

CFD 共通基盤プログラム UPACS の現状と今後の課題

Common CFD Platform - UPACS : Status and Prospects

高木 亮治, 航技研, 調布市深大寺東町 7-44-1, ryo@nal.go.jp
山本一臣, 牧田, 光正, 榎本俊治, 山根敬, 向井純一, 山崎裕之 (航技研)
Ryoji Takaki, NAL, 7-44-1 Jindaiji-Higasi, Chofu
K. Yamamoto, M. Makida, S. Enomoto, T. Yamane, J. Mukai, H. Yamazaki (NAL)

UPACS, *Unified Platform for Aerospace Computational Simulation*, is a project to develop a common CFD platform since 1998 at National Aerospace Laboratory. This project firstly aims to overcome the increasing difficulties in recent CFD code programming on parallel computers which includes complex geometry problems and coupling with different types of simulations. Secondly it also aims to accelerate the development of CFD technology by sharing a common base code among research scientists and engineers. The development of UPACS for compressible flows with multi-block structured grids is completed. The current status and prospects of UPACS are described in this paper.

1. はじめに

計算流体力学 (CFD) の計算手法の発達とコンピュータ技術の急速な進歩により、高い計算精度が要求される航空宇宙分野においても CFD を用いた流れ場の解析が脚光を浴びようになってきた。それにともない CFD が CFD そのものための研究という側面から脱皮して、設計や流体现象の研究ツールとして様々な事を要求され、また期待されるようになった。中でも航空宇宙分野では航空機、宇宙往還機や推進機等の複雑形状への適用、乱流等の複雑な非定常流及び燃焼流、極超流速等の非平衡流の解析、構造や伝熱との連成解析、さらにそれらを組み合わせた最適設計等が求められている。このような要求に答えるべく、専門の研究者がそれぞれの研究目的に合わせて CFD プログラムが作成された結果細分化された分野毎に CFD プログラムが乱立する事となった。CFD への要求が高度化する中で、個々のプログラムはますます複雑になり、研究者個々人が独立に開発を行うことは非常に困難になってきた。しかも、基本的な数値解析手法には共通性があるにしても、それらを乱立した個々の複雑なプログラム間で共通的に利用することが困難となり、汎用化や CFD 技術の継承の著しい障害となっている。

一方、近年コンピュータは急速な進化を遂げた。CFD の発展はコンピュータの高速化に支えられてきたといっても過言ではない。ベクトル計算機により CFD の可能性が広がり、並列計算機の登場でさらに大規模計算が可能となり、CFD の実用化への道が開けた。ハードウェアの性能向上、半導体素子、ベクトルプロセッサ、クロスパーネットワーク等、確かに目を見張るものがある。しかしながらそれらを使いこなすだけのソフトウェア技術の進歩は遅々として進まず、その代表的なものとして並列計算技術がある。スーパーコンピュータの殆んどが今や並列計算機である事を考えれば、大規模計算を実施する上で並列化は避けて通れない問題である。しかしながらコンパイラによる自動並列化はまだまだ実用的とは言い難いレベルであり、並列計算機の性能を十分に活用するためにはプログラマ自身がプログラムの並列化を行う必要がある。そのためには並列処理、ハードウェアの知識が流体解析の知識以外に必要となってくる。このように決して負荷の低くない、しかも流体解析とは直接関係しない作業、更には CFD プログラムの構造がかなり似通っているためほとんど同じ作業を個々の研究者がそれぞれ独自に実施する事は非常に効率が悪いと言える。

以上のように CFD プログラムを開発するに当たって、要求の高度化及び並列化といった問題のため、開発すべきプログラムが複雑化している。そのためこういったプログラム開発上の困難を克服し、CFD プログラムにより柔軟な拡張性と汎用性を持たせる試みとして UPACS (Unified

Platform for Aerospace Computational Simulation) プロジェクト^{1, 2, 3)}を行っている。UPACS はマルチブロック構造格子を用いた圧縮性流体の数値シミュレーションを、分散/共有メモリマルチプロセッサマシンにおいて実行することができる共通の基盤コード (プラットフォーム) を確立することを目的としている。さらにこの基盤コードを核にして、従来同じような計算手法であるにもかかわらず、適用する問題や解析を実行する計算機環境が異なるために、その開発に多重投資を強いられていた状況を改善し、系統だった CFD コードの検証や計算実行におけるノウハウの蓄積および共有を効率的に実現できるようにすることを目標としている。これらの目的を実現するための鍵として「コードの階層化」と「データおよび計算手法のカプセル化」といったオブジェクト指向的な考え方を用いている。これらの手法を用いることで

- マルチブロック構造格子を並列化のための領域分割にマッピング
- 並列処理、マルチブロック処理をソルバーから分離
- ハードウェアへの依存性を押えることで移植性を確保
- ソースコードのモジュール化による共有の促進

を実現している。またソルバーを中核として解析に必要な前後処理ツールも併せて整備している。

2. UPACS の現状

UPACS は昨年、2000 年 10 月 2 日に version 1.00 がリリースされた。その後主にバグフィックスのバッチが配付された (version 1.01) 後、2001 年 6 月に新たな機能を追加した version 1.1 がリリースされた。version 1.1 ではバグフィックス (AUSMDV スキーム、Spalart-Allmaras 乱流モデル、その他)、Spalart-Allmaras 乱流モデルの非保存系から保存系への変更が行われている。更に新機能として、ターボ機械等の解析が行えるように回転座標系への対応を行った⁴⁾。

2.1 動作環境

UPACS はこれまでのプログラミング資産からの移行を容易にしつつ、並列処理部の分離、コードの共有化を実現するため Fortran90 を採用した。また並列処理には MPI⁵⁾ を採用することで、ハードウェアへの依存性を可能な限り排除した。そのため大型計算機からパソコンまで多くの計算機環境で実行できるようになった。表 1 に動作が確認された計算機を示す。パソコン (PC-UNIX) において

は現状で動作確認が取れたフリーの Fortran90 コンパイラは存在しないが、市販のコンパイラでの動作確認は取れている。また MPI に関してはフリーの mpich⁶⁾, LAM⁷⁾での動作が確認されている。UPACS の実行環境は UNIX 環境を前提としており、現在パソコンで多く使われている Windows には対応していない。

	NWT、SGI Origin
スパコン	富士通 VPP シリーズ NEC SX シリーズ
WS	Sun, SGI, Compaq
パソコン	Linux (Intel 及び Alpha)

UPACS ではパッケージのインストールを容易にするためオープンソフトで使われる configure を用意している。configure を使うことでコンパイルに必要な設定、つまり makefile の自動生成を行っている。configure はコンパイルに必要なコマンド、ツール、ライブラリの検索、確認を行うと同時に Fortran90 コンパイラの同定を行う。Fortran90 コンパイラの同定に失敗した場合はとりえず makefile が生成されるが特に Fortran90 コンパイラのオプションが正しく設定されないので手動で設定してやる必要がある。また configure は UPACS のインストール環境を切替えるためのオプションを受け付ける。主なものとして、インストール場所の指定、X11 用グラフィックス及び MPI の使用/不使用、最適化等である。UPACS のインストールは基本的に

```
upacs% configure
upacs% make
upacs% make install
```

で行われる。

2.2 複雑形状への対応と並列処理

UPASC はマルチブロックに分割された計算対象を並列処理する。一つのプロセッサに複数のブロックを割り当てるかを任意に指定できる。ブロック間通信は UPACS で共通的なモジュールとして提供しており、プロセッサ間通信およびプロセッサ内通信をユーザーは意識する必要がない。今回新しく重合格子への対応を行った。ここではマルチブロック格子群が重なった状況を想定している。その場合、マルチブロック間接合及び重合格子間のデータ通信が必要となるが、重なったブロック間の通信は従来の通信モジュールを拡張する形で実装しており、ユーザーは特に意識する必要はない。重合格子法に関しては文献⁸⁾があるのでそちらを参考にさせていただきたい。

2.3 前処理ツール

UPACS では本体の CFD ソルバーだけでなく、解析を実行する際に必要なツールが用意されている。マルチブロック格子の接合情報を生成するツールとして *createConnect3D* が用意されている。*createConnect3D* はマルチブロック格子のそれぞれの格子データを読み込んで各ブロック間の接合状態を解析してそれらの情報を connect.txt に出力する。また物理境界条件の設定ファイルとして bc.txt が用意されているが、未定義の境界があればそれらを bc.txt に UNDEFINED(未定義) と出力される。ユーザーは bc.txt に出力された UNDEFINED の部分を編集することで境界条件の設定を行うことができる。図 1、図 2、に bc.txt 及び connect.txt の例を示す。図からわかるようにそれぞれのファイルには単なる数字の羅

列ではなく、ある程度読んで意味がわかるよう具体的に記述されている。

重合格子の補間インデックスを作成するツールとして *createOversetIndex* が用意されている。マルチブロック格子群を読み込みそれらの包含関係を解析して、包含関係にあるセルのインデックスの作成及び物理量の補間係数を出力する。

他にも並列計算時の静的負荷分散を図る(各プロセッサへのブロックの割り当てを決定する)ツールとして *createPEAssignment*、流体中の物体に作用する力(圧力、摩擦力)を積分していわゆる揚力、抵抗、モーメントを計算する *calcAeroCharact*、UPACS の入出力ファイルのフォーマット (fub) を PLOT3D フォーマットへ変換する *upacsToPlot3d* 等が用意されている。入力データ編集ツールとして *pdbEditor* も用意されているが、これについては後で詳しく述べる。

2.4 パラメータデータベース

UPACS では計算条件やコードの動作制御などのパラメータを、タグとともに管理するパラメータデータベース (pdb:Parameter Data Base) をモジュールとして構築した。解析問題や、手法などにより、異ってくるパラメータを、プログラム中の任意の箇所でタグによってデータベースに登録、参照することができ、コードの共通性と拡張性が高まった。PDB への入力はテキストファイルで行われ、UPACS の場合は input.txt となっている。通常のテキストファイルであるため一般のエディターで編集することができる。PDB の入力ファイルフォーマットは非常に単純であり、タグ名とその値を併記するだけである。区切り記号として =、:、' (スペース) が使える。図 3 に入力ファイル (input.txt) 例、図 4 にプログラム内における PDB の使用例を示す。pdbRead で入力ファイルを

```
equation_type : Euler
convect_flux = roe
flow_ref_Mach 0.0
restart = .false.
```

図 3: 入力ファイル (input.txt) 例

```
call pdbRead('input.txt')
...
etype = pdbStr('equation_type')
scheme = pdbStr('convect_flux','roe')
mach_ref = pdbReal8('flow_ref_Mach')
if(pdbLogical('restart',.false.)) then
...
```

図 4: プログラムでの利用例

指定すると、そのファイルに記述されたタグとその値が PDB の中に読み込まれる。するとプログラムの任意の場所で pdbStr, pdbReal8, pdbLogical にタグ名を指定してコールするとその値を得ることができる。それぞれのサブルーチンの最初の引数がタグ名である。二番目の引数はオプションで、データベースに、つまり入力ファイル(この場合は input.txt) にタグが記述されていない場合のデフォルトの値も指定できる。PDB モジュールにはより汎用的な pdbGet という関数も用意されている。この関数を使えばタグの型(整数型、文字型、論理型、浮動小数点型等)を気にしないで呼び出すことができる。

```
# ID Block Face Window start(i,j,k) end(i,j,k) bcType
 1 1 I_min 1 1 1 1 21 21 ENTRY_SUBSONIC
 2 1 I_max 91 1 1 91 21 21 EXIT_SUBSONIC
 3 1 J_min 1 1 1 91 1 21 WALL
 4 1 J_max 1 21 1 91 21 21 WALL
 5 1 K_min 1 1 1 91 21 1 WALL_SLIP
 6 1 K_max 1 1 21 91 21 21 WALL_SLIP
```

図 1: 物理境界条件設定ファイル (bc.txt) 例

```
# ID connect-type
# Src. Block-ID Window start(i,j,k) end(i,j,k) Face
# Dst. Block-ID Window start(i,j,k) end(i,j,k) Face
# Dst. Block-Axis connected to (i,j,k)-axis of Src. Block
 1 INTER_BLOCK
      1 1 1 1 13 21 I_min
      2 46 1 1 46 13 21 I_max
      I J K
 2 INTER_BLOCK
      1 1 13 1 19 13 21 J_max
      4 1 19 1 1 1 21 I_min
      -J I K
 3 PERIODIC X-ROTATION UPPER_SIDE
      2 19 13 1 46 13 21 J_max
      5 28 13 1 1 13 21 J_max
      -I -J K
```

図 2: 接合情報ファイル (connect.txt) 例

2.4.1 pdbEditor

pdb(パラメータデータベース)ではパラメータ間の依存関係、値の適切な範囲といった事は管理していないため、これを援助し、適切な入力パラメータファイルの作成を手助けするGUIツールとして *pdbEditor* を開発した。パラメータ間の依存関係、値の適切な範囲といった付加情報は、パラメータの作成者が定義ファイルに記述する。*pdbEditor*はこの定義ファイルを用いてユーザーに対して適切なパラメータの入力を支援する。図5にPDBの定義ファイル例を示す。図5に示したように、各パラメータはグルーピングすることができる。この例では *equation* と *condition* の二つのグループが設定されており、*pdbEditor*ではそれぞれ図6、図7のように表示される。入力パラメータの型にしたがって選択型に対してはラジオボタンが、論理型に対してはトグルボタンで表示される。整数、実数に対しては入力フィールド及びスライドバーが表示される。この場合、値の範囲も指定できる。図5の *cfl* の場合はデフォルト値が1で0以上、*aoa* の場合初期値は0.0で-180から180までのスライドバーが表示される。また、パラメータ同士の依存関係も記述できる。図5の *vis_model* は、*equation_model == Euler* の場合は入力の必要がないことを *disable* で記述している。図6(b)には *equation_model == Euler* を選択した場合の画面が表示されている。入力の必要がない *vis_mode* はグレー表示になり、入力できないようになっている。

PDB及び *pdbEditor* は汎用的に設計されており、UP-ACS以外のコードでもこのモジュールを組み込み、定義ファイルを生成することで使用可能である。

3. UPACS ユーザー会

UPACSをより多くの人に利用してもらい、CFDを核にしたシミュレーション技術の普及を促進するためにUP-

```
group:equation {
  message:equation_type {equation}
  choice:equation_type = NS {Euler NS RANS}

  message:vis_model {viscous model}
  disable:vis_model {equation_type == Euler}
  choice:vis_model = full {full thin}
}

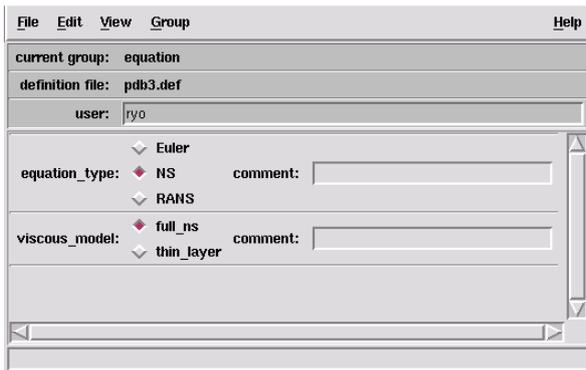
group:condition {
  message:restart {restart ?}
  logical:restart=.false.

  message:cfl {CFL number}
  real:cfl=1.0 {1 0}

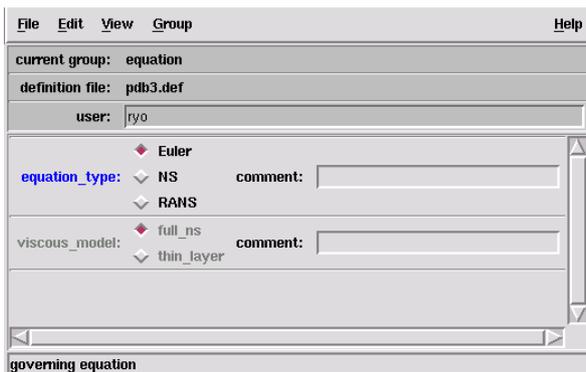
  message:aoa {attack angle}
  real:aoa=0.0 {1 -180 180}
}
```

図 5: PDB 定義ファイル例

ACSユーザー会を発足させた。現段階では学術目的に限定しているが、ユーザー会に入会することでUPACSのソースプログラムを無償で利用することができる。ユーザー会は現在会員数が31名でその内訳は、大学19名、民間7名、国研その他6名となっている(航技研以外の会員、2001年12月現在)。UPACSの最新情報やUPACSユーザー会に関する問い合わせは「航空宇宙技術研究所インターネットホームページ」<http://www.nal.go.jp/cfd/jpn/upacs/>で提供しているのでそちらを参照していただきたい。も



(a) equation_type = NS を選択したところ



(b) equation_type = Euler を選択したところ

図 6: pdbEditor 画面 (equation group)

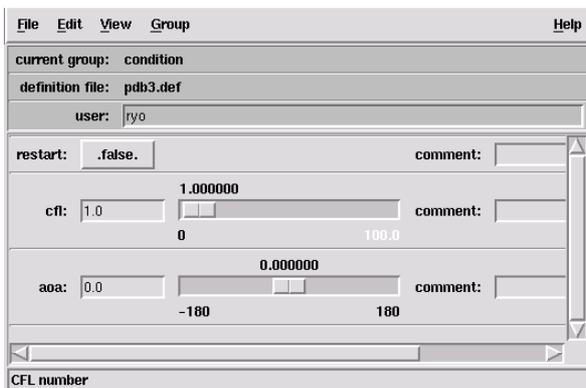
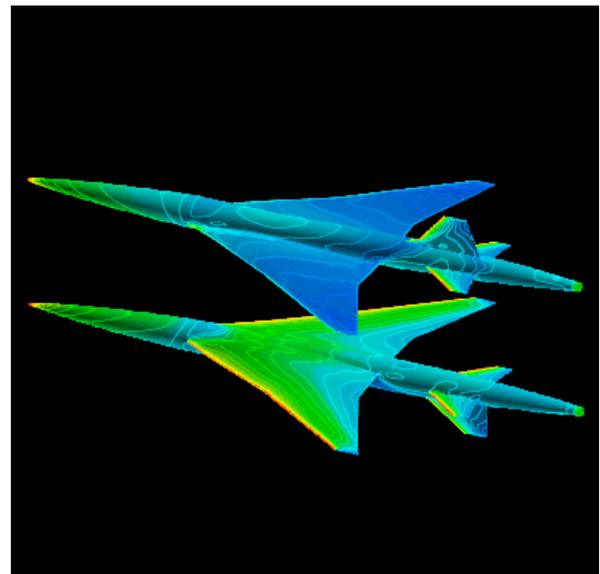


図 7: pdbEditor 画面 (condition group)

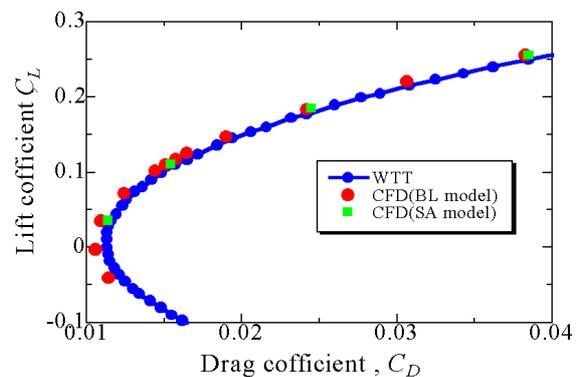
しくは upacs@nal.go.jp まで問い合わせさせていただきたい。

4. UPACS の計算例

UPACS を用いた計算例について紹介する。現在航技研で開発が行われている小型超音速実験機 (NEXST-1) まわりの超音速流の解析結果を示す。図 8(a) はマッハ数 2、迎角 2 度で飛行する NEXST-1 の表面圧力分布を示している。図の赤い領域および青い領域がそれぞれ圧力の高い領域、低い領域を示している。ブロック数は 88、計算格子点数は約 450 万点である。この計算では Spalart-Allmaras の乱流モデル (SA モデル) を用い全域乱流の条件で計算している。図 8(b) に揚抗特性を風洞実験と比較した図を



(a) 表面圧力分布



(b) 揚抗力特性

図 8: NEXST-1 まわりの流れの解析

示す。この図では同時に SA モデルと Baldwin-Lomax 乱流モデルの差をも示している。このように NEXST-1 の開発に置いては種々の風洞データが蓄積されておりそれらのデータを用いて UPACS の検証を実施している。特に飛行実験に向けて格子依存性の確認を行い、飛行解析に用いる標準格子の策定を行った。また乱流モデルの違いによる影響なども検証している。これら検証結果はソースコード、実験データとともにワークショップ、検証データベースを通じて広く公開していく予定である。

図 9 は ONERA-M5 周りの遷音速流の解析例である。マッハ数は 0.84、迎角は -1.0 度で、レイノルズ数は 6×10^7 、全域乱流の計算である。ブロック数が 95 ブロック、格子点数は約 500 万点である。

図 10 は遷音速圧縮機動翼 NASA Rotor35 の流れの解析例で、動翼前方に存在する動翼ディスクとセンタボディの隙間、およびセンタボディ内部の空間を考慮した計算で、漏れ流れが性能に及ぼす影響を調べたものである。計算条件は圧力比が 1.96、チップマッハ数が 1.34、ブロック数は 27、格子点数は約 150 万点である。図 10(a) は表面圧力分布と翼端およびハブ面隙間の流れを示す流線を、図 10(b) は翼間のエントロピ分布 (高い領域のみ) およびハブ面隙間からの流線を示す。また図 11(a) 及び図 11(b)

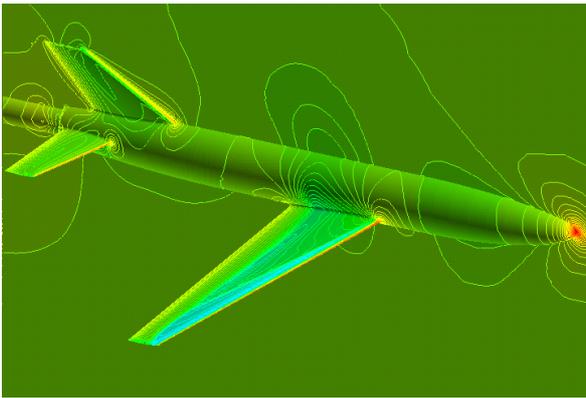
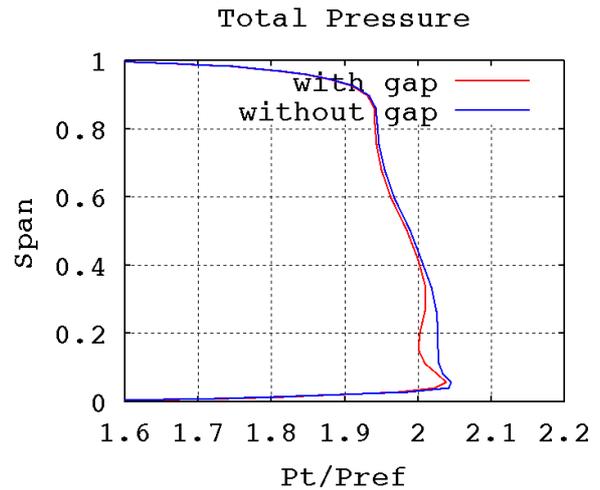
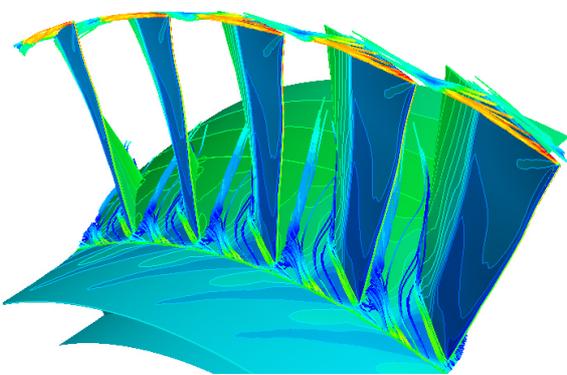


図 9: ONERA-M5 の圧力分布

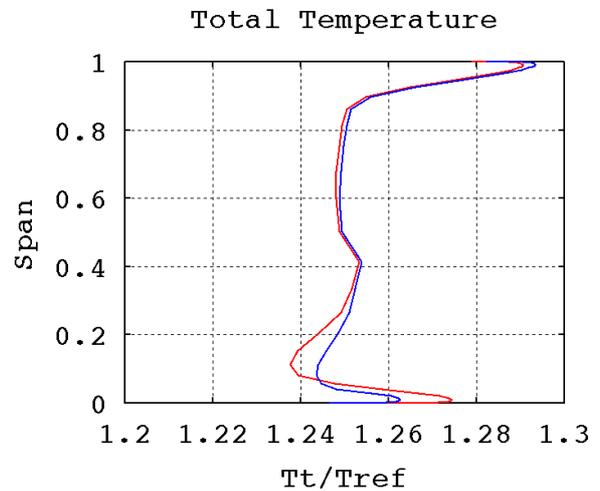
は翼後縁より 180% コード下流位置における全圧および全温のスパン方向分布を示している。



(a) 全圧

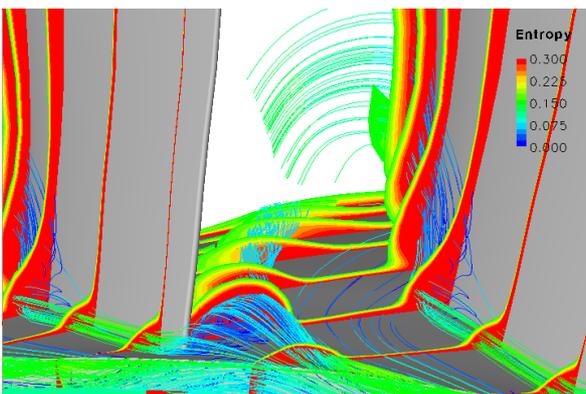


(a) 表面静圧力分布と流線



(b) 全温

図 11: 下流位置でのスパン方向分布



(b) 翼間エントロピ分布と流線

図 10: 遷音速圧縮機動翼まわりの流れ

5. UPACS の今後

UPACS をより使いやすく、また多くの問題に適用できるようにするため、境界条件 (物理的、重合境界、その他) を総合的に設定するための GUI ツールの整備、各種計算スキームの移植、連成問題 (構造、化学反応、伝熱解析) への拡張、非構造格子法の利用等更なる改良と機能の追加、各種ツールの整備を予定している。

5.1 境界条件編集ツール

解析対象の物体形状が複雑になれば、それらを反映するマルチブロック計算格子のブロック数も飛躍的に増大する。そのためそれらの計算格子の壁、流入、流出、遠方といった物理的境界条件の設定も困難になってくる。ブロック間の接合情報は *createConnect3d* で自動的に生成されるが、物理的境界条件は解析実行者が手動で設定してやらなくてはならない。ブロック数が多くなったり、物体形状が複雑になったりするとこの作業は非常に面倒となり、また設定ミスを犯しやすくなる。また重合格子の場合、3 つ以上格子が重なった場合、補間データの受渡しを人間が必ず指示してやる必要がある。その場合格子形状はもとより、重合部分は極めて複雑な形状となり、解析実行者がそれらを入力してやるのは非常に困難となる。そのためそういった設定作業を支援するツールとして GUI を持った境界条件編集ツールの開発を行っている。設定すべき境界及びその形状、位置をグラフィカルに表示し、マウス操作でそれらの選択、境界条件等の設定を行う。このツールを用いることで複雑な境界条件等の入力を容易に行えるようになる。

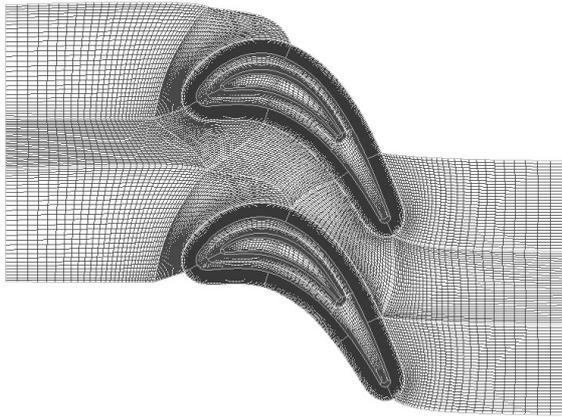


図 12: 熱伝導との連成用計算格子

5.2 連成問題

CFD を用いた解析で今後重要となるのは様々な問題と連成した統合シミュレーションだと言われている。UPACS においてもそういった機能が求められているが、UPACS はその構造上そういった連成問題に対する柔軟性を保有している。つまり、マルチブロック処理/並列通信部とブロック内ソルバーが分離しており、各ブロック毎に別々のソルバーを使用する事はそれほど困難なことではない。そのため流体と熱伝導の連成解析の開発を行っている。図 12 に熱伝導との連成解析用の計算格子を示す。

5.3 UPACS の検証

様々な機能を追加し、使いやすくしていくことは重要であるが、それ以上に UPACS の信頼性を確保することも重要なことと考えている。特に UPACS の場合ユーザー会を通じて広く公開しており、コードの信頼性を確保することが強く求められるし、同時にそうできる立場にあると考えている。ユーザーが様々な問題に UPACS を適用することで、様々な問題が浮き彫りとなり、それに対処して行くことでより良いコードへと進化していくと期待される。また同時に様々な問題に対する解析結果が蓄積されていく事も期待される。航技研 UPACS チームとしても今後 UPACS を対象に系統だった検証作業を行い、解析データの積み上げを行う予定である。こういった検証作業、データの積み上げにおいては新しいデータ記述言語である XML を用いた検証データベース⁹⁾との連携、航技研で開催している CFD on Web と連携していきたいと考える。このように検証データの積み上げを行うことで今後 UPACS の信頼性、ひいては CFD 技術全体の信頼性を確保したいと考えている。

6. おわりに

UPACS の現状と今後の展開について紹介した。特に UPACS の現状に関しては今まで紹介されていなかったインストール方法、前処理ツール等、UPACS を使った解析を実行する上で必要となる事に関して紹介した。現時点では圧縮性流体の解析がとりあえず行えるようになり、実際の適用例も少しずつ報告されるようになった。今後は更なる改良と機能の追加、各種ツールの整備を行うと同時に検証計算を積み上げる事で UPACS の信頼性を確保したいと考えている。

参考文献

- 1) 山本一臣、他 (航技研 UPACS プロジェクトチーム). 並列計算 CFD プラットフォーム UPACS について. In *Proceedings of Aerospace Numerical Simulation Symposium '99*, pp. 351-355. NAL SP-44, February 1999.
- 2) 山根敬、他 (航技研 UPACS プロジェクトチーム). CFD コード共有化プロジェクト UPACS の現状. In *Proceedings of Aerospace Numerical Simulation Symposium 2000*, pp. 45-50. NAL SP-46, December 2000.
- 3) 高木亮治、他 (航技研 UPACS プロジェクトチーム). UPACS の現状と今後の課題. In *Proceedings of Aerospace Numerical Simulation Symposium 2001*. NAL SP 出版準備中, 2001.
- 4) 向井純一, 山本一臣, 山根敬, 野崎理. UPACS によるエンジン内部流れの計算. In *Proceedings of Aerospace Numerical Simulation Symposium 2001*. NAL SP 出版準備中, 2001.
- 5) The message passing interface (mpi) standard. <http://www-unix.mcs.anl.gov/mpi/>.
- 6) MPICH-A Portable Implementation of MPI. <http://www.mcs.anl.gov/mpi/mpich/>.
- 7) LAM/MPI Parallel Computing. <http://www.lam-mpi.org/>.
- 8) 牧田光正, 高木亮治. UPACS の重合格子対応方法と適用例. In *Proceedings of Aerospace Numerical Simulation Symposium 2001*. NAL SP 出版準備中, 2001.
- 9) 高木亮治, 大西柁平. XML を用いたデータベースの構築. In *Proceedings of Aerospace Numerical Simulation Symposium 2001*. NAL SP 出版準備中, 2001.