。各解析プログラムにおいてはそれぞれの計算手法による安定性から時間刻み幅が決定されるが、それぞれについての最適な値が一致するとは限らない。小さい方に一致させる方法も考えられるが、それも含めてより柔軟な扱いをするためにそれぞれの時間刻みの整数倍が一致するような範囲でそれぞれを設定し、データ転送はその一致する時間毎に行なう (Fig.5.4)。これは、それぞれの最適な時間刻みの最小公倍数をとるという意味ではなく、1：2とか2：3のように最適時間刻みの中で簡単な整数比となるように調整して設定するということである。いくつの時間刻み同士で対応させるかというのは並列処理時の負荷バランスに関わる問題であるが、それについては次の節で述べる。

5.3 並列化手法

初めの節でも述べたように、ここでは各プロセスを並列に実行するアプローチをとる。流体解析の章にあった移動格子の例と異なり、この連成解析においては流体の格子生成とその解析プロセスは、格子形状がもうひとつの並列プロセスである構造解析により決定されるために並列に実行することは出来ず、逐次的に扱うことになる。従ってここで並列に実行することになるのは格子生成も含めた流体のプロセスと構造のプロセスということになる (Fig.5.5)。

それぞれのプロセスはMPMD(Multiple Program Multiple Data) と呼ばれる方法で並列実行される。すなわち、流体と構造解析はそれぞれの独立したプログラムとしてそれぞれ異なるCPUの上で同時に実行される。ところで、本研究においては並列実行というのは異なる解析プログラム間の並列だけではなく、各プログラム内においても領域分割により行なわれる。従って、それぞれの解析プログラムが複数のCPUに分割され自分の担当の計算領域を担当することになる。単一の解析プログラムの並列計算と同様に、ここでも各CPU上の計算量、計算時間のバランスが問題となるが、異なる解析系間のバランスをとることになるので単一のプログラムのみを扱う場合より多少複雑になる。問題を考えるために、まず流体解析 (格子生成) と構造解析それぞれが実行さ

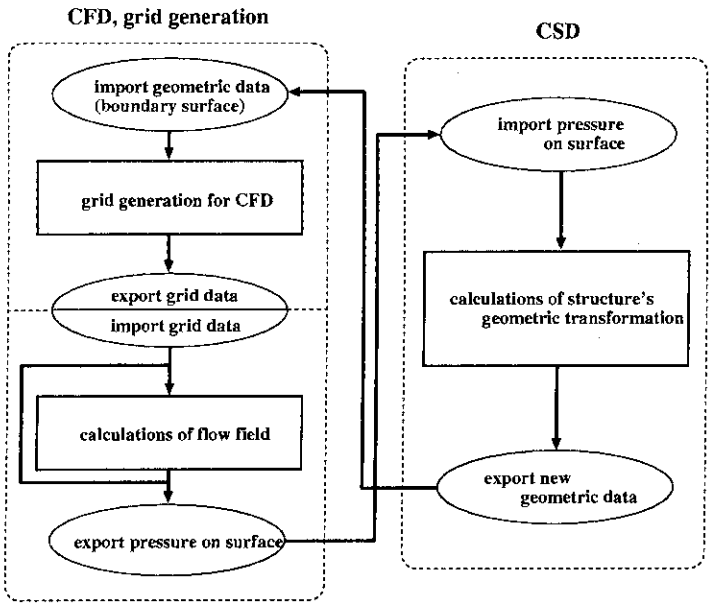


Fig.5.5 並列処理の流れ

れる CPU 数をそれぞれ $\boldsymbol{NCPU}(cfd)$ と $\boldsymbol{NCPU}(csd)$ とする。また、それぞれの解析系で 1 計算ステップに必要な計算時間を $\boldsymbol{TIME}$、データの受渡しが行なわれる時間単位内に実行される計算ステップ数を $\boldsymbol{NLOOP}$で同様に表現すると、計算量のバランスをとる式は次のように書くことが出来る。

$$\frac{TIME_{(grid)} + TIME_{(cfd)} \cdot NLOOP_{(cfd)}}{NCPU_{(cfd)}} = \frac{TIME_{(csd)} \cdot NLOOP_{(csd)}}{NCPU_{(csd)}} \tag{5.1}$$

ここで計算時間のなかには CPU 間のデータ通信にかかる時間も含むものとする。このようにして考えると、バランスをとるためにはそれぞれのプロセスをいくつの CPU で並列実行するかということと、いくつの時間ステップ数をひとつの単位として考えるかということの組み合わせで決定出来る。バランスのとり方の例を Fig.5.6 に示す。ここでは各プロセス内での並列化のバランスはとれていることを仮定して、プロセス間でのバランスだけを問題としている。

Fig.5.6 の上の図は流体と構造解析のプロセス間のバランスがとれずに、構造解析の方でデータ転送を行なう前に待ちの時間が発生してしまう例を示している。それに対して下の左側の図で示しているのは構造解析側でもう 1 時間ステップ単位時間内に行なう計算を増やしてバランスをとっているものである。こうすることによって確かに無駄な待ち時間というものは少なくすることが可能であるが、ちょっと考えてみると例えば流体の方はその計算手法による安定性などから決まる時間刻みの大きさが変えられないとすると、結局この時間単位の間に進めることの出来る物理的な時間と、それにかかる計算時間は変わらない事になる。データ転送時の物理的な時間を合わせなくてはならないために、構造解析の方のステップ数を増やしたところで、その時間刻みを小さくして合計の時間と同じにしなくてはならないからである。これでは単に待ちの時間を無くした

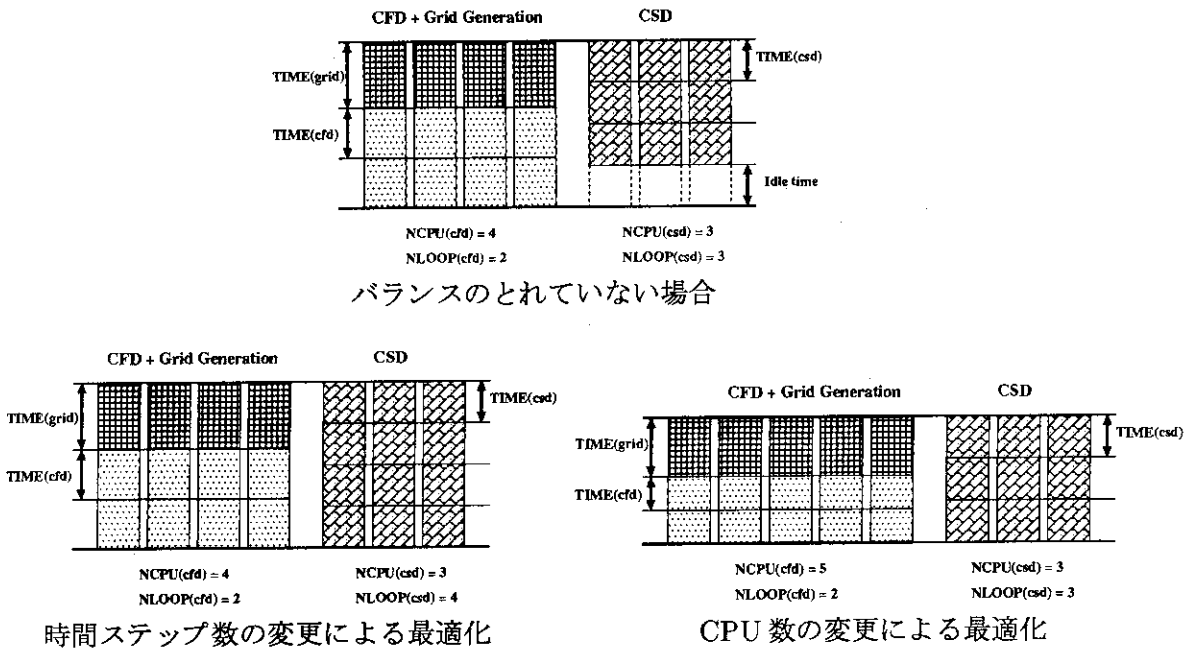


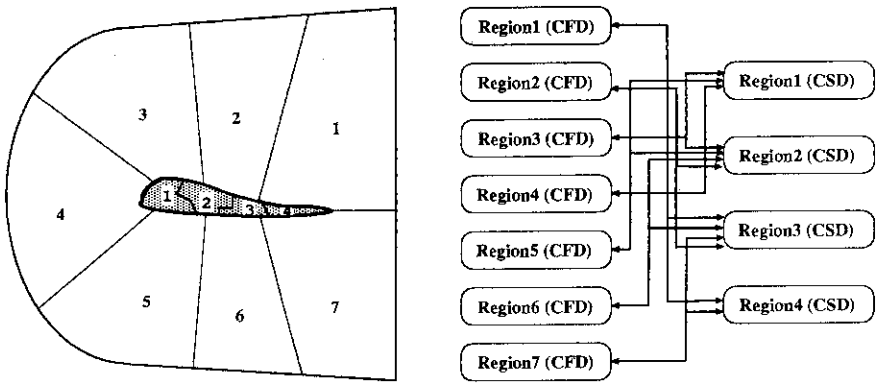
Fig.5.6 並列実行における負荷分散

というだけのことになってしまう。それに対して、右の図は流体のプロセスの分割数を増やす事によりバランスをとっている例を示している。この方法では流体解析の並列実行数を多くすることで、計算時間が短い構造解析のものと合わせている。これによると同じ物理時間の計算を行なうのにかかる処理時間が短くすることが出来る。実際には使用できる CPU の数と扱う問題のそれぞれの解析における問題のサイズなどによって上のような観点から最適のものを見つけることになる。

5.4 データ転送 (インターフェース)

先に述べたように、一定時間経過毎にお互いの計算データを相手の計算条件として渡し合うことを行なう。このとき、それぞれのプロセスが並列処理していることからデータを受け渡す CPU がどれとどれが対応しているかということを知っておかなければならない。流体解析と構造解析においてはそれぞれ領域を分割することにより並列化を達成する。分割された領域間でデータを受け渡すものが一体一に対応していれば簡単であるが、それぞれが扱う問題のサイズが異なり並列処理の分割数が異なる場合が一般的である。たとえ分割数が同じであってもそれぞれの計算対象領域は幾何学的に同じような範囲にひろがっていない場合には一体一に対応するようにそれぞれを分割することは困難である。

Fig.5.7 は 2 次元でそのような領域間の対応を図示したものである。全体の領域の中程に位置している翼状の領域が構造解析の計算領域であり、それと外周の間を埋めるのが流体解析の計算領域となる。この図ではそれぞれ 4 と 7 に分割している。データ転送の対応は左側に示しているが、例えば流体の領域 1 を担当する CPU は構造の領域 3 と 4 を扱う CPU とそれぞれデータの受渡し



各計算領域の分割　　データ転送の対応

Fig.5.7 分割領域間の対応

を行なわなければならない。また、構造の領域１についてみると流体の領域３、４、５との連絡が必要である。これらの対応は計算を始める前に全部の CPU において情報を持つようにしておかねばならない。具体的には、格子生成プロセスが構造のデータを構造解析でどのように分割するかの情報も含めてもらい、流体の格子を生成するときに対応表を作って流体、構造解析に渡すようにすればよいだろう。

5.5　結果の可視化

数値計算結果の可視化 (visualization) は、数値計算の発達と共に近年重要性を増しており、可視化の手法だけでも重要な研究分野になっている。本研究においても計算結果の可視化というのは非常に重要な要素である。なぜならば、流体と構造の連成問題では、流体と物体とがそれぞれ非定常かつ非線形に作用し合う現象であり、流れ場の状態、物体の形状や位置の変化、流体から受ける物体表面の圧力分布、等、数値データだけでは結果の把握は不可能であるからである。

したがって、本研究においても計算結果を何らかの形で可視化を試みる。数値計算自体は並列計算機上において行なわれ、流れ場の物理量、物体の形状データは計算終了後もしくは計算途中において PLOT3d 等の形式でデータファイルの形で磁気ディスク上に格納される。結果の可視化、すなわち等密度図、等圧力図、速度ベクトル図等は、画像処理用の高速ワークステーションからそれらのデータをアクセスし、AVS 等の画像処理ソフトを用いて行なわれる。また、扱われる現象は非定常現象であり、時間変化を直接イメージとして捉えることも重要である。したがって、一定の時間間隔でデータを出力し時系列データとしてディスク上に蓄積し、後にワークステーションでアニメーション化を行なう。

現時点では、結果の可視化は上記のような方法で計算の後処理として行なわれるが (Fig.5.8、上)、将来的には、後処理も並列計算機上で並列に高速に行ない、画像データをワークステーションに送り実時間で結果をアニメーションで出力する (Fig.5.8、下) ことが考えられる。

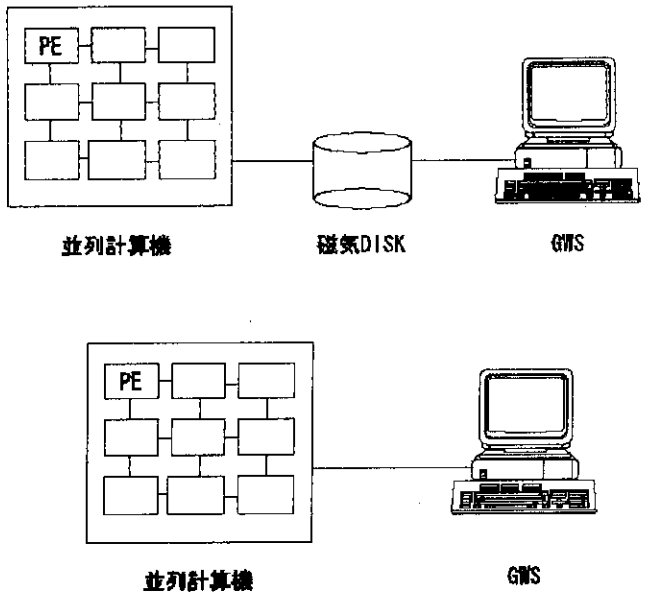


Fig.5.8 可視化システムの構成

参 考 文 献

1) Man Liu and D.G.Gorman, *'Symmetrization of Equations of Motion for Coupled Systems Subject to Fluid-Structural Interactions'*, Communications in Numerical Methods in Engineering, Vol.11, 831-838 (1995)

2) F.Belanger, et.al., *'Dynamics of Coaxial Cylinders in Laminar Annular Flow By Simultaneous Integration of the Navier-Stokes and Structural Equations'* Jounal of Fluids and Structures, Vol.8, 747-770 (1994)

3) G.P.Guruswamy and C.Byun, *'Fluid-Structural Interactions Using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures'*, AIAA Paper, 93-3087 (1993)

4) C.Byun and G.P.Guruswamy, *'Wing-Body Aeroelasticity Using Finite-Difference Fluid/Finite-Element Structural Equations on Parallel Computers'*, AIAA-94-1487-CP (1994)

5) R.D.Rausch, *'Three-Dimensional Time-Marching Aeroelastic Analyses Using an Unstructured-Grid Euler Method'*, AIAA Journal, Vol.31, No.9, 1626-1633 (1993)

6. 制御システムとの統合概念

前章までに空力と構造の連成方法を述べたが、本章では更に制御システムとの結合を図りシミュレーションを行う案を示す。

1950年代にジェット旅客機が出現して以来、航空機の制御システムは飛行経路の安定化、自動操縦、空力荷重や運動荷重の軽減、省エネルギー、機動性向上などの目的で発展し、近年は概念設計の段階から空力や構造と同等に取り扱われる様になった。高速大型機の代表とも言える超音速旅客機では、静安定緩和(Relaxed Static Stability:RSS)や突風荷重軽減(Gust Load Alleviation:GLA)など、多くの先進制御技術の取り込みが計画されている。RSSはもともと軍用機の機動性向上の為に開発された技術であるが、民間機では舵面の釣り合い面積を減らし、抵抗と重量の低減を図る目的で適用される。舵面面積の減少により安定性が損なわれるが、これを補うために安定増大装置(Stability Augmentation System)を組み込んだ制御系が導入される。また、GLAは主に長大な主翼を持つ大型輸送機に適用される制御システムで、突風による揚力増大を適切な操舵によって抑え、構造材にかかる荷重を減らし軽量化を図る事を目的としている。

空力と構造に加え制御システムとも結合する事により、上に述べた各種制御系を含んだ統合シミュレーションを行う事が期待出来る。現時点では詳細仕様を出すまでには至らないが、以下に基本方針を示す。

(1) 制御システムの開発

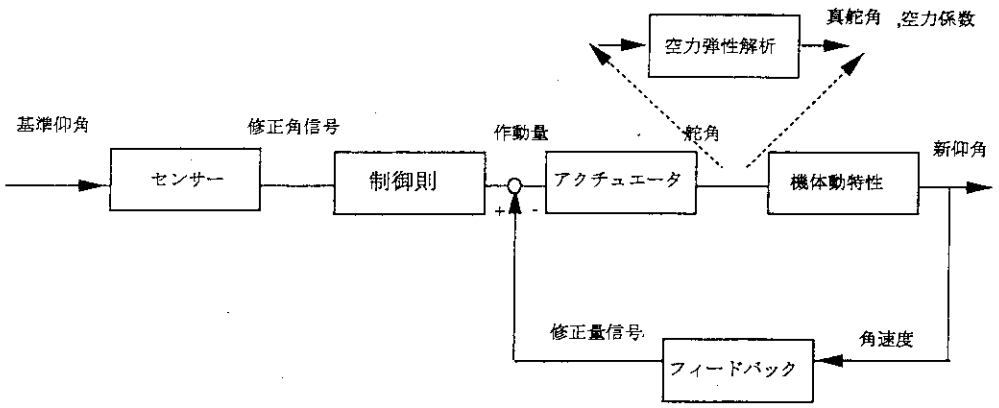


Fig.6.1 ピッチ制御の概念

Fig.6.1はピッチダンパ制御システムと空力弾性解析との結合概念で、センサー(通常はレートジャイロ)が基準仰角とのずれを検出し、ある制御則でアクチュエータの作動量を計算し、修正のための水平尾翼の操舵を行う。それを条件として機体の運動方程式を解き修正後の仰角を算出する一連のブロックから成っている。修正後の仰角にフィードバックゲインがかけられ、基準仰角に安定するまで繰り返えされる。各ブロックにはセンサーのゲイン、制御則

やアクチュータ特性を示す伝達関数が定義され、これを解いて時間応答などを見る事が出来る。実際には複数の舵面との連成（Cross Coupling）などにより、ブロック線図は非常に複雑なものとなる。

この様な制御システムを解析する事により、機体の弾性変形を考慮した飛行特性や、形状や材質の相違による運動の変化を見る事が出来る。例えば、後退角を変えた時の安定性の変化、舵面面積を小さくした際の安定性の変化と空力抵抗の低減、突風などの外乱を入力し主翼の変形や応力状態を見ながら機体の時間応答を見るなど、部分的な形態の変更が飛行性能に及ぼす影響を観察する事が可能となる。 空力弾性計算コードは、アクチュータからの舵角変化を受け、流れ場の解析と構造解析を行い揚力、抗力、モーメント、構造変形による真の舵角、などを算出し機体運動計算ブロックに入力する。センサーやフィードバックのゲイン、制御則やアクチュータ特性の伝達関数、操舵のカップリングなどは適用機種により構成が大きく異なる。例えば、大型旅客機と小型軍用機では舵面の構成が大きく異なり、かつ正反対の応答特性（安定性に対し高機動性）が要求される。この為、適用機種に合わせて制御システムの構成を変更可能としておく事が考えられる。開発は、まず最小構成の制御システム、例えばピッチ制御のみを先に実現し、これを適用機種に応じて拡張する方針で進める事が考えられる。プログラムの概要を Fig.6.2 に示す。

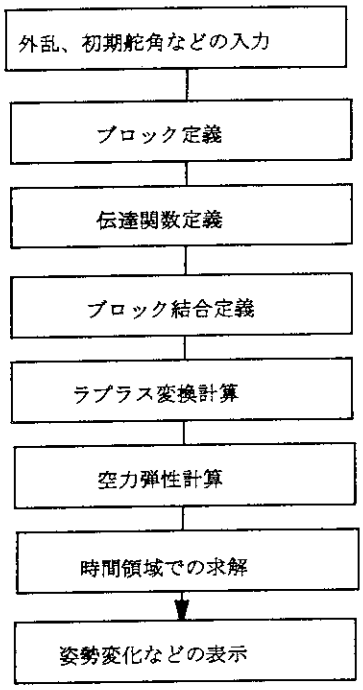


Fig.6.2 制御プログラムの概要

Fig.6.2 は Fig.6.1 の様なブロックを定義し、舵角や突風条件などを与え、姿勢の時間応答を

見るプログラムの流れを示している。全体の構成は、各種の制御装置や機体などのブロック、それらの伝達関数、ブロック間の結合、などを機種に合わせて定義する部分と、ラプラス変換計算を行う実行部分から成る。定義部は、適用機種に応じてブロックや結合の追加変更が出来るプログラム構造としておく。

(2) 連成解析コードとのインタフェース

機体の姿勢は、舵角を機体の動特性式に与えて計算する。動特性式は揚力、抗力、モーメントの釣り合いから導かれる式で、通常は実験的に得られた翼型空力係数などをもとに、機体を剛体として作られる。ここではCFDによる流れ場の圧力データをもとに各種係数を作成し、動特性式を作る。これにより、旋回などで受ける運動荷重(G)や空力により構造が変形した場合の効果も含んだ動特性式を得る事が出来る。また、弾性変形を考慮した真の舵角を求めたり、弾性変形による舵の逆利き現象を模擬する事も考えられる。

(3) マルチブロック・オーバラップ格子生成

空力舵面の操作を模擬するには、流れ場の中で舵面を独立して動かす事が出来なければならない。大型旅客機の空力舵面は複雑な形状の物が多く、これを一つの構造格子でモデル化する事は非常に難しいと思われる。特に多重隙間フラップやスポイラなどは、操作時には流れ場のトポロジーが変化するため、連続して格子変形をさせる事は極めて困難と思われる。このため、主翼、胴体、舵面などを、境界で重なりを持つ独立した構造格子のブロックで作成する事を考慮している。各々の格子ブロックはコンポーネント毎に独立して動きを与える事が出来る。また、本研究中で提案している変形格子を使う事により、各コンポーネントの独立した弾性変形も加える事が可能になる。例えば、空力振動を伴うエルロンの操舵、などが可能となる。マルチブロック・オーバラップ格子(複合格子やキメラメッシュなどとも呼ばれる)は、既に多くの文献が出されている[1)2)]。これらの中には、航空機からの兵装セパレーションなど、動きを伴うものもある[3)4)]。

ヘリコプターの回転翼やVTOL機の排気ジェットなどにも、適用可能と思われる。

参考文献

1) 高倉葉子、小川哲、"遷音速風洞内全機模型流れのベクトル・パラレル・シミュレーション",航空宇宙技術研究所特別資料、第12回航空機計算空気力学シンポジウム論文集,1994年12月

2) 飯田雅宣、吉田康夫、前田達夫、小川哲、和田安弘,"高速鉄道に係わる空力問題の数値シミュレーション(その2)",航空宇宙技術研究所特別資料、第12回航空機計算空気力学シンポジウム論文集,1994年12月

3) Christopher,A. A., "Computation of a Controled Store Separation from a Cavity",J. of Aircraft,Vol. 32, NO. 4,1995,pp. 846-852.

4) Belk,D. M., "Automated Assembly of Structured Grids for Moving Body Problems",AIAA-95-1680-CP

7. 今後の進め方

当面の目標は、本文中で述べた様に流体（空力）解析と構造解析の連成解析を行うコードを開発する事である。具体的には、3次元動的変形格子生成プログラム、3次元非定常圧縮性流体コード、3次元有限要素構造解析コード、並列処理インタフェース、可視化プログラムを開発する。ターゲットマシンはIBM-SP2とし、MPMD型の並列処理を行う。流体解析は、既に開発した境界移動を伴う2次元非定常圧縮性コードの3次元化とMPIによる並列化を行う。構造解析は、現時点で交渉中であるが理化学研究所より提供の3次元非線形動解析コードの変更とMPI並列化を行う予定である。可視化は、当センターの画像サーバに置かれたAVSに入力する事により行う。また、将来は当センターで進められている実時間可視化システムとの結合も検討している。3次元動的変形格子生成プログラムと並列処理インタフェースは、既存の物が無いため新たに開発する。

謝　　辞

本研究にあたり、科学技術庁 航空宇宙技術研究所 数理解析部 応用解析研究室 岩宮敏幸室長及び青山剛史研究員、並びに理化学研究所 素形材工学研究室室長牧野内昭武主任研究員及び郭智宏研究員からは、コードやデータの提供などで多大な協力を受けており、ここに感謝の意を表する次第である。

7. 今後の進め方

当面の目標は、本文中で述べた様に流体（空力）解析と構造解析の連成解析を行うコードを開発する事である。具体的には、3次元動的変形格子生成プログラム、3次元非定常圧縮性流体コード、3次元有限要素構造解析コード、並列処理インタフェース、可視化プログラムを開発する。ターゲットマシンはIBM-SP2とし、MPMD型の並列処理を行う。流体解析は、既に開発した境界移動を伴う2次元非定常圧縮性コードの3次元化とMPIによる並列化を行う。構造解析は、現時点で交渉中であるが理化学研究所より提供の3次元非線形動解析コードの変更とMPI並列化を行う予定である。可視化は、当センターの画像サーバに置かれたAVSに入力する事により行う。また、将来は当センターで進められている実時間可視化システムとの結合も検討している。3次元動的変形格子生成プログラムと並列処理インタフェースは、既存の物が無いため新たに開発する。

謝　　辞

本研究にあたり、科学技術庁 航空宇宙技術研究所 数理解析部 応用解析研究室 岩宮敏幸室長及び青山剛史研究員、並びに理化学研究所 素形材工学研究室室長牧野内昭武主任研究員及び郭智宏研究員からは、コードやデータの提供などで多大な協力を受けており、ここに感謝の意を表する次第である。

APPENDIX A

(1) 剛性マトリックス

$$
\begin{aligned}
[\boldsymbol{k}] &= det\,|\boldsymbol{J}| \sum_{i=1}^{4}\sum_{j=1}^{4} H_1 H_2\,[\boldsymbol{B}]^T\,[\boldsymbol{D}]\,[\boldsymbol{B}] \\
[\boldsymbol{B}] &= [\boldsymbol{A}]\,[\boldsymbol{N}] \\
[\boldsymbol{A}] &= \begin{bmatrix} \frac{\partial}{\partial \mathrm{x}} & 0 & 0 \\ 0 & \frac{\partial}{\partial \mathrm{y}} & 0 \\ \frac{\partial}{\partial \mathrm{y}} & \frac{\partial}{\partial \mathrm{x}} & 0 \\ 0 & \frac{\partial}{\partial \mathrm{z}} & \frac{\partial}{\partial \mathrm{y}} \\ \frac{\partial}{\partial \mathrm{z}} & 0 & \frac{\partial}{\partial \mathrm{x}} \end{bmatrix}
\end{aligned}
$$

(2) 形状関数マトリックス

$$
\begin{aligned}
[\boldsymbol{N}] &= \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \\
N_1 &= (1-\xi)(1-\eta)/4 \\
N_2 &= (1-\xi)(1+\eta)/4 \\
N_3 &= (1+\xi)(1+\eta)/4 \\
N_4 &= (1+\xi)(1-\eta)/4 \\
[\boldsymbol{J}] &= \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \\
x_i, y_i &: \text{要素節点の座標} \\
[\boldsymbol{D}] &= \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 & 0 & 0 \\ \nu & 1 & 0 & 0 & 0 \\ 0 & 0 & (1-\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & (1-\nu)/2k & 0 \\ 0 & 0 & 0 & 0 & (1-\nu)/2k \end{bmatrix} \\
k &= 1.2
\end{aligned}
$$

(3) 座標変換マトリックス

$$
[T] = \begin{bmatrix} [L] & 0 & 0 & 0 & 0 & 0 \\ 0 & [L] & 0 & 0 & 0 & 0 \\ 0 & 0 & [L] & 0 & 0 & 0 \\ 0 & 0 & 0 & [L] & 0 & 0 \\ 0 & 0 & 0 & 0 & [L] & 0 \\ 0 & 0 & 0 & 0 & 0 & [L] \end{bmatrix}
$$

$$
[L] = [\{l_x\}\,\{l_y\}\,\{l_z\}]
$$

$\{l_x\}$、$\{l_y\}$、$\{l_z\}$は要素座標軸を決めるベクトルで、$\{l_x\}$は第1節点と第2節点が作る方向ベクトル、$\{l_z\}$は第3節点と第2節点が作る方向ベクトル、$\{l_y\}$ は l_xと l_zの外積ベクトルである。

APPENDIX B

(1) 多層複合材の [D] マトリックス

$$[\boldsymbol{D}] = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix}$$

$$\begin{aligned} A_{ij} &= \sum_{k=1}^{n} Q_{ij}(\theta_k)(z_{k+1} - z_k) \\ B_{ij} &= \sum_{k=1}^{n} Q_{ij}(\theta_k)(z_{k+1}^2 - z_k^2)/2 \\ D_{ij} &= \sum_{k=1}^{n} Q_{ij}(\theta_k)(z_{k+1}^3 - z_k^3)/3 \end{aligned}$$

$$\begin{bmatrix} Q_{11}(\theta_k) \\ Q_{12}(\theta_k) \\ Q_{22}(\theta_k) \\ Q_{16}(\theta_k) \\ Q_{26}(\theta_k) \\ Q_{66}(\theta_k) \end{bmatrix} = \begin{bmatrix} c^4 & 2c^2s^2 & s^4 & 4c^2s^2 \\ c^2s^2 & c^4+s^4 & c^2s^2 & -4c^2s^2 \\ s^4 & 2c^2s^2 & c^4 & 4c^2s^2 \\ c^3s & -cs(c^2-s^2) & -cs^3 & -2cs(c^2-s^2) \\ cs^3 & cs(c^2-s^2) & -c^3s & 2cs(c^2-s^2) \\ c^2s^2 & -2c^2s^2 & c^2s^2 & (c^2-s^2)^2 \end{bmatrix} \times \begin{bmatrix} Q_{11}(0) \\ Q_{12}(0) \\ Q_{22}(0) \\ Q_{66}(0) \end{bmatrix}$$

$$\begin{aligned} c &= \cos(\theta_k) \\ s &= \sin(\theta_k) \end{aligned}$$

$$\begin{aligned} Q_{11}(0) &= \frac{E_1}{(1-\nu_{12}\nu_{21})} \\ Q_{12}(0) &= \frac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})} \\ Q_{22}(0) &= \frac{E_2}{(1-\nu_{12}\nu_{21})} \\ Q_{66}(0) &= G_{12} \end{aligned}$$