JAXA-RM-09-003

宇宙航空研究開発機構研究開発資料

JAXA Research and Development Memorandum

超音速機形態のフラップ舵角変更に対する自動格子生成ツール AutoFlap-GG

Automatic Grid Generator for Flap Deflection of Supersonic Transport Configuration — AutoFlap-GG —

永田　靖典*[1]，雷　　忠*[2]

Yasunori NAGATA*[1] and Zhong LEI*[2]

＊１株式会社　菱友システムズ
Ryoyu Systems Co.,Ltd.

＊２航空プログラムグループ　超音速機チーム
Supersonic Transport Team, Aviation Program Group
現在，諏訪東京理科大学　システム工学部　機械システム工学科

２００９年９月
September 2009

宇宙航空研究開発機構
Japan Aerospace Exploration Agency

超音速機形態のフラップ舵角変更に対する自動格子生成ツール AutoFlap-GG*

永田　靖典[*1]，雷　　忠[*2]

Automatic Grid Generator for Flap Deflection of Supersonic Transport Configuration — AutoFlap-GG —*

Yasunori NAGATA[*1] and Zhong LEI[*2]

Abstract

An automatic grid generator was developed to modify the surface shape and computational mesh automatically for a supersonic transport configuration as the leading- and trailing-edge flaps were deflected. In the design of high-lift devices with CFD technology, a large number of geometries with different combination of design parameters are required for parametric study or shape optimization to improve aerodynamic performance. Hence, it is necessary to generate the shape and grid of each case. With the automatic tool, time cost for shape modification and grid generation is dramatically reduced. In this AutoFlap-GG, the unique script language Glyph included in the grid generation software, Gridgen, is used to automate the process of shape modification and grid generation. In this report, the automatic process of shape modification and grid generation are described, and examples of the generated grid and computational result are presented.

Keywords: SST, High-Lift Device, Automatic Grid Generation, CFD

概　　要

現在，JAXA超音速機チームでは超音速航空機（SuperSonic Transport, SST）の高揚力装置の最適設計に関する研究が行われている．高揚力装置は，離着陸の際の低速時において揚力を増加させるためのものであり，翼前・後縁の一部を操舵するようなフラップが多く使用される．高揚力装置の性能を検証するために，実験，数値解析を通して研究が進められている．

CFD解析を用いて最適設計を行うことで，より高いレベルの最適化を実現できると考えられるが，設計パラメータに応じて多数ケースの形態について解析する必要がある．形態変更には，形状変更や計算格子生成などの前処理が伴い，多大な労力を要する．設計期間を短縮し，コストを削減するためには，計算時間の短縮とともに形状作成や格子生成の自動化が必要となる．

本報告では，高揚力装置として前・後縁フラップを採用した超音速機高揚力形態について，フラップ操舵前の形状・格子に対して修正を加えることで，フラップ操舵後の形状と計算格子を自動生成するツールAutoFlap-GG（Automatic deflecting Flap-Grid Generator）について述べる．市販格子生成ソフトGridgen用のスクリプト言語Glyphを用いて，形状・格子修正に伴う作業を自動化することで，フラップ操舵後の格子を短時間で得られるようになった．

キーワード：SST，高揚力装置，自動格子生成，CFD

---

*　平成21年7月15日受付 (received 15 July 2009)

*1　株式会社 菱友システムズ (Ryoyu Systems Co.,Ltd.)

*2　航空プログラムグループ　超音速機チーム (Supersonic Transport Team, Aviation Program Group)

現在，諏訪東京理科大学　システム工学部　機械システム工学科

1．はじめに

近年，計算機性能と計算手法の飛躍的な進捗により数値流体（Computational Fluid Dynamics, CFD）解析が多くの設計に適用され，実用的になりつつある．非線形性を織り込んだCFD解析によって，より高いレベルの設計を実現できると考えられる．この手法は従来の風洞試験を中心とする手法に対して時間，経費を大幅に短縮，削減することが可能な技術である．

一方，CFDを用いた設計には設計パラメータ数と設計手法によって，数十から数百ケースの形態を解析することが必要であり，形態変更の際に形状修正や，計算格子生成などの前処理に多大な労力を要する．例えば，航空機のような複雑な形態の場合では，この作業に非常に時間と手間が掛かるため，CFDを適用する実機設計にとって，大きな障害になっている．設計期間を短縮し，コストを低減するためには，より高性能な計算機と計算手法の改善が要求され，それと同時に形状作成と格子生成を自動化することが必要となる．

宇宙航空研究開発機構航空プログラムグループは次世代超音速旅客機の研究開発を進めてきた[1]．翼が巡航時の高速に合わせて設計されるため，低速の離着陸時において揚力が不足してしまう．超音速機の離着陸性能を改善するために, 高揚力装置は不可欠である．本研究では，比較的簡単な機構により実現できる前・後縁フラップを採用した．この前縁フラップは，翼前縁の一部を下方へ折り曲げることによって，翼前縁から剥離渦の形成を抑制する装置である[2]．翼に働く抵抗力を低減させ，揚抗比が改善される. 後縁フラップは, 内翼後縁付近でヒンジ・ラインを軸にして翼の一部だけを下方へ折り曲げることで揚力を増加させ，離着陸時に必要な揚力を得る装置である．フラップによる空力性能向上の効果はフラップの形状と折り曲げる舵角に大きく依存する．

CFDを用いた最適化設計システムは，図1に示すように, 前処理（形状生成と計算格子生成）と性能評価（CFD解析），最適化プロセスにより構成される．フラップ効果を最大限に得るためには，設計パラメータを組み合わせた多くの解析が必要となる．パラメータを変更すると，機体形状とCFD計算に必要な計算格子の修正を行わなければならない．多数の形状について計算を行う必要があるため，従来の手作業による格子生成では非常に多くの時間が割かれることになる．

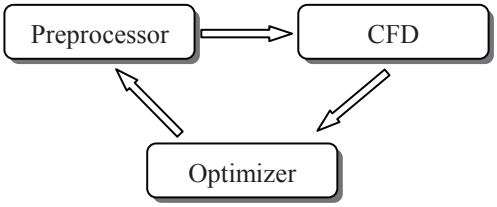


図1　最適化設計ループ

本研究では，形状変更と格子生成に要する時間の短縮および手作業の労力削減のために，主翼の前・後縁にフラップを有する超音速機形態まわりの自動形状変更・格子生成ツールを開発することを目的とする．米国POINTWISE社が開発したGridgen [3]を利用して，超音速機高揚力形態についてフラップを操舵させた際の形状と計算格子を自動修正する手法を述べる. なお, 本報告では, 本ツールを用いて修正前の形状・格子に対して修正を施し, フラップ操舵後の形状･格子に作り変えることを形状･格子生成と呼ぶこととする．

2．AutoFlap-GG解説

2.1　概要

AutoFlap-GG（Automatic deflecting Flap-Grid Generator）は，フラップを有する超音速機高揚力形態について，フラップ操舵後の形状と計算格子の自動生成を行うツールである．格子はマルチ･ブロック構造格子を対象とする．本ツールは，米国POINTWISE社が開発した市販格子生成ソフトGridgen V15.10を用いて，あらかじめ作成されたフラップ操舵前の形態（基本形態）の形状・格子に対して，形状と格子の修正を行うことでフラップ操舵後の形状・格子を自動生成する．フラップ操舵に伴う形状・格子の修正についてはGridgen独自のスクリプト言語であるGlyphを用いて作業を自動化した．Gridgenは格子生成が主な機能であるが，形状データを編集する機能もいくつか搭載しており, それらを用いて形状修正を行っている．出力される格子データは，並列CFD解析ソルバーにそのまま入力できるようにブロックを適当に分割した状態で出力される．

本ツールはGUI（Graphical User Interface）を実装しており，GUIを用いた対話的な実行が可能である．GlyphはTcl/Tkをベースとして，Gridgen操作用のコマンドを加えたスクリプト言語である．Tcl/Tkは，スクリプト言語Tcl（Tool Command Language）とそのGUI作成ツールTk（Tool Kit）から成る．Tclは，アプリケーションに組み込むための拡張用スクリプト言語として開発された言語であり，インタプリタ言語，リスト処理，クロス・プラットフォームなどの特徴がある．Tkは，Tcl用に開発されたGUI作成ツールキットであるが，PerlやPythonといった他の言語でも標準的なGUIキットとして利用されている．GlyphでもTkを用いてGUIを作成することができ，これを用いて本ツールにGUIを実装させた．GUI画面にパラメータや出力ファイル名を入力することで形状・格子生成を行うことができる．また，GUIを用いずにバッチ処理で実行することも可能となっている．GUIを用いる場合，ユ

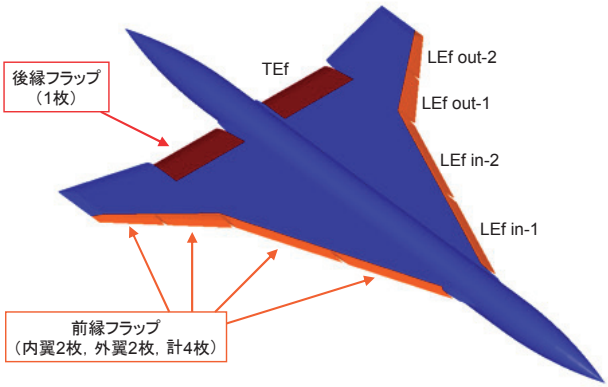


図2　JAXAジェット実験機01次形状高揚力形態

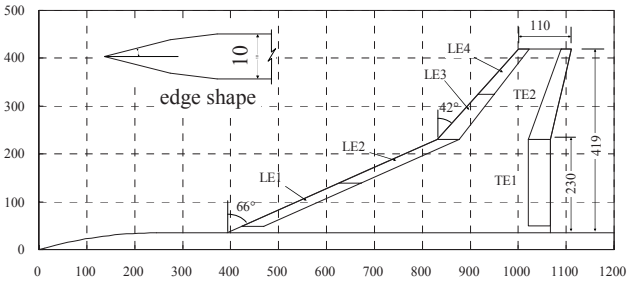


図3　平面形状（単位：mm）

ーザが手入力でパラメータを入力するため，多数ケースの処理には向かない．GUIを用いないバッチ処理機能を用意することで，多数ケースの処理に対応でき，そのまま最適化設計ループに組み込むこともできる．

本ツールでは制限事項として，各フラップの舵角の対応範囲を設けている．これは，対象としている機体形状の場合，舵角の組み合わせによってはフラップとフラップが干渉し，形状として成り立たない場合があるためである．また，形状として成り立ったとしても，トポロジーの関係から格子が大きく歪む場合があるためである．対応舵角範囲をさらに広げるためには，自動処理の空間格子修正時に使用されるパラメータを調節する必要があり，コードを書き換えなければならない．ただし，この範囲を超えた舵角に対しても適切に格子が生成される場合もある．

本ツールを実行させるにはVersion 15.10以降のGridgenがインストールされている必要がある．これは，形状・格子生成の元データであるフラップ操舵前の形状・格子データ（基本形態形状・格子データ）がGridgen V15.10を用いて作成されているためである．Gridgenでは使用しているソフトのVersionよりも新しいVersionで作成されたデータを読み込むことができない．すなわち，Gridgen V15.10より古いVersionのGridgenでは，基本形態データを読み込むことはできない．OSについてはWindows, UNIX/Linuxともに動作し，それぞれのOSの起動スクリプトを用意している．

2.2　対象機体形状

図2に本ツールの対象である超音速機の高揚力形態を，図3にその平面形状を示す．以降，この形状についてフラップ操舵していない状態を基本形態と呼ぶ．基本形態の平面形は，宇宙航空研究開発機構航空プログラムグループで進められた小型超音速ジェット実験機の第01次形状として採用された機体の翼胴形態を8%に縮小したものである．寸法諸元は，全長1.36 [m]，スパン長 b =0.419 [m]×2，翼面積S_w=0.292 [m^2]，主翼アスペクト比AR=2.42，空力平均翼弦長（Mean Aerodynamic Chord, MAC）0.459 [m]である．基本形態の主翼は厚さ30 [mm]を持つクランクド・アロー平板翼であり，前・後縁および翼端を30 [deg.]頂角で尖らせたものである．主翼の平面形には，超音速巡航時（M=1.7）で設計した結果としてArrow型が採用され，内翼が後退角λ_{inner}=66[deg.]を持つ亜音速前縁であり，低速と遷音速性能の改善を考慮して外翼が後退角λ_{outer}=42[deg.]を持つ形状である．機体軸から翼端方向に半スパンの55%位置をキンクとして内翼と外翼がつながる．

高揚力装置には様々なものが考案されているが，ここでは効果的，かつ実用性の高い前縁フラップと後縁フラップを想定している．各フラップはフラップの平面形状を固定し，単にヒンジ・ラインまわりに下方に折り曲げる簡素な構造である．内翼前縁，外翼前縁にそれぞれ2枚，後縁に1枚，計5枚のフラップを取り付ける．内翼と外翼の前縁フラップはセグメント2枚ずつで均等に分割される．それぞれのフラップセグメントに個別に舵角を設定する．内翼前縁フラップの弦長はスパン方向全ての翼断面において機軸方向で10%MAC，ヒンジ・ライン（回転軸）は前縁に対して平行し，後退角は前縁と同じ66[deg.]をとっている．外翼前縁フラップの弦長は各翼断面において局所翼弦長の20%，ヒンジ・ラインは前縁と異なる後退角37.7[deg.]をとっている．前縁フラップの舵角は前縁を下方へ操舵する場合，正とする．後縁フラップは内翼のみに取り付け，弦長が10%MACを持つ．後縁フラップの舵角は後縁を下方へ操舵する場合，正とする．フラップ間には操舵時に互いに重なり合うのを避けるため，5[mm]の隙間を設けてある．

2.3　Gridgen解説

Gridgenは米国POINTWISE社が開発した3次元格子生成ソフトウェアである．1984年以来，航空宇宙分野のCFD解析用格子をはじめ，機械，環境，医療など多くの分野で活用されている．Gridgenを用いることで高品質の格子を生成でき，複雑形状にも柔軟に対応することがで

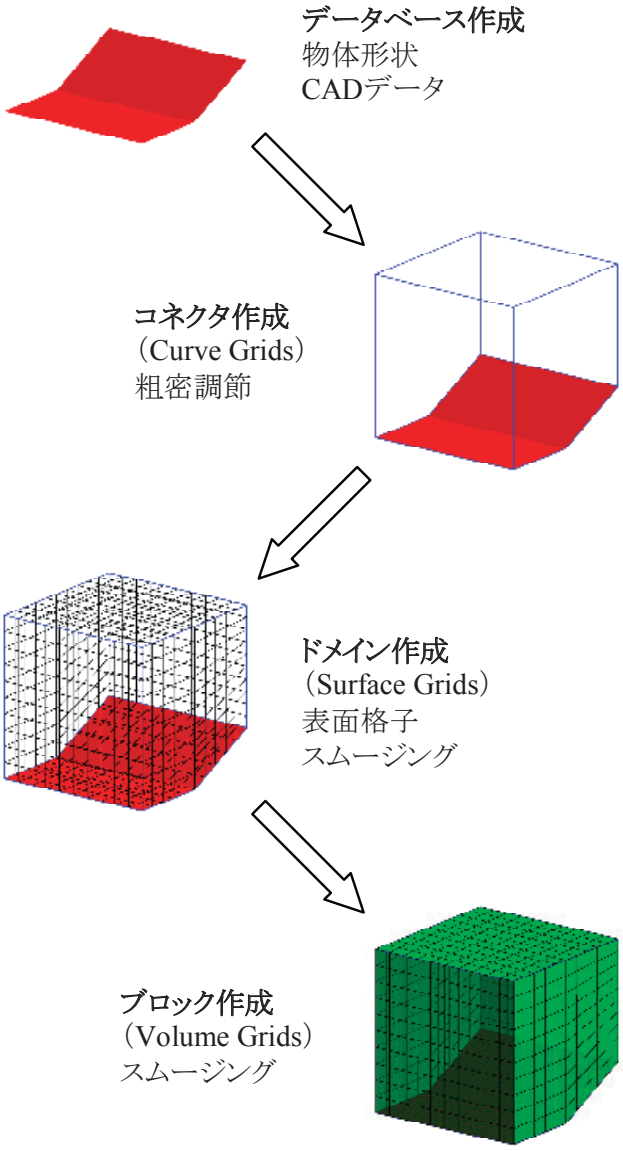


図4　Gridgenにおける格子生成
（ボトムアップ・アプローチ）

きる．

Gridgenは構造格子，非構造格子，ハイブリッド格子を生成することが可能である．表面・空間格子の生成には代数型格子生成法（TransFinite Interpolation, TFI）が使用され，楕円型格子生成ソルバー（Elliptic PDE Solver）を用いた格子のスムージングが可能である．生成した格子について，ヤコビアンやアスペクト比，ねじれなどの格子品質をチェックする機能が搭載されている．IGES（Initial Graphics Exchange Specification）やSTL（Standard Triangulated Language）などのCAD（Computer Aided Design）データを取り込むことができ，これを用いて格子を生成することができる．また，Gridgen内でCADデータを編集することも可能である．PLOT3DやSTL，その他数多くの市販ソルバー用の書式でデータを出力することが可能である．Tcl/Tkをベースとしたスクリプト言語Glyphを用いて処理を自動化することが可能である．

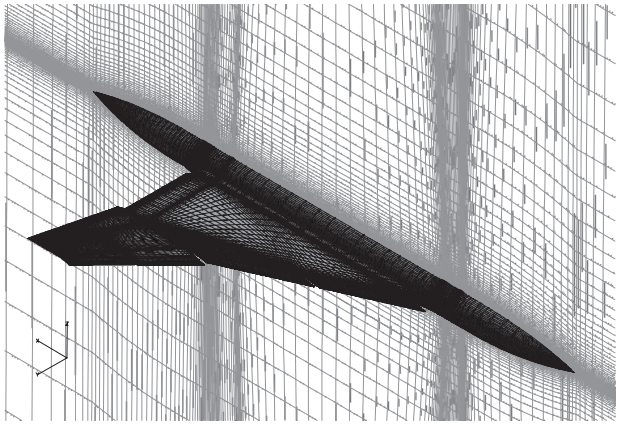
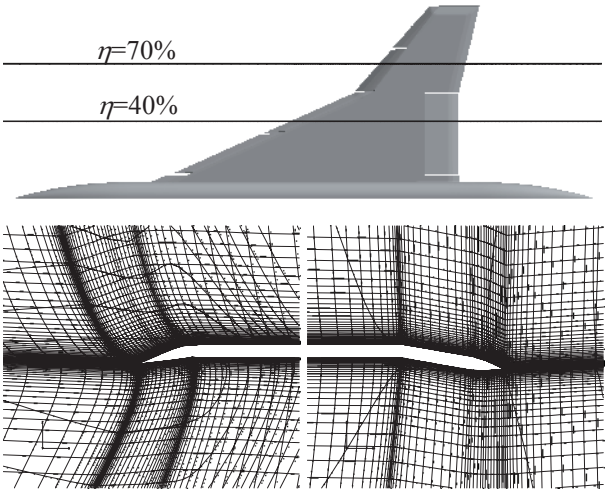


図5　Autoflap-GGを用いて生成した格子

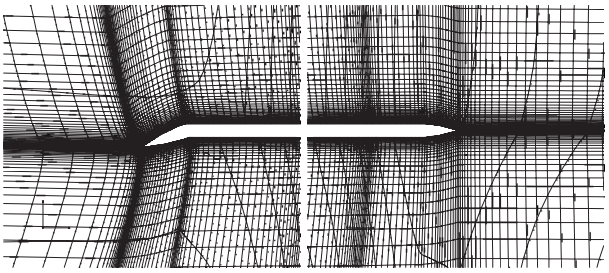
Gridgenでは図4に示すようなボトムアップ手法で格子生成を行う．まず，CADデータを読み込むなどして，物体形状を作成する．Gridgenでは曲線や曲面といった形状を表す要素はデータベース（Database）と呼ばれる．次に，形状の角や計算領域の縁などに沿って，コネクタ（Connector）と呼ばれる曲線状の格子（Curve Grids）を作成する．このコネクタは最終的にドメイン（Domain）の境界とブロックの角に相当し，コネクタ上の格子点分布を調整することで粗密を調節できる．続いて，コネクタで囲むことによって，ドメインと呼ばれる面格子（Surface Grids）を作成する．ドメインは物体表面や物理境界面，ブロック間境界面を表すことになる．ドメイン内部の格子点はTFIによって生成されるが，スムージングを施すことによって品質の高い格子を得ることができる．最後に，ドメインで囲むことによって，ブロックと呼ばれる体積格子（Volume Grids）を作成する．ブロック内部の格子点もTFIで作成され，スムージングを施すことができる．作成した格子にさらに境界条件などを設定した後，格子データ，境界条件データを出力して，ソルバーに渡すことになる．

2.4　格子生成例と計算例

本ツールを用いて生成したフラップ操舵後の形状・格子の一例として，全体図を図5に，前･後縁フラップの内･外翼断面の形状および格子を図6にそれぞれ示す．格子点数は約430万点である．格子トポロジーはH-H型格子を使用し，フラップ間に1個のブロックを配置している．並列CFD計算を行う際の計算効率を考慮して，最終的に計算格子は72個のブロックに分割される．この格子を用いたCFD解析が現在約1日かかることに対して，従来のGridgenを用いた手作業による格子生成では1ケースで約1週間程度の期間を要する．本ツールならば，形状と格子の修正が数分だけで自動的に完了し，1日で1ケース，さらに数ケースを並行して解析することも可能となる．



(a) η=40%（内翼）



(b) η=70%（外翼）

図6　翼断面格子（左：前縁側，右：後縁側）

表1　計算例のフラップ舵角

フラップ	舵角[deg.]
Lef in-1	29.00
Lef in-2	31.00
Lef out-1	23.30
Lef out-2	26.70
TEf	10.00

JAXA超音速機チームで独自に開発されたCFD解析ソルバーを用いて，機体まわりの流れを解析した．図7に代表的な解析結果を示す．このときの形状パラメータを表1に，計算条件を表2にそれぞれ示す．図7のコンターは機体表面の圧力係数C_p分布を，等値線は機軸方向に垂直する断面における総圧分布を示している．このような大きな後退角を持った翼の場合，前縁剥離渦が大きく発生するが，前縁フラップを操舵することによって，前縁剥離渦が小さくなり，剥離が抑制されることになる．

3．入出力ファイルについて

3.1　入力ファイル

AutoFlap-GGで使用する入力ファイルの一覧を表3に示す．基本形態形状・格子データはフラップ操舵前の基本形態の形状・格子ファイルであり，Gridgen V15.10を

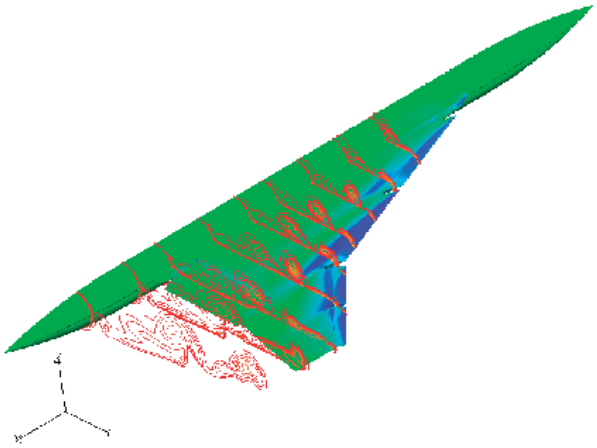


図7　CFD解析結果（機体表面：C_p分布，断面：総圧分布）

表2　計算パラメータ

支配方程式	RANS方程式
乱流モデル	Menter's SST
一様流流速	30[m/s]
迎角	12[deg.]
レイノルズ数	0.945×10^6

表3　入力ファイル一覧

ファイル名	説明
AutoFlap-GG_original.dba	基本形態形状データ
AutoFlap-GG_original.gg	基本形態格子データ
AutoFlap-GG.param	出力ファイル名，舵角の入力ファイル

用いて作成される．パラメータファイル“AutoFlap-GG.param”には出力ファイル名と各フラップの舵角が記述される．パラメータファイルの書式を図8に示す．ここで”#”で始まる行はコメント行であり，読込時に無視される．出力ファイル名は絶対パス，相対パスどちらでもよいが，相対パスの場合にはAutoFlap-GGのインストール先ディレクトリを基準として記述する必要があるため，絶対パスで記述することを推奨する．フラップ舵角は度(degree)で記述する．GUIを用いて対話的に実行する場合には，GUIの入力欄に記述することで自動的にパラメータファイルが作成されるため，別途用意する必要はない．GUIを用いずバッチ処理を行う際には，AutoFlap-GG起動前にあらかじめ書式に合わせてパラメータファイルを作成する必要がある．入力ファイルは全てAutoFlap-GGのインストール先ディレクトリに置かれる．

3.2　出力ファイル

AutoFlap-GGの出力ファイルの一覧を表4に示す．出力ファイルは，生成結果データ，履歴保存用データに大きく分けられる．生成結果データには格子データと境界データがある．格子データは，マルチ・ブロックの

```
#=== AutoFlap-GG.param ===
# Output File Name
C:/user/SST/AutoFlap-GG/SST_LEflap.dba
C:/user/SST/AutoFlap-GG/SST_LEflap.gg
C:/user/SST/AutoFlap-GG/SST_LEflap.grd
C:/user/SST/AutoFlap-GG/SST_LEflap.bnd
C:/user/SST/AutoFlap-GG/SST_LEflap.grd.fvbnd
# LE flap angles
28.33
31.67
25.0
19.33
```

図8　AutoFlap-GG.paramの書式

表4　出力ファイル一覧

拡張子	説明
.dba	Gridgen用格子データ
.gg	Gridgen用格子データ
.grd	PLOT3D形式（ASCII）格子データ
.bnd	境界条件データ
.grd.fvbnd	FIELDVIEW用境界情報データ
.log	ログファイル

PLOT3D形式でテキストデータとして出力される．境界データには，計算で必要となる境界条件を記述したファイルとFIELDVIEWを用いて可視化を行う際に使用するファイルがある．境界データは，Gridgenでソルバーに“generic”を選択した場合の書式で出力され，境界条件IDは，JAXA超音速機チームで開発された並列CFD解析ソルバーADCS用の値で出力される．履歴保存用データにはGridgen用のファイルと実行履歴を記述したログファイルがある．Gridgen用ファイルは，生成された形状データと格子データが別々のファイルに収められる．生成時に使用したフラップ舵角の値や格子品質チェックの結果はログファイルに記述される．特にフラップ舵角は生成された形状や格子から読み取るのは困難であるので，ログファイルを格子データなどと一緒に残すことが望ましい．

各出力ファイルはGUI画面，もしくはパラメータファイルで指定したディレクトリ，ファイル名で出力される．ただし，AutoFlap-GGをバッチ処理用に実行させた場合には，ログファイルはAutoFlap-GGのインストール先ディレクトリに“AutoFlap-GG.log”の名前で出力される．

4．操作手順

AutoFlap-GGのインストール方法，操作手順について解説する．

本ツールを用いた形状・格子生成方法は次のように分けられる．

- 形状・格子の自動生成
- 形状・格子の半自動生成
- バッチ処理

以下に，それぞれについて説明を行う．

4.1　インストール方法

AutoFlap-GGを実行するにはGridgen V15.10以降が必要となる．Gridgenがインストールされていない場合には，先にGridgenのインストールを行う．

(1) AutoFlap-GGの構成ファイルをディレクトリ構造を維持したまま適当なディレクトリに置く．

(2) Gridgenの起動コマンド，インストール先を確認し，AutoFlap-GGの起動スクリプトを修正して，起動スクリプトからGridgenが起動するようにする．起動スクリプトにはWindows用（図9）とUnix/Linux用（図10）があるので，使用環境に該当するOSのものを使用する．修正するのは変数“gg_path”（図中の下線部）である．

(3) AutoFlap-GGを実行するには，修正した起動スクリプトを実行する．実行すると自動的にGridgenとAutoFlap-GGのGUIが起動する．

4.2　GUI画面

AutoFlap-GGには対話的な実行を可能にするためにGUIが実装されている．図11にパラメータ設定用GUIを示す．このGUIは，基本形態データファイル名表示欄，パラメータ入力欄，出力ファイル名入力欄，実行ボタン群，実行経過表示欄で構成される．GUIを立ち上げるとすでに入力欄には何らかのパラメータが入力されている．この値は初回起動の場合には適当な値が入力され，次回以降は前回実行時の値が入力されるようになる．具体的には，AutoFlap-GGのインストール先ディレクトリにあるAutoFlap-GG.param内の値が入力欄に入力された状態でGUIが立ち上がる．

基本形態データファイル名表示欄には基本形態形状・格子データのファイル名がパスを含めて表示される．この欄は使用するファイルの確認用であり，変更することはできない．

パラメータ入力欄では各フラップの舵角を入力する．舵角は度（degree）で入力する．基本的には対応舵角範囲内の値を入力すべきであるが，範囲外の値を入力することもできる．この場合，形状は生成されるが，適切な格子が生成されるとは限らないため，必要に応じて後から手動で修正を行う（4.4節参照）．

出力ファイル名入力欄では各出力ファイルの名前を入力する．出力ファイルについては3.2節を参照すること．出力先ディレクトリを指定するには，絶対パス，もしく

```
: ========================================
:  AutoFlap-GG.bat : ver.1.1.2b
: ========================================
@echo off

:--- Gridgen Install Path ---
set
gg_path=C:¥appli¥Pointwise¥GridgenV15¥Win32¥bin

if "%1"=="-b" goto batch
:normal
  echo ^<^<^< Normal Mode ^>^>^>
  %gg_path%¥gridgen.exe .¥src¥main.glf
  goto end
:batch
  echo ^<^<^< Batch Mode ^>^>^>
  %gg_path%¥gridgen.exe %1 .¥src¥main-b.glf
:end

: End
```

図9　Windows用起動スクリプト

```
#!/bin/sh
# ========================================
#  AutoFlap-GG.sh : ver.1.1.2b
# ========================================

#--- Gridgen Install Path ---
gg_path=/appli/GRIDGEN/gridgen15

if [ $# -eq 0 ] ; then
  echo "<<< Normal Mode >>>"
  $gg_path/gridgen ./src/main.glf
else
  if [ $1 = "-b" ] ; then
    echo "<<< Batch Mode >>>"
    $gg_path/gridgen $1 ./src/main-b.glf
  else
    echo "Error : Unknown Parameter."
  fi
fi

# End
```

図10　Unix/Linux用起動スクリプト
(AutoFlap-GG.sh)

はAutoFlap-GGのインストール先ディレクトリを基準とした相対パスを用いる．各入力欄の右側にあるボタンを押すことでファイル選択ダイアログが表示され，このダイアログを用いて出力先を指定することが可能である．Databaseファイル（Gridgen用形状データファイル）名を入力する際に，ファイル選択ダイアログを用いて出力先を指定すると，他の出力ファイル名，ログファイル名の入力欄には，Databaseファイル名の拡張子のみを変えたファイル名が自動的に入力される．このとき，各ファイルの拡張子は表4に示す拡張子に設定される．

実行ボタン群では各処理を実行するためのボタンが配置されている．形状・格子生成，格子品質チェック，目視による格子確認，ファイル出力，ツールの終了をそれぞれ実行するボタンがある．ファイル出力ボタンは，GUI起動時には選択不可となっているが，格子品質チェックを行い全てのヤコビアンの値が正となれば，選択することが可能となる．

実行経過表示欄では実行中の経過もしくは実行後の結果が表示される．表示される内容は，実行日時，舵角の値，処理内容，格子品質の情報，出力ファイル名である．表示された内容はそのままログファイルにも書き出される．舵角の値を後日確認する場合には，ログファイルを見ればよい．

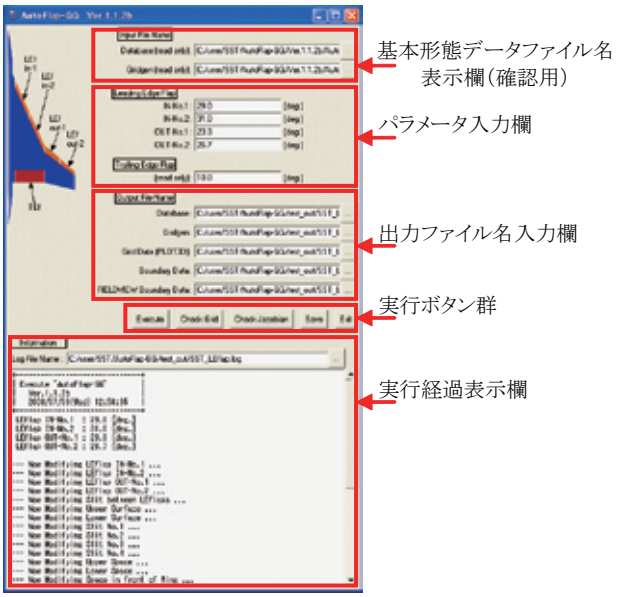


図11　パラメータ設定用GUI

4.3　操作の流れ（形状・格子の自動生成）

GUIを用いた対話的な実行によって，形状・格子を自動的に生成する手順について述べる．本ツールを利用する際の最も一般的な方法である．

(1) 起動スクリプトからGridgenおよびAutoFlap-GGを起動させる．

(2) 表示されたGUIのパラメータ入力欄に各フラップの舵角を度（degree）単位で入力する（図12）．

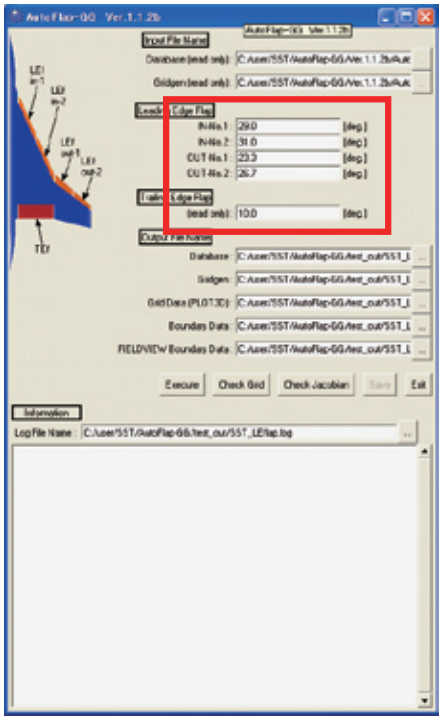


図12　フラップ舵角の入力

(3) 出力ファイル名入力欄に出力ファイル名をパスを含めて入力する．このとき，入力欄右側のボタンを押すことで，ファイル選択ダイアログを用いた入力を行うこともできる（図13）．ファイル選択ダイアログから行う場合には，Databaseの出力ファイル名を選択すれば，残りの出力ファイルのファイル名がDatabaseファイル名の拡張子違いのものに自動で変更される（図14）．

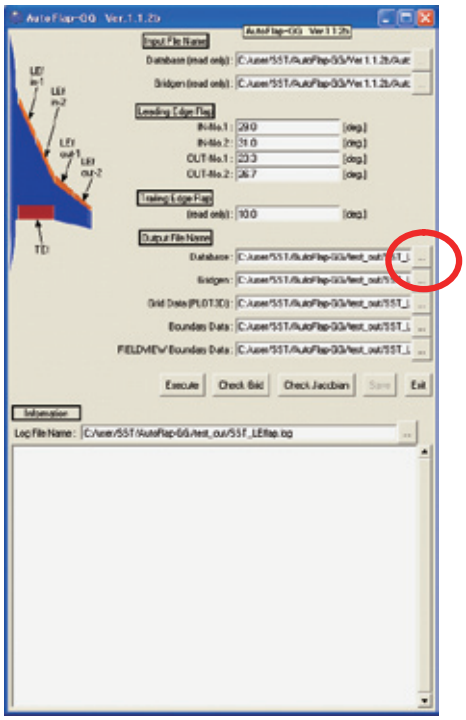


図13　出力ファイル選択ダイアログの表示

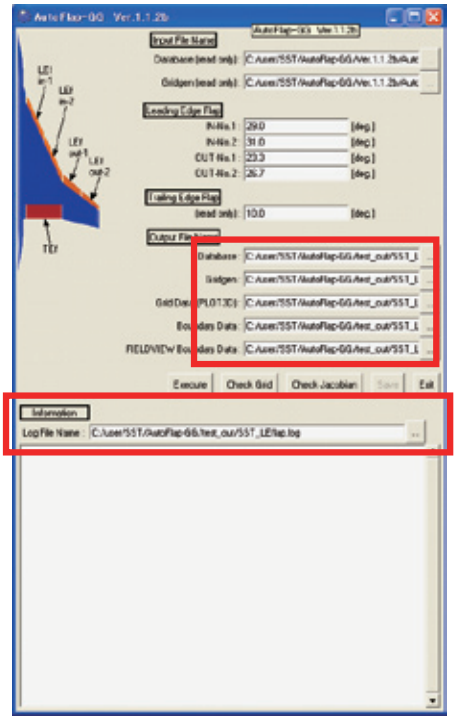


図14　出力ファイル名の入力

(4) 入力を終えたら実行ボタン群の「Execute」ボタンを押し，形状･格子の自動生成を行う（図15）．実行中，処理の経過をGridgen画面上で確認することができ（図16），処理内容を示すメッセージが実行経過表示欄に表示される．

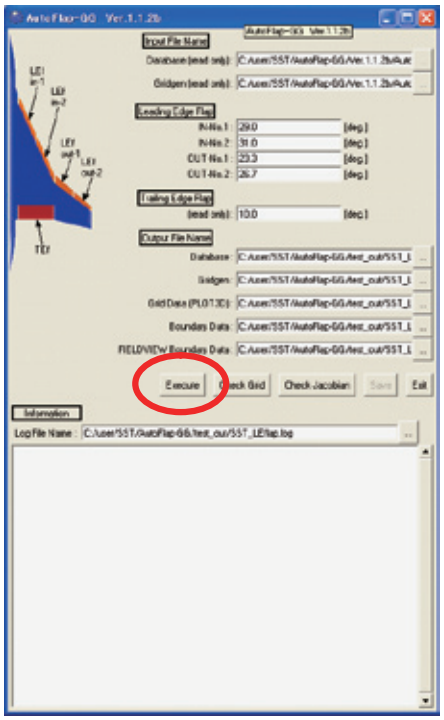


図15　自動形状・格子生成の実行

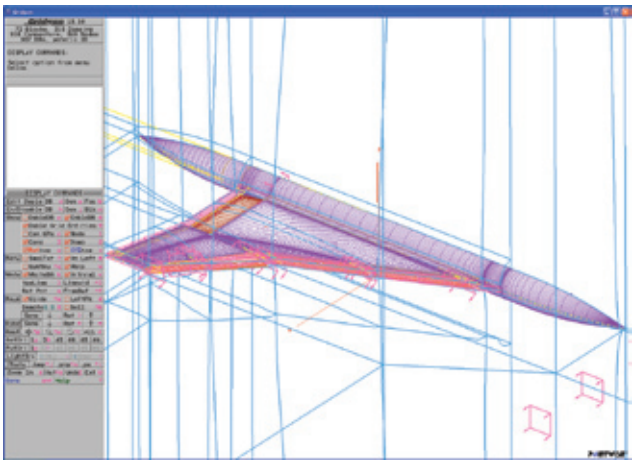


図16　Gridgen実行画面

(5) 形状・格子の自動生成が終了すると，格子品質チェックが行われ，ヤコビアンの値が実行経過表示欄に表示される（図17）．ここで，ヤコビアンの値が全て正（Positive）であることを確認する．

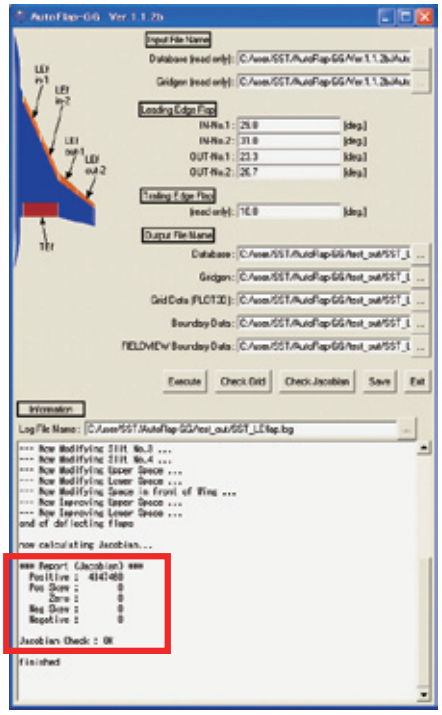


図17　格子品質チェックの結果

(6)「Check Grid」ボタンを押すと，Gridgen画面を操作し，拡大縮小，回転，平行移動させることができるようになるので，目視で格子を確認する（図18）．確認を終えたらEnterを押し，GUI画面に戻る．

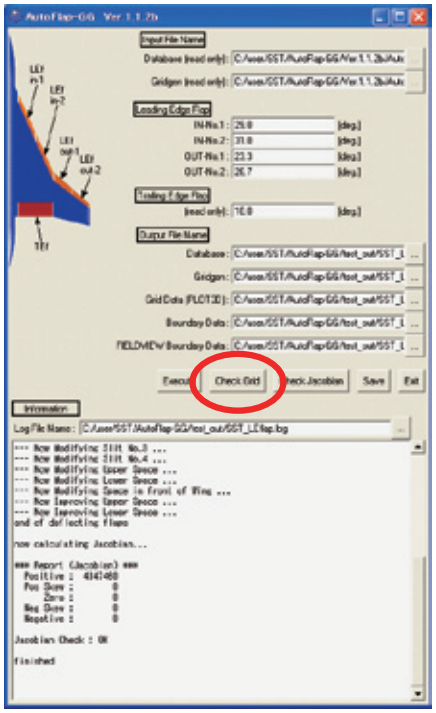


図18　格子の目視での確認

(7) 目視での確認でも問題がなければ，「Save」ボタンを押して，生成された形状・格子データをファイルに出力する（図19）.

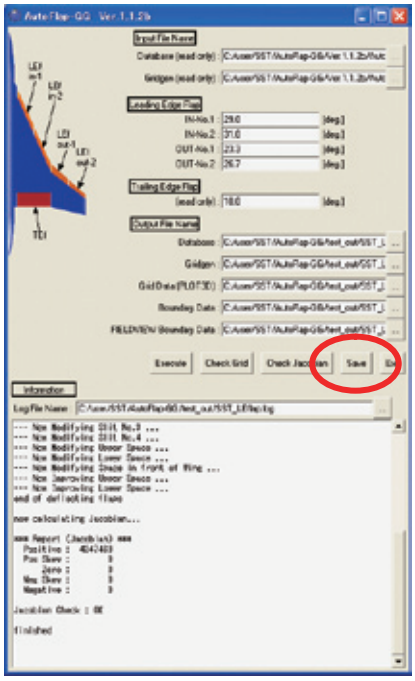


図19　データ出力の実行

(8) 他の舵角についての形状・格子を続けて生成する場合には，そのまま(2)の，フラップ舵角の入力に戻る．終了する場合には，「Exit」ボタンを押し，AutoFlap-GGを終了させる（図20）.

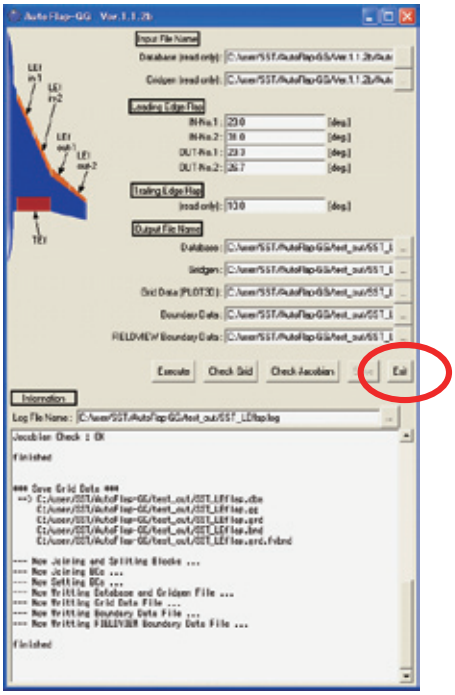


図20　ツールの終了

4.4　操作の流れ（形状・格子の半自動生成）

GUIを用いた対話的な実行によって，形状・格子を自動生成後，Gridgenを用いて手動で格子の修正を行う手順について述べる．これはAutoFlap-GGの対応舵角範囲外の舵角について自動格子生成を行い，その結果適切な格子が得られなかった場合に用いられる．

(1)「Execute」ボタンを押し，形状・格子の自動生成を行うところまでは，4.3節の(1)～(4)と同じである．

(2) 形状・格子の自動生成が終了したら，「Exit」ボタンを押してAutoFlap-GGを一旦終了させる．

(3) その後，Gridgenを用いて格子の修正を行う．このとき，各ブロックのグループ名とインデックスの方向（I, J, K方向）を変えないようにする．これらの情報はファイル出力前のブロック結合・分割の際に使用される．ブロックを作り直す場合にもこれらの情報は元のブロックと同じ状態にする必要がある．格子の修正が終了したら，メニューから「Glyph」－「Reexecute Glyph」を選択，もしくは「Glyph」－「Execute Glyph」を選択後，AutoFlap-GGのインストール先ディレクトリ内にある“src/main.glf”を選択して，AutoFlap-GGを再度起動させる．

(4) AutoFlap-GGのGUIが表示されたら，「Check Jacobian」を押し，格子品質チェックを行う（図21）．これ以降，実行経過表示欄に表示されるメッセージはログファイルに追記される．

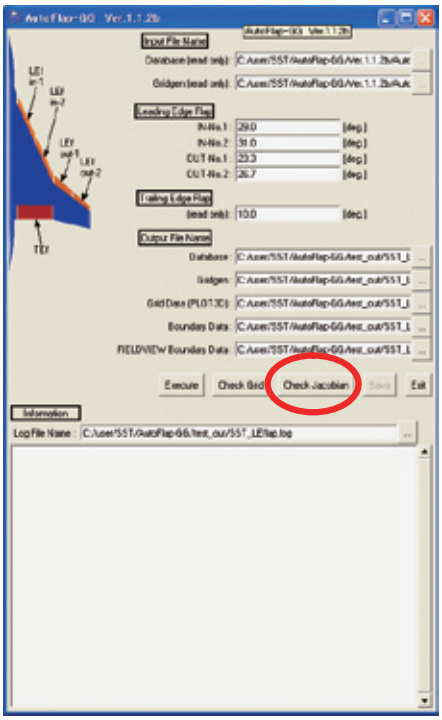


図21　格子品質チェックの再実行

(5) 格子品質チェックで全てのセルのヤコビアンの値が正であれば,「Save」ボタンが有効になる．ヤコビアンに問題がある場合には(2)に戻り,「Exit」ボタンを押して，再度格子の修正を行う．
(6) ヤコビアンに問題がなければ「Save」ボタンを押して形状・格子データをファイルに出力する．
(7) 他の舵角についての形状・格子を続けて生成する場合には，そのまま舵角，出力ファイル名の入力に戻る．終了する場合には,「Exit」ボタンを押し，AutoFlap-GGを終了させる．

4.5 操作の流れ（バッチ処理）

GUIを用いず，バッチ処理によって形状・格子生成からデータ出力までを全て自動で行う場合の手順について述べる．最適化設計ループに組み込む場合など，人を介さず，指定されたパラメータに従って自動的に形状・格子を生成する場合に用いられる．

(1) パラメータファイル“AutoFlap-GG.param”を作成し，各フラップの舵角と出力ファイル名を記述する．パラメータファイルの書式については3.1節を参照すること．
(2) パラメータファイルをAutoFlap-GGのインストール先ディレクトリに置き，GridgenおよびAutoFlap-GGを起動させる．このとき，コマンドライン引数として“-b”を与える．“-b”を与えることでGridgenおよびAutoFlap-GGはバッチモードで実行され，バックグラウンドで自動的に処理が行われるようになる．GUI使用時に実行経過表示欄に表示されるメッセージは，標準出力に出力される．この際のログファイルは，AutoFlap-GGのインストール先ディレクトリに“AutoFlap-GG.log”の名前で出力される．起動時のコマンドは以下のようになる．

```
Windowsの場合
> AutoFlap-GG.bat –b
UNIX/Linuxの場合
> AutoFlap-GG.sh -b
```

5． AutoFlap-GG処理内容

5.1 概要

AutoFlap-GGの処理とデータの流れを図 22に示す．AutoFlap-GGは自動処理部とGUI部に大きく分けられ，自動処理部は形状・格子生成処理，格子品質チェック，データ出力の3つのルーチンで構成されている．自動処理部とGUI部とは独立して実装されているため，通常のGUIを介した処理とバッチモードでの処理は全く同じルーチンが用いられる．GUI画面で入力された値はパラメータファイルに一旦出力され，自動処理部で再び読み込んでいる．バッチモードで動作させる場合には，あらかじめユーザがパラメータファイルを用意しておき，自動処理部で同様に読み込んでいる．このようにGUIを介する場合にもパラメータファイルを用いることで，同じルーチンをバッチモードでも使用できるようにしている．

起動スクリプトはGridgenを起動させ，起動直後に指定されたGlyphファイル，すなわちAutoFlap-GGを実行させている．Gridgenにコマンドライン引数としてGlyphスクリプトファイル名を与えることで，Gridgen起動直後にそのGlyphが実行される．また，“-b”のオプションを付けることでGridgenがバッチ処理用にバックグラウンドで実行されるようになる．Gridgenを起動させるコマンドの書式は以下のようになる．

```
Gridgenを起動させる場合（通常使用）
> gridgen

Gridgen起動直後にGlyphを実行する場合
> gridgen ”Glyphファイル名”

バッチ処理用にGridgenを実行する場合
> gridgen –b ”Glyphファイル名”
```

起動スクリプト実行時にコマンドライン引数を指定しない場合にはGUIが表示される．このとき，AutoFlap-GGのインストール先ディレクトリにすでにパラメータファイルが存在している場合には，パラメータファイル内の値が自動的にGUI上の入力欄に入力される．パラメータファイルが存在しない場合には, 適当な値が入力される．GUI画面上の「Exit」を除くいずれかのボタンを押すと，GUI画面で入力した舵角や出力ファイル名がパラメータファイルに出力され，その後，それぞれの処理に入る．形状・格子生成処理では，もしパラメータファイルや基本形態形状・格子データの読込に失敗すると，GUI画面に戻る．無事形状・格子を生成できると，続けて格子品質チェックの処理に移行する．格子品質チェックは単独でも実行できるようになっている．これは，自動生成された格子をGridgenを用いて手動で修正する場合，格子修正後に品質チェックのみを行う必要があるためである．格子品質チェックで全てのヤコビアンが正であることを確認すると,「Save」ボタンが有効になり，データ出力に移行することができるようになる．格子品質チェックをパスするまでは「Save」ボタンは無効になっており，データ出力を実行することはできない．データ出力と形状・格子生成処理とは独立であるので，出力ファイル名を取得するために，データ出力の最初で再びパラメータファ

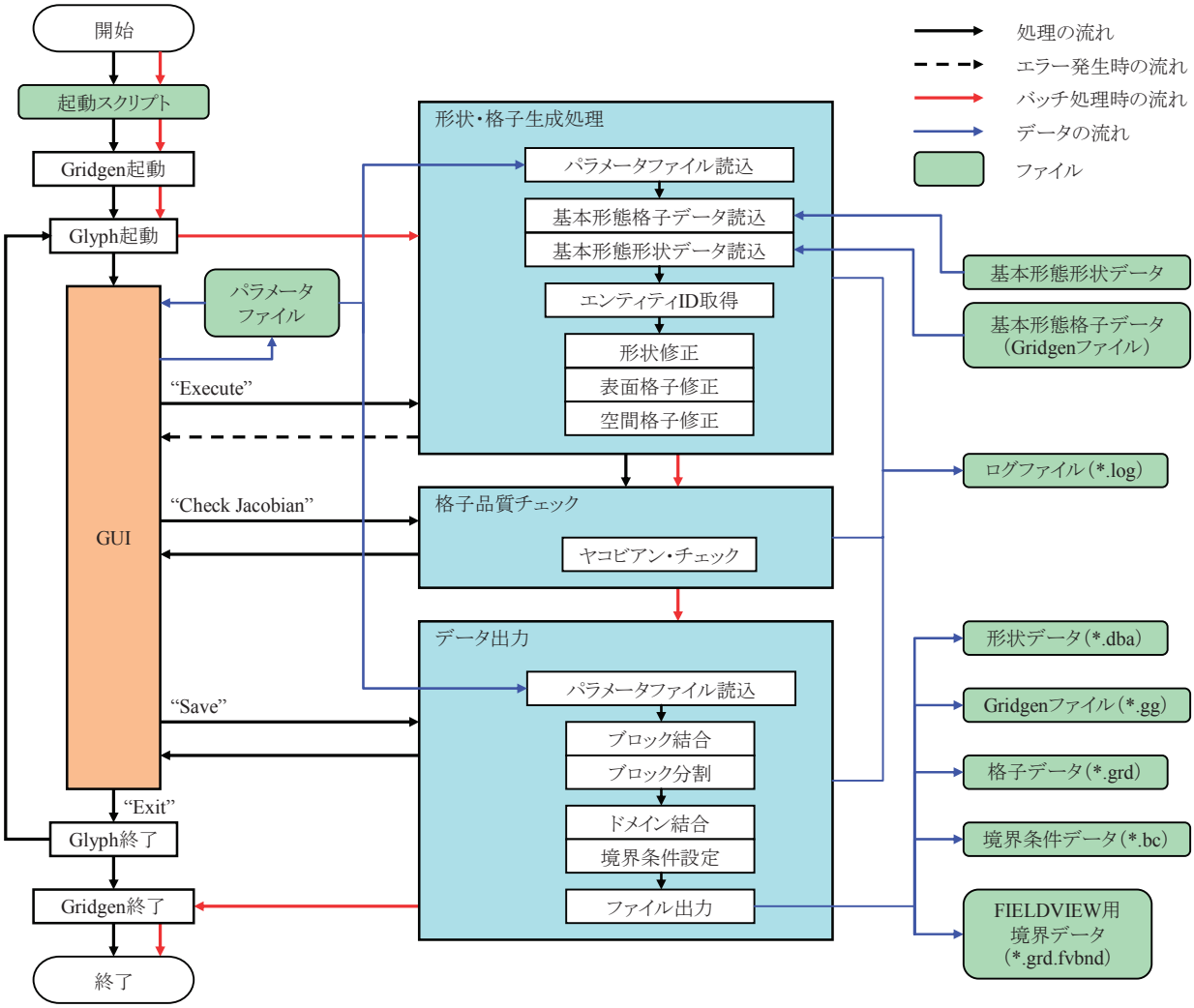


図22　AutoFlap-GGの処理とデータの流れ

イルを読み込んでいる．自動処理部で出力されたメッセージはGUI画面の実行経過表示欄に表示されると同時に，ログファイルにも出力される．GUI画面の「Exit」ボタンを押すことで，AutoFlap-GGが終了しGridgenの通常の操作が行えるようになる．この状態で再びGlyphスクリプトを実行することで，AutoFlap-GGを起動し，GUI画面に戻ることもできる．

一方，バッチモードでは図22中の赤い矢印の順に処理が行われる．バッチモードでAutoFlap-GGを実行させると，GUIを通らず，そのまま自動処理部に入る．自動処理部では，形状・格子生成処理，格子品質チェック，データ出力の順に実行される．データ出力が終わると，Gridgenを終了させ，一連の処理が終了する．

自動処理部における各処理の詳細は次節以降で述べる．

5.2　形状・格子生成処理

形状・格子生成処理は，AutoFlap-GGで中心的な処理を行うルーチンであり，形状・格子生成に関わる処理は全てこのルーチンに集約されている．形状・格子生成処理は，コードの独立性を確保するためにパラメータファイルおよび基本形態形状・格子データの読込から始まる．基本形態形状・格子データから必要なエンティティ（Entity）のIDを取得し，形状・格子の修正を順次行っている．ここでエンティティとは，形状データ（曲線，曲面）やコネクタ，ドメイン，ブロックをまとめて表す用語である．各処理の詳細を以下に述べる．

5.2.1　パラメータファイル読込

パラメータファイルを読み込み，各フラップの舵角を取得する．ここでパラメータファイルは，AutoFlap-GGのインストール先ディレクトリ上に“AutoFlap-GG.param”の名前で置いておく必要がある．パラメータファイルの書式については3.1節を参照すること．

5.2.2　基本形態形状・格子データ読込

基本形態形状・格子データとして，Gridgen用形状データファイル（Database File）とGridgen用格子データファイル（Gridgen File）を読み込む．AutoFlap-GGでは，生

成する形状・格子の基準として基本形態形状・格子を用いており，これに対して修正を加えることにより形状・格子生成を自動化している．基本形態は各フラップの舵角が0[deg.]の，操舵前の状態を指し，この形態に対する形状・格子がそれぞれ基本形態形状・格子である．生成される格子の格子点数，トポロジー，および格子間隔は基本形態格子と同じになるように処理が行われる．そのため，基本形態格子の格子間隔やコネクタ形状等を修正することで，AutoFlap-GGのコードを書き変えることなく，生成される格子をある程度制御することができる．格子点数，トポロジーの変更に対応するためにはコードを修正する必要がある．基本形態形状・格子データはAutoFlap-GGのインストール先ディレクトリに置いてあり, 基本形態形状データファイルが“AutoFlap-GG_original.dba”，基本格子データファイルが“AutoFlap-GG_original.gg”である．

5.2.3 エンティティ IDの取得

形状・格子修正に必要なエンティティのIDを基本形態形状・格子データから取得する．Glyphではコネクタやドメイン等を修正する際，その修正するエンティティのIDを指定しなければならない．このIDを取得する方法は複数あり，エンティティの種類によっても異なる．ここではエンティティ IDの取得方法について述べる．

Gridgenでは各エンティティのIDは文字列で表される．そのため，IDそのものを保持しておく方法がある．しかしIDには一貫性がなく，同じエンティティに同じIDが常に割り振られるわけではない．データの読込や修正などの処理の順番が異なると，同じIDを使用していても異なるエンティティが選択される恐れがある．この方法は再現性が確保できない可能性があるため基本的には使用すべきではない．Gridgenに搭載されているGlyphのジャーナリング機能を使用して生成されるスクリプト内でも，直接IDを用いる方法はとっておらずエンティティ番号から取得している．

コネクタIDは，コネクタ番号，グループ名，ドメイン，境界条件名から取得することができる．ここで境界条件名からの取得は2次元格子作成時のみ可能である．ドメインから得る場合には，指定されたドメインを構成するエッジに対応するコネクタのIDを得る．グループ名，ドメイン，境界条件名から取得する場合，1つもしくは複数のIDを一度に取得することになる．なお，グループ名はコネクタ，ドメイン，ブロックそれぞれ別々に定義され，コネクタに関するグループにはコネクタのみが含まれており，ドメイン，ブロックについても同様である．

ドメインIDは，ドメイン番号，グループ名，コネクタ，ブロック，境界条件名から取得することができる．ここで境界条件名からの取得は3次元格子生成時のみ可能である．コネクタから取得する場合には，指定されたコネクタをエッジに持つドメインのIDを得る．ブロックから取得する場合には，指定されたブロックを構成するフェイスに対応するドメインのIDを得る．グループ名，コネクタ，ブロック，境界条件名から取得する場合，1つもしくは複数のIDを一度に取得することになる．

表5 エンティティ IDの取得

エンティティの種類	指定元	関連プロシージャ
コネクタ	番号	conGetAll
	グループ名	conGroupGetCons
	ドメイン	domGetEdge
	境界条件名	aswSetBC
ドメイン	番号	domGetAll domGetByNum
	グループ名	domGroupGetDoms
	コネクタ	domGetEdge
	ブロック	blkGetFace
	境界条件名	aswSetBC
ブロック	番号	blkGetAll
	グループ名	blkGroupGetBlks
	ブロック名	blkName
	ドメイン	blkGetFace
	体積条件名	aswSetVC
データベース（形状データ）	番号	dbGetAll
	名前	dbGetByName
	ドメイン	domEllSolverAtt

ブロックIDは, ブロック番号, グループ名, ブロック名, ドメイン，体積条件名から取得することができる．ドメインから取得する場合には，指定されたドメインをフェイスに持つブロックのIDを得る．体積条件は，解析時にセルが流体か固体かを識別するときなどに使用されるパラメータである．グループ名，ドメイン，体積条件名から取得する場合，1つもしくは複数のIDを一度に取得することになる．

データベースIDは，データベースの番号，名前，ドメインから取得することができる．ドメインから取得する場合には，ドメインのスムージング時に格子点が投影される先のデータベースのIDを得る．

実際にGlyph内でIDを取得する際には，Glyphで用意されているプロシージャ（C言語における関数とほぼ同じものであり，Tcl/Tkにおける呼び名）を用いるだけでよいものと，プロシージャを組み合わせなければならないものとがある．また，各方法を組み合わせることによって，より細かい条件を用いてエンティティのIDを取得することも可能となる．表5にエンティティ ID取得方法と関連するプロシージャをまとめる．

AutoFlap-GGでは，基本形態形状・格子を修正したとしてもコードを修正する必要が生じない方法を採用した．

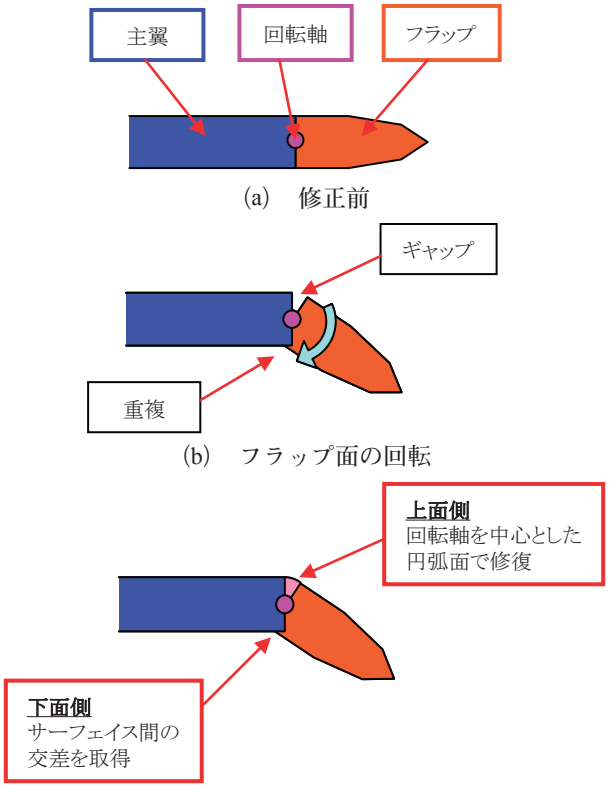


図23　形状修正

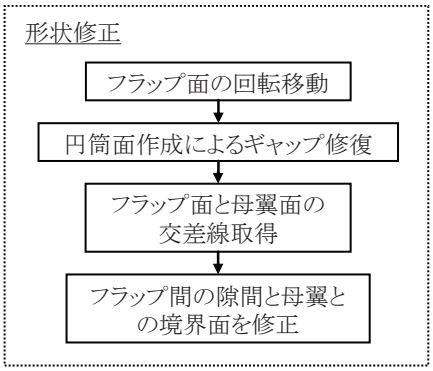


図24　形状修正の流れ

採用したエンティティ ID取得方法は以下の通りである．

- グループ名からコネクタ ID を取得
- グループ名からドメイン ID を取得
- 番号，グループ名からブロック ID を取得
- 名前からデータベース ID を取得

エンティティ番号は基本形態形状・格子データを修正することで大きく変化してしまう可能性があるため，番号からのID取得はなるべく避けるようにした．グループ名からのID取得では一度に複数のIDを取得できるが，個々のエンティティに対して修正する場合にはさらにその中から単一のIDを抽出する必要がある．そのため，1つのエンティティに対して1つのグループ名を設定し，複数のIDの中から単一のIDを抽出する手間を省いている．ブロックについては，コネクタやドメインに比べて

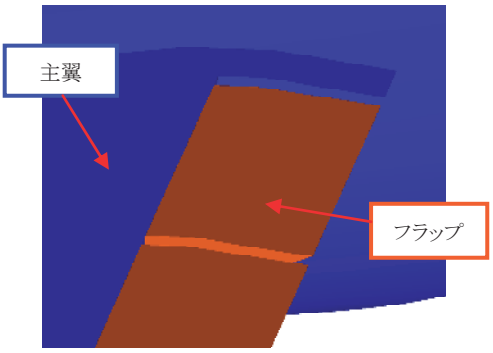


図25　形状修正前

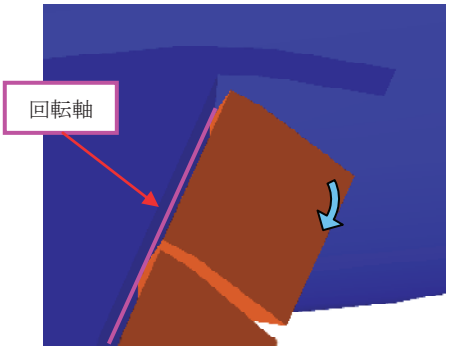


図26　フラップ面の回転

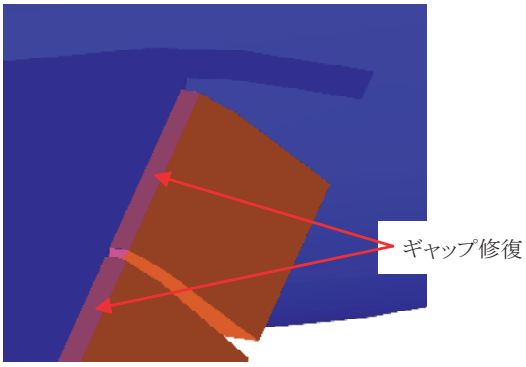
数が少なく，Gridgen内で順番を操作することが可能であるため，一部番号からIDを取得した．また，エンティティの分割を行った際など，場合に応じてエッジやフェイスを用いてIDを取得した．データベースについては全ての要素に名前が付けられているため，名前からIDを取得した．

5.2.4　形状の修正

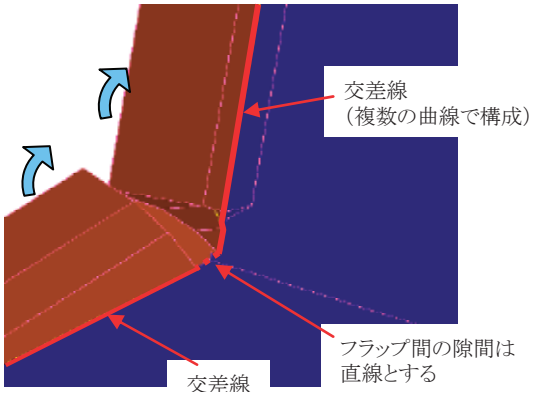
フラップの操舵は形状変化を伴う．ここではAutoFlap-GG内で行っている形状修正について述べる．形状修正として図 23に示すように，ヒンジ・ライン周りのフラップ面の回転移動と，ギャップや曲面間の交差に対する修復を行う．以降，操舵しない主翼の固定部を母翼と呼ぶ．

図 25は形状修正前の基本形態の状態である．ヒンジ・ラインは直線であり，指定された舵角の分だけフラップ面をヒンジ・ライン周りに回転移動させる．このとき回転の方向は直線の向き，すなわち直線の始点と終点の位置関係によって変わり，回転角を正負どちらで指定するかが変わる．ここで使用した基本形態形状のデータでは，ヒンジ・ラインは胴体側を始点，翼端側を終点となっているため，回転角は負の値で指定する必要がある．フラップ面を回転させることで図 26の状態になる．

フラップ面の回転によって，上面側ではフラップ面と母翼面との間にギャップが生じ，下面側ではそれぞれの面が交差してしまう．上面側のギャップは新たに曲面を作成することで対処した．元のフラップ面と母翼面との境



(a)　上面側



(b)　下面側

図27　形状の修復

界線を境界とし，ヒンジ・ラインを中心線とした円筒面を回転角の分だけ作成する．フラップ端面は平面としており，これは操舵後も変わらないものとする．円筒面はフラップ幅より大きめに作成し，後からフラップ端面に対応する平面で分割している．このとき分割に使用する平面は，回転後のフラップ面について端面に接する3点の座標を取り出して作成した．円筒面を作成しギャップを修正した状態が図 27(a)である．元のフラップ面と母翼面との境界線が複数の直線で構成されている場合から成っている場合，円筒面も複数の曲面で構成される．このような場合には，胴体側から順に番号を振ることで，位置関係を明確にしている．下面側のフラップ面と母翼面との境界線は，回転後のフラップ面と母翼面との交差をとり，これを境界線とした．境界線が複数の曲線で構成される場合には，胴体側から順に交差をとることで，得られる境界線のデータベースIDと位置関係の対応がとれるようにした．下面側の交差線は図 27(b)のようになる．

図 28に示すように，フラップとフラップの間，フラップ端と母翼の間には隙間が設けてあり，形状が変化したことによってこれらの隙間についても形状修正が必要となる．フラップ端面は平面であり，特に処理は行わず，端面そのものを表す平面を作成することはしない．隙間と母翼との境界面については次のように修正を行った．上面側はヒンジ・ラインがフラップ端面に対して垂直で

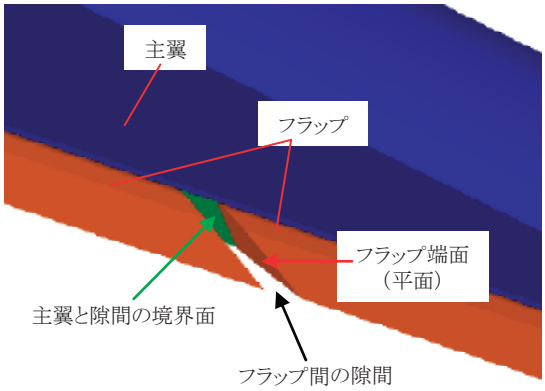


図28　フラップ間の隙間と隣接する面

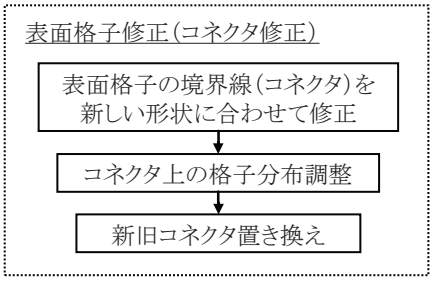


図29　表面格子修正におけるコネクタ修正の流れ

はないため，フラップ面と母翼面との境界線が移動することになる．隙間と母翼との境界面の上面側の辺は元々直線であり，変形後も直線としている．下面側については，フラップの操舵によって境界線の位置が元々の境界線より母翼側に位置するようになる．境界面の下面側の辺についても元々直線であり，変形後も直線であるとして，フラップ面と母翼面との交差線の端を直線で結んで作成している．最終的には隙間と母翼との境界面はねじれた面となるが，境界面そのものは作成せず，上下の辺のみ修正を行った．

5.2.5　表面格子の修正

5.2.5.1　コネクタの修正

コネクタ（Curve Grids）は物体形状の角を形作り，格子点の粗密を調節する上で重要である．図 29に表面格子修正の処理におけるコネクタ修正の流れを示す．ここではAutoFlap-GG内で行っているコネクタの修正について述べる．

Gridgenではデータベースの変化に追随してコネクタやドメインを自動的に修正する機能があるが，必ずしもうまく働くわけではない．そのため本ツール内では，データベースの修正によって影響を受けるコネクタに対して別途修正を施している．コネクタについて，始点や終点，中間点を示すコントロールポイントが配置されると，コントロールポイント間の形状は直線やスプライン曲線，Bezier曲線などで表現される．コントロールポイントは図 31のように，3次元デカルト座標系，もしくは各形状

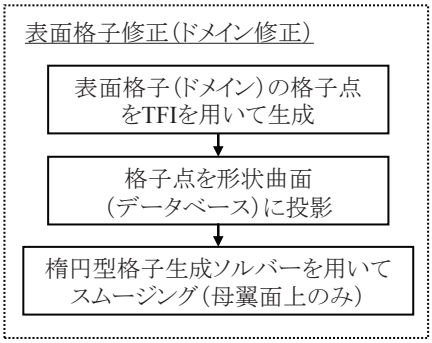


図30　表面格子修正におけるドメイン修正の流れ

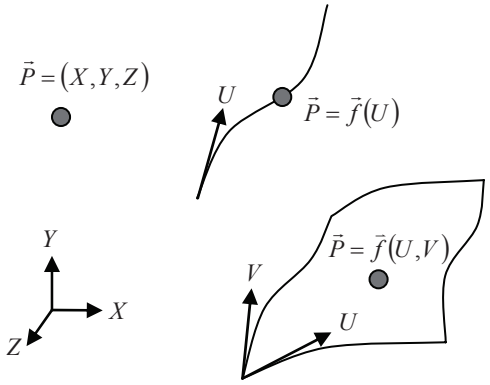


図31　デカルト座標系とパラメトリック座標系

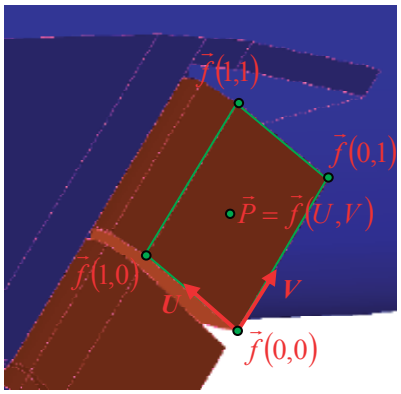


図32　パラメトリック座標を用いた位置指定

に対するパラメトリック座標系で指定することができる．パラメトリック座標系で指定した場合には，ポイント間の曲線もパラメトリック座標系で表現されるようになる．そのためパラメトリック座標系で指定することによって，コネクタを確実に形状上に乗せることができる．パラメトリック座標の値は0～1の間で表現され，曲面の場合，例えば図 32に示すような位置関係となる．曲線の場合，始点を原点として終点に向かう方向にU方向が定義され，始点と終点の間は$0<U<1$で表される．曲面の場合，相対する2辺に沿う方向にUもしくはV方向が定義され，UとVはそれぞれ0～1の間で変化する．曲面の各頂点は0と1の組み合わせで表現される．

Gridgenでは，コネクタ上の格子点分布はtanh関数を用いて算出され，始点，終点の格子幅を与えることによって粗密を調節できる．実際にはGridgenにはより細かく格子点分布を制御する機能があるが，本ツールで対象としている形状の場合，そのような機能を使う必要性はなかった．コネクタ上の格子点数，端点の格子幅は，基本形態格子の値が自動生成後の格子に引き継がれるようにしている．

実際のコネクタ修正は，コネクタを新しく作成し，これを古いものと置き換えている．既存のコネクタに修正を加えた場合，処理の内容によっては他のコネクタやドメインに変更が加えられてしまう可能性がある．特にGlyphを用いて自動処理する際には，全ての修正処理を一度には行えず，途中で一旦確定してから，再度修正処理を行わなければならない場合がある．この一旦確定する段階で他のコネクタやドメインに変更が加わってしまうのである．そうなると修正前後で格子点幅のパラメータに差異が生じる恐れがある．また，中間点の有無など，それぞれのコネクタの状態に合わせてコードを組む必要があり，コードが複雑になってしまう．それに対してコネクタを新たに作成する場合には，最終的に得られるコネクタと元のコネクタとを置き換えればよいので，作成途中の状態が格子データに影響することはない．また，既存のコネクタの状態に依存せず，各コントロールポイントの座標値がわかればよいのでコードを簡素化，一般化することが容易である．

格子点分布が変化する箇所は，形状修正を行ったフラップ面近傍だけである．フラップのコード方向については上面側でコネクタが長くなり，下面側では逆に短くなる．コネクタ上の格子点分布は，端点の格子幅のみで決まるので，この値を基本形態格子と同じ値にした．図 33に示すように，形状変更に伴ってフラップ面と母翼との境界位置が変化し，フラップ間の隙間の幅が変化するため，ここの格子間隔を修正する必要がある．この隙間の幅はフラップの舵角の組み合わせによっては，極端に狭くなる場合があり，そのような場合には基本形態格子の格子幅をそのまま当てはめてもエラーが発生してしまう．そのため，コネクタ長さを用いて，基本形態格子の格子幅を用いるか，分布を等間隔にするかを判断させている．

形状修正が必要なコネクタとして，図 34に示す内翼，外翼の境界を表すコネクタがある．AutoFlap-GGが対象としている機体はクランクド･アロー翼を採用しており，主翼は内翼と外翼とがキンク位置でつながった形状である．内翼と外翼とでは後退角が異なり，ドメインをキンク位置で分けた方が良い格子が生成できる．また，結果処理の観点からも，内翼と外翼は分けておいた方が都合が良い．図 34に示すようにフラップの操舵に伴い，キン

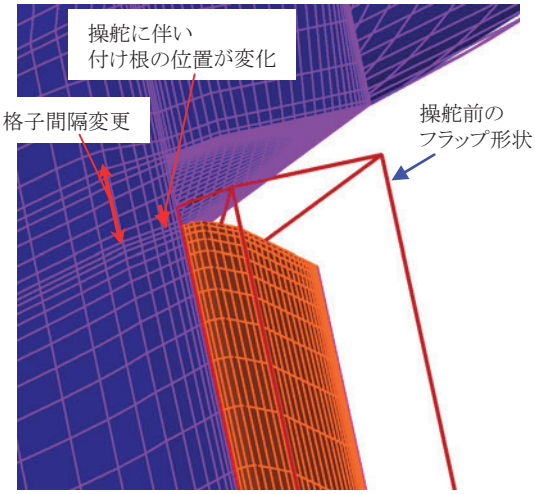


図33　形状変更に伴う格子点分布の変化

クにおける前縁位置が変化するので，コネクタを途中で曲げる必要がある．ここでは単純にコネクタを前縁フラップ近くで折り曲げて，キンク位置に対応するフラップ端面と結んだ．

5.2.5.2　ドメインの修正

ドメイン（Surface Grids）は物体形状を正確に再現するために重要である．ドメインはコネクタで囲まれた面格子であり，ブロックの境界面を表す．ブロック内部の格子は境界面であるドメインを元に作成されるため，ドメインの格子品質は格子全体の品質に大きく影響する．図30に表面格子修正処理におけるドメイン修正の流れを示す．ここではAutoFlap-GG内で行っているドメイン，特に物体表面格子の修正について述べる．

Gridgenでは代数型格子生成法（TransFinite Interpolation, TFI）を用いて格子を生成し，楕円型格子生成ソルバー（Elliptic PDE Solver）を用いてスムージングを施すことができる．AutoFlap-GGでは，比較的面積の広い主翼母翼面上の格子に対してのみスムージングを施しており，それ以外の修正箇所についてはTFIを用いて格子を生成している．ただし，TFIは境界上の格子点座標を用いて内部の格子点を内挿し生成する手法であり，境界上以外は拘束されていない．そのため，TFIで生成した直後の格子点は物体形状に乗っていない．従って，TFIで生成された格子点を物体形状に投影する処理が行われる．ドメインをデータベースに投影する処理をGlyphで行う場合には，有効となっているデータベース全てが投影先の対象となり，その中で最も距離の近いものに投影される．そのため余分なデータベースが有効になっていると，意図しない場所に投影される恐れがあるので，投影したいデータベースのみを有効とし，それ以外は全て無効に切り替える必要がある．また，データベースそのものに余分な形状が含まれている場合にも同様のことが起きる恐れがあ

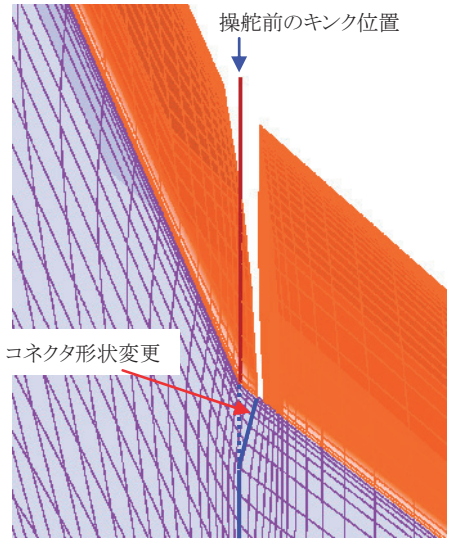
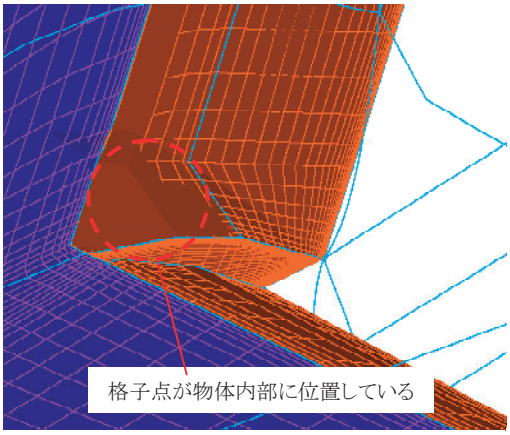


図34　形状変更に伴う格子点分布の変化

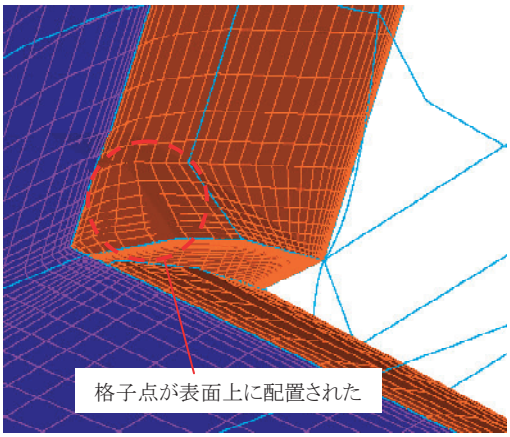
るため，Trimmed Surfaceを用いて余分な形状を無くすようにした．図 35に投影前後の格子点の例を示す．投影前の図 35(a)では，TFIで生成した格子点は物体内部に位置しており，正しく物体形状を再現できていない．格子点のデータベースへの投影を行うことで，図 35(b)のように格子点が物体表面に乗り，正しく物体形状を再現できるようになる．

スムージングは格子品質を改善するために行われるが，AutoFlap-GGでは主翼母翼面上のドメインに対してのみスムージングを施し，それ以外のドメインについてはTFIのみで十分品質の良い格子が得られた．AutoFlap-GGで対象としている機体形状の場合，フラップの操舵によってフラップ面と母翼面の境界位置が変化し，その結果境界線位置の格子点分布が変化する．そのため，母翼面上の格子も影響を受け，格子を生成し直す必要が生じる．主翼母翼面は比較的面積が広く，空力性能を見積もる上で重要であるため，スムージングを施すことによって品質向上を図った．スムージングを施す場合には，あらかじめ投影先のデータベースを設定しておけば，格子点の物体形状への投影はスムージングの処理の中で自動的に行われるようになる．ここでのスムージングはドメイン内部の格子を滑らかにすると同時に，境界付近の格子も隣接するドメインの格子と滑らかにつながるように修正される設定をしている．また主翼母翼面上のドメインは，内翼と外翼の2つのドメインに分けられており，この境界付近の格子についても格子線が滑らかにつながるようにしている．

Gridgenのスムージングでは，ドメインの境界の設定を変えることによって，スムージング時の境界付近の処理を変えることができるようになっている．ドメイン境界に“Adjust”を設定することによって，隣接するドメイ

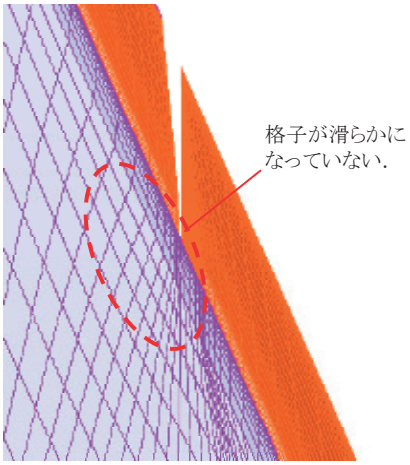


(a)　投影前

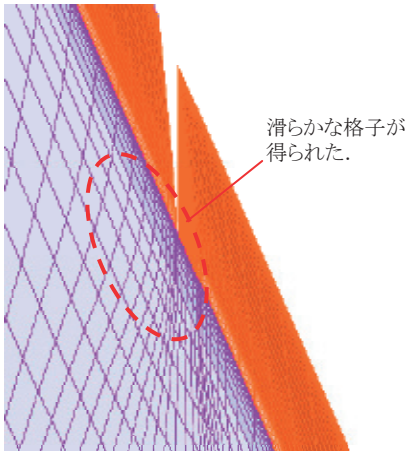


(b)　投影後

図35　表面格子の形状への投影



(a)　スムージング前



(b)　スムージング後

図36　表面格子のスムージング

ンと滑らかに接続する格子を得ることができる.

ただし，Glyphに限ったことではないが，この“Adjust”が有効に働くためにはいくつか条件がある．まず，スムージングの処理は複数のドメインに対して実行することができるが，“Adjust”が設定されている境界に隣接するドメインもそのスムージング処理の対象に入っている必要がある．そのため，ドメインを選ぶ際に，スムージングを施すドメインだけではなく，格子線を合わせたい隣のドメインも一緒に選択しなければならない．もし，隣接するドメインに対してスムージング処理を行う必要がなければ，そのドメインに対してスムージングのパラメータである緩和係数の値を0に設定すればよい．このように設定することで，スムージング処理の結果はそのドメインに反映されなくなる．次に，スムージング処理の反復計算の過程で，“Adjust”が設定された境界が実際に滑らかにつなごうとする格子線は，反復計算開始時の隣接ドメインの格子線である．このことは，2つの隣接するドメインをスムージングしつつ，両ドメインの境界の格子線を滑らかにする場合に問題となる．“Adjust”が合わせようとする格子線は反復計算開始時のものであるため，スムージングによって隣の格子線が大きく変わったとしても，それに追従するようなことはしてくれない．このような場合には片方のドメインにスムージングを施してから，他方のドメインをそれに合わせてスムージングする必要がある．

AutoFlap-GGでは，内翼面上のドメインに対して先にスムージングを施してから，内翼面上の格子線に合わせて外翼面上のドメインにスムージングを施している．図36はスムージング前後の格子線の例であり，スムージング前に折れ曲った格子線があったのが，スムージングを施すことで滑らかになっているのが分かる．

5.2.6　空間格子の修正

ブロックの格子品質は計算精度に影響を与え，非常に歪んだ格子の場合には計算そのものが実行できないことがある．表面格子ができたら，次に空間格子を修正する．図 37に空間格子修正の処理の流れを示す．ここではAutoFlap-GG内で行っているブロックの修正について述べる．

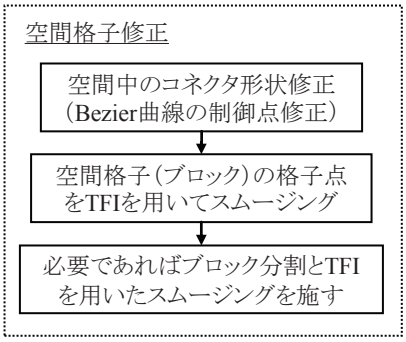


図37　空間格子修正の流れ

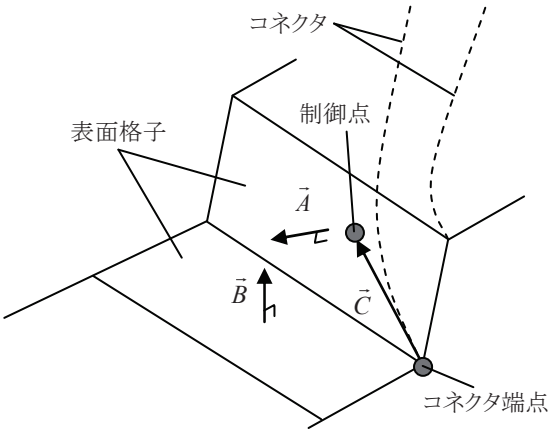
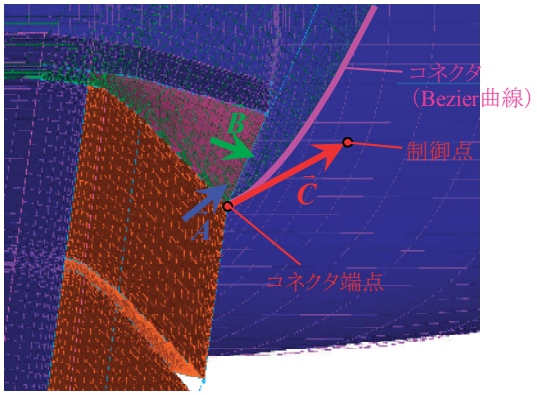


図38　空間中のコネクタ修正概念図

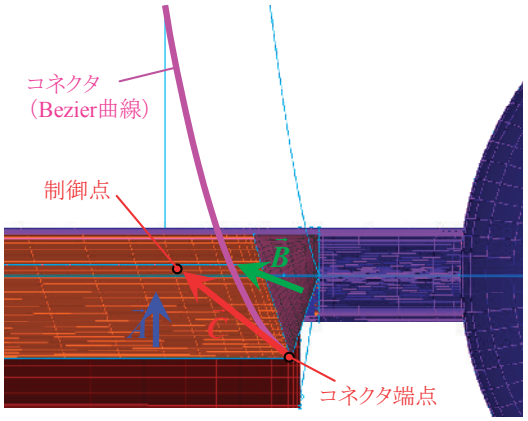
ブロックはドメインで囲むことによって作成され，ドメイン上の格子点を元にTFIを用いてブロック内部の格子点が生成される．そのため，ブロック内部の格子の修正は境界面を修正することで対応しており，主に機体表面から外に向かうコネクタまたはドメインを修正することで対応している．コネクタはBezier曲線で表現することができ，Bezier曲線の制御点を修正することで，格子の直交性をある程度制御可能である．制御点の位置はコネクタ端点に隣接する表面格子の法線ベクトルを用いて決定している．このときの概念図を図 38に示す．制御点位置を決めるためのベクトル$\vec{C}$は以下の式で求めた．

$$\vec{C} = w_A\vec{A} + w_B\vec{B} \tag{1}$$

ここで，$\vec{A}$, $\vec{B}$はコネクタ端点に隣接する表面格子の法線ベクトル，w_A, w_Bはそれぞれのベクトルにかける重みである．具体的な の値はコネクタによって異なり，試行錯誤によって適切な値を求めた．ただし，フラップの舵角が大きくなると式(1)で求めた制御点位置は機体前方に大きく偏ってしまい，適切なコネクタにはならなかった．そのため，ベクトル$\vec{C}$とX軸の成す角がある角度以下にならないよう制限を設けることで，対応舵角範囲を広げた．実際のコネクタに適用すると図 39のようになる．これは内翼前縁フラップを大きく操舵させた場合の例であり，胴体側フラップ端から出るコネクタについて表している．



(a)　斜方視



(b)　下流視

図39　空間中のコネクタ修正

ブロックを囲むドメインの形状が複雑である場合には，ブロック内部で格子線が交差してしまうことがある．格子線が交差してしまうとヤコビアンの値が負になってしまい，CFD計算を行うことができない．AutoFlap-GGで対象としている機体形状の場合には，前縁フラップの舵角の組み合わせによって複雑な形状になってしまい，主翼上下面に接するブロック内部でヤコビアンが負になる場合がある．このような場合に対処するために，格子線が交差している辺りでブロックを分割し，分割面および分割後のブロックに対してTFIをかけることで，格子線の交差を解消している．この処理が不要である場合もあるため，ヤコビアンの値をチェックしてブロックの分割を行うかどうか事前に判断している．

トポロジーの関係から，フラップとフラップの間の隙間に対応するブロックは，舵角が大きくなると歪んでしまうが，同一のトポロジーで格子を生成するため，ここでは問題とはしていない．ただし，ヤコビアンの値は正になるよう調整した．

5.3　格子品質チェック

生成された格子に対して格子品質チェックとしてヤコビアンの値を調べる．これにはGridgenに搭載されている機能を利用している．CFD計算を行うためにはヤコビア

ンの値が正であることが必要である．全ての格子のヤコビアンが正である場合のみ，GUI画面上の「Save」ボタンが有効になり，次のステップに進むことができるようになる．ヤコビアンが0以下である場合には「Save」ボタンは無効のままであり，次に進むことはできず，格子を手動で修正する必要が生じる．格子品質チェックは「Execute」ボタンから形状・格子自動生成後に自動的に実行されるが，「Check Jacobian」ボタンによって単独で実行することもでき，この場合も全てのヤコビアンが正であれば「Save」ボタンは有効になる．

5.4　データ出力

CFD計算時の計算効率を高めるために，生成された格子に対しブロック・ドメインの結合，ブロックの分割を施し，CFD計算用の境界条件を設定して，データをファイルに出力する．ここまでで格子生成は終了しており，以降の処理では格子点の位置は変更せず，CFD計算を行う上で必要な処理のみを行っている．

5.4.1　ブロック・境界面の結合

生成された格子に対して，ブロックの結合および境界面を構成するドメインの結合を行う．格子修正が完了した時点では，ブロックは格子修正がしやすいように分割されているが，必ずしもCFD計算を効率良く行えるブロック分割とはなっていない．そのため，一旦ブロックをできるだけ結合し，単純なブロック構成に変えている．ドメインに対しても，物理境界面やブロック間境界面を余分に分割しないようできるだけ結合している．

結合するブロックはそれぞれグループに分けられており，UPPERグループとLOWERグループがある．UPPERグループは機体の上側，LOWERグループは機体の下側にそれぞれ位置するブロックが属する．UPPERグループに属するブロックを全て結合し，1個のブロックにしている．LOWERグループについても同様に1個のブロックに結合している．この段階でブロックは，機体上下の2個とフラップ間の隙間の6個，計8個になる．

ドメインについては，ドメイン数が可能な限り少なくなるように結合している．ブロック間境界のドメインについては，2個のブロックを任意に選び，各ブロックのフェイスを構成するドメインの中から2個のブロックで共通のものを選び，その中の2個のドメインを組み合わせて順次結合を試みている．これを全てのブロックの組について行っている．計算領域の境界面にあたるドメインについては，1個のブロックを任意に選び，そのブロックを構成するドメインの中から同じ境界条件が設定されているものを2個選び，結合を試みている．これを全てのブロックに対して行っている．

Gridgenでは，トポロジーなどの関係でブロックやドメインを結合できなかった場合にはエラーを出力するが，Glyph内で適切にエラー処理を行うことで，エラーによってGlyphが途中で止まることなく全てのブロック，ドメインに対して処理を行うことができる．

5.4.2　ブロック分割

ここでは，マルチ・ブロック対応の並列CFD解析ソルバーを用いて計算する場合に，計算効率が良くなる格子を得るためにブロックの分割を行う．ここで対象としているソルバーは，JAXAで開発されたUPACS（Unified Platform for Aerospace Computational Simulation）のように，各CPUにブロックを割り当てる方法で領域分割している並列ソルバーである．AutoFlap-GGでは，32個のCPUを用いて計算を行う際に，各CPUに割り当てられる格子点数が均等になるよう，ブロックを72個に分割する．格子点数，トポロジーは舵角によらず固定であるので，常に同じ分割方法で分割するようにコードに組み込んである．格子点数や使用するCPU数によって適切な分割方法が異なるため，これらを変更する際にはコードを変更する必要がある．

ブロックの分割位置は，インデックスの方向（I, J, K）とインデックスの値で指定している．さらに，分割したブロックをさらに分割する場合にも，一度に指定できるようにしている．これにより，分割したブロックのIDを取得する別のコードを記述する必要がなくなった．分割後のブロックのIDは，分割ルーチン内で取得，使用されており，コードを書き換える際には最初に与えるブロックIDと分割位置を示すインデックスのみを考えればよい．図 40にブロック分割を行うコードの例を，図 41にコード例を用いて分割したブロックの概念図を示す．コード例のindex_listは分割位置を示し，blk_idには分割するブロックのIDが入る．分割してできたブロックのIDはblk_idsに返される．このコード例では，まずブロックをK=21で分割し，次にインデックスの小さい側のブロックについてI=28とI=92で分割して，残りのブロックについてさらに分割を進めていく流れとなる．index_listの最初の要素は分割方向を示し，それ以降は分割位置，もしくはリストである．内部のリストも同様に分割方向と分割位置，リストを記述でき，入れ子にすることができる．

5.4.3　境界条件設定

ここでは，CFD解析ソルバーにあわせて境界条件の設定を行っている．現在対象としているCFD解析ソルバーはJAXA超音速機チームで開発されたADCS（AeroDynamic Computational System）であり，これに合わせて境界条件を設定している．条件を設定するドメインはあらかじめ

```
set index_list {-k {-i 28    92        } 21
                   {-i 28    92     145} 46
                   {-i 28 59 92 118 145} 138
                   {-i 28    92     145} 168
                   {-i       92     145}
               }
set blk_ids [entitySplit "-blk" $blk_id $index_list]
```

図40　ブロック分割位置指定のコード例

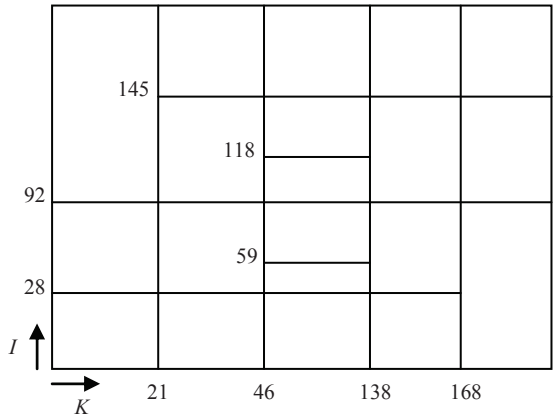


図41　ブロック分割コード例の実行結果

条件毎にグループに分けられている．他のソルバーに対応させるために境界条件を書きかえる場合には，コードを修正する必要がある．

5.4.4　ファイル出力

ここでは，生成した格子データ，境界データ等をファイルに出力する．出力するファイル名はパラメータファイルから読み取られる．出力されるファイルについては3.2節を参照すること．

境界条件ファイルについては，Gridgenの境界条件データ出力機能とともにADCS用の出力ルーチンを使用している．これはADCS用の境界条件IDの番号が4桁であり，Gridgenから出力される境界条件IDでは飛び飛びの値を設定することができないためである．そのため，一旦Gridgenから出力された境界条件ファイルについて，境界条件IDをADCS用の値に変換する処理を行っている．

表6　グループ名と対応する境界条件一覧

グループ名	境界条件名	ADCS用境界条件ID
wall	滑り無し壁	2001
sym	対称	3001
farfield	遠方	4002
inflow	流入	5002
outflow	流出	6002
pole	ポール(特異点)	7001

6.　まとめ

高揚力装置として前・後縁フラップを採用した超音速機翼胴形態に対して，フラップ操舵後の形状と計算格子の自動生成を行うツールAutoFlap-GGを開発した．市販格子生成ソフトGridgenを利用し，Gridgen独自のスクリプト言語であるGlyphを用いて，超音速機高揚力形態のフラップ操舵に伴い発生する，フラップ操舵前の形状・格子からの形状変更と格子修正の操作作業を自動化した．生成例，使用方法，処理内容について解説を行った．AutoFlap-GGを用いることで従来の手作業による格子生成に比べ，非常に短時間で格子を生成することができる．これにより，フラップ操舵後の形状と計算格子を素早く得ることができるようになり，CFD計算に注力することができる．また，多数の舵角に関する格子を容易に得ることができるため，パラメトリック・スタディを行いやすくなる．また，バッチ処理を行うこともできるため，最適化設計ループに組み込むことも可能となっている．

AutoFlap-GGは実用に耐えうるものとなってはいるが，その生成される格子の品質，ツールの汎用性については課題が残されている．AutoFlap-GGで生成される格子はCFD計算用であるが，この格子を用いて計算を行った結果を検証することで，格子品質の向上を図っていく．また，現在のVersionでは，JAXAジェット実験機01次形状の高揚力形態にのみ適応可能となっているが，処理を一般化し，他の形態についても迅速に対応できるよう改良を行っていく．

参考文献

1. 坂田，“超音速実験機について　－NAL次世代超音速機技術の研究開発―”，第36回飛行機シンポジウム講演集，1998.
2. 雷忠，“超音速機高揚力装置に関する数値解析及び考察”，宇宙航空研究開発機構報告，JAXA-RR-07-050.
3. Pointwise Inc., “Gridgen User Manual,” Version 15, 2006.

宇宙航空研究開発機構研究開発資料　JAXA-RM-09-003

発　　行　平成21年9月30日
編集・発行　宇宙航空研究開発機構
〒182-8522　東京都調布市深大寺東町7-44-1
URL: http://www.jaxa.jp/
印刷・製本　株式会社　実業公報社

本書及び内容についてのお問い合わせは、下記にお願いいたします。
宇宙航空研究開発機構　情報システム部　研究開発情報センター
〒305-8505　茨城県つくば市千現2-1-1
TEL：029-868-2079　FAX：029-868-2956