# 航技研汎用ソフトウェア資産の利用について

中村絹代*、山崎裕之*、廣瀬直喜*、吉田正廣*、岩宮敏幸*

# Practical Use of General Numerical Simulation Software Property of NAL

by

K. NAKAMURA, H. YAMAZAKI, N. HIROSE, M. YOSHIDA, T. IWAMIYA

## ABSTRACT

In the National Aerospace Laboratory (NAL), many programs have been developed, and the programs which fulfill certain conditions are registered as general-purpose programs, and are used within NAL. These programs are unusable by a new system, unless porting work is done appropriately whenever a computer system is replaced. Moreover, in order to aim at effective use of those programs, it is required to offer a graphical user interface (GUI) environment that is easy to use for a user. This report describes examination of the execution environment for realizing software porting and user-friendly utilization of those programs, and presents its example.

## 1．はじめに

航空宇宙技術研究所（以下航技研という）では開発されたプログラムを所内利用プログラムとして登録し、利用に供することとしているが、計算機の性能向上を考えると、従来、大型計算機を前提として開発されたプログラムも、現在ではワークステーション（WS）上で稼動させた方が利用者にとって大きなメリットが得られる場合がある。また、実験計画の立案や空力設計等の作業のように、必ずしも計算の専門家でない人がプログラムを利用することが今後ますます増えていくことが予想されるため、グラフィカル・ユーザー・インターフェース（GUI)を充実させ、より使いやすい環境を同時に提供することが必要である。

本研究は、航技研が開発したプログラム成果の共用促進のため、既存のプログラムの入出力を見直し、GUI の整備を行い、ユーザーにとって使いやすい格子生成プログラムや空力解析プログラムを製作することを目的とする。

また、所内利用プログラムとして登録されたプログラムは計算機システムがリプレースされるたびに適切に移植作業が行われないと、新しいシステムで活用することができないので、移植作業をスムーズに進めるための方法についての検討も必要である。

本報告では、所内利用プログラムの移植およびユーザーフレンドリーな利用方法を実現するための実行環境の検討及びその例について報告する。

## 2．対象プログラム

表１は主な所内登録プログラムを一覧表にしたものである。２次元問題／３次元問題、また、ポテンシャル、オイラー、ナビエ・ストークス、ポアソン方程式、差分法、境界要素法といった手法、計算法も直接法、SLOR(Succesive Line Over Relaxation)法、IAF(Implicit Approximate Factorization)法など多岐にわたる。これらの所内利用プログラムはオリジナルプログラム、ベクトル計算機向きチューニングプログラム等含めて数十本ある。

これらのプログラムのVP2100システム及びSun Enterprize4500システムにおける実行時間を図１に示す。VP2100システムは大型計算機システムとして利用されてきたシステムであり、後者はSun SystemのWSである。図より大型計算機を前提として開発されたプログラムも現在ではWSで充分実行可能であることがわかる。

＊　航空宇宙技術研究所

表1　主な所内登録プログラム

| プログラム名 | プログラム内容 |
|---|---|
| AFMESH | 遷音速翼型解析のための格子形成法 |
| FPAWING | 遷音速3次元翼解析 |
| FPWING | 遷音速翼周り完全ポテンシャル流の数値解析 |
| INVERSE | 2次元遷音速翼型設計 |
| MAFIA | 3次元・軸対称2次元静磁場解析 |
| NSFOIL | 2次元遷音速粘性流解析 |
| PNCCP | 2次元遷音速流解析 |
| PQCCP | 2次元遷音速流解析 |
| TSFOIL | 2次元遷音速微小擾乱解析 |
| YOKUDOG | 翼胴結合体まわり非粘性圧縮流の数値解析用格子形成コード |
| YOKUDOP | 翼胴結合体まわり完全ポテンシャル流の数値解析 |
| YOKUDOE | 翼胴結合体まわり流れのＥｕｌｅｒ解析 |
| ZENKIPNL | 亜音速3次元揚力物体の空力計算 |
| ZENKIUNS | 境界要素法による全機形状の非定常空力計算 |
| ZENKSDSP | 境界要素法による全機形状の空力計算 |

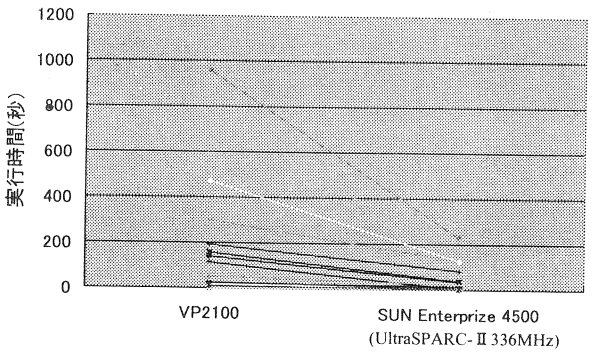


図1　所内登録プログラムの実行時間

また、所内利用プログラムの他システムへの移植を通して、移植を阻害する要因をいくつか結果として得たので、述べておく。

① 実行において入力となるデータの形式はバイナリデータよりテキスト形式のデータの方が扱いやすい。移植先のシステムにおける検証作業においても扱いやすい。

② 科学技術計算ライブラリ及び組込ルーチンの利用がある場合は、プログラム名及び機能を把握しておくことにより、移植がスムーズに行なえる。

③ コンパイラー特有の機能を利用した処理は移植先のコンパイラーがその機能を持たない場合もあるので、標準的でない機能の利用は避ける。

これらの要因はなるべく回避できるように、差し支えなければ標準的でない機能の利用を避けることを勧める。また、こういった内容をプログラムの情報として残しておくことを勧める。

本報告で述べるWSで活用対象とするプログラムは、遷音速翼型解析のための格子形成法プログラム「AFMESH」及び2次元遷音速翼粘性流解析プログラム「NSFOIL」である。これらのプログラムは2次元問題解析プログラムであり、比較的小規模なプログラムであるが、所内利用プログラムとして登録されたプログラムの中では、プログラム使用手引書、プログラム解説書、多くの使用例等が揃っており、内容が最も充実しているものの部類に入るものである。

（1）AFMESHプログラム概要

・遷音速翼型解析のための格子形成法プログラムである。

・幾何学的構成法、連立非線型偏微分方程式の差分解による構成法（ポアソン／ラプラス方程式）、後方領域構成法を組み合わせることにより、利用者の要求を満たす翼付近の格子網を発生させるプログラムである。

・手法、計算法は差分法、SLOR法である。

（2）NSFOILプログラムの概要

・翼型まわり高レイノルズ数遷音速粘性流の数値解析プログラムである。

・時間平均2次元ナビエ・ストークス方程式に2層代数乱流モデルを取り入れている。

・手法、計算法は差分法、IAF 法である。

## 3．ＧＵＩ設計

GUI の整備を行うにあたり、利便性を考慮して実行環境を整えるために、以下の点を考慮して進めることとした。

① 開発者等の専門家が利用するために、細かい入力設定や出力が行える。

② 使用ソフトウェアをよく知らない専門家でない人も入力設定などを理解しやすく、簡単に行え、利用できる。

③ 計算機が高速化され、可視化技術が進み、利便性が進められた事により可能となったことを、機能として盛り込む。

3．1　GUI 環境の実装

GUI 環境の実装の方法として、次のようなものが考えられる。

①アプリケーションとしてのGUI 実行システム

GUI システムを通常のアプリケーションとしてインストールした計算機にログインして実行する形式をいう。 X Window System のアプリケーションであるので、ネットワーク経由で他の計算機から実行する場合は通常通りの使用方法でよい。GUI システムがインストールされていない他の計算機で実行したい場合は、「AFMESH」，「NSFOIL」がインストールできること、およびGUI システムに必要なソフトウエアがインストールできることが必要である。

②ブラウザを使用した実行システム

ソルバーを実行する計算機で CGI（ Common Gateway Interface)を動かし、ブラウザを通して実行する方法である。

GUI 環境の実装を①の方法を用いて行なう。理由は以下のとおりである。利用ソフトウェアの知識がない人も実行可能である、すなわち、ソルバーの知識がなくても利用可能とするためには、ブラウザを使用することによる制約がユーザー支援を行なうということを難しくすると考えられるからである。①の方法を試みた後、②の方法ではどの程度可能かを検討する方がよいと考えられる。

また、他のGUI 環境の例としてUPACS(Unified Platform Aerospace Computational Simulation) があげられる。UPACS では、pdbEditor (parameter database editor) でパラメータの設定を行っている。UPACS の前処理を図２に示す。図中、定義ファイルは入力させる項目の名前、入力の形式（数値、文字列、選択）、項目間の依存関係を指定するもので、ソルバーに機能追加した場合などは必要に応じてその定義ファイルへの追加・変更を行う。pdbEditor は、定義ファイルを読み、GUI 画面を構成し、ユーザーからの入力に応じてパラメータファイルを生成する。ソルバーはそのパラメータファイルを読み、指定されたスキームやパラメータの値を用いて計算を行う。

UPACS ツールは前処理のみであるが、本GUI 環境では前処理だけでなく後処理も行う。これを図３に示す。前処理については、対象とするソルバーを変更可能とするためにはソルバーの入力形式を限定することができない。Fortran のnamelist記述文が生成できればよいというわけではない。

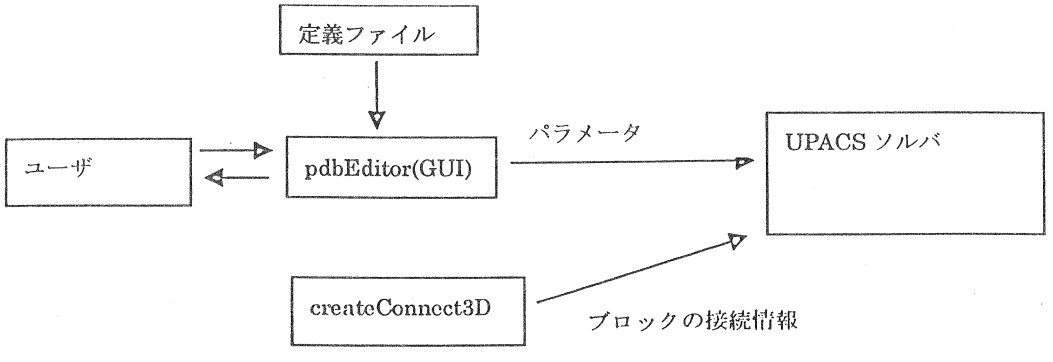


図２　UPACS の前処理

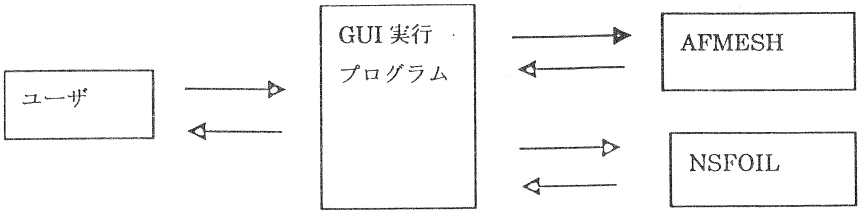


図３　GUI 環境

たとえば、翼型指定などの入力形式もソルバー毎に異なり、個別に対応する必要があるものが考えられ、単なる入力変数の並びだけでは済まない。

このように、pdbEditor のような定義ファイルを設計するだけでは対応できないと考えられるので、前処理部のGUI の構成をプログラム言語で記述することとした。言語を用いることで次のような利点がある。

UPACS ツールのpdbEditor に対応する部分（パラメータなどの入力）と、ジョブ実行や後処理など他の部分を単一のプログラムとして構築することができるので、過去に実行したジョブのパラメータを入力部に呼び出すなどの連携した処理を行ないやすい。また、一つの入力項目（マッハ数、最大反復回数など）あたり2,3 行程度で画面の構築が実現可能である。

### 3．2　使用言語

Ruby言語を用いて作成することとした。Rubyの特徴およびGUI 環境に用いる利点を以下に示す。

・オブジェクト指向言語として設計された言語である。オブジェクト指向言語としての継承、差分プログラミングなどの概念によって、機能の追加、変更を行うことができる。
・簡易なスクリプト言語であり、コンパイルが不要である。変数の宣言、ポインタ、メモリ管理の必要がない。
・インタプリタのため速度は遅いが、GUI 環境としてはあまり問題にはならないと考えられる。C 言語などで記述したライブラリを動的にリンクする（拡張ライブラリと呼ぶ）ことにより速度を速くすることもできる。
・拡張ライブラリは約束事が少ないので、簡単に記述することができる。また、既存のライブラリを結合することもできる。画面表示は後述のGtk グラフィックライブラリをRubyから使うためにRuby/Gtk拡張ライブラリを用いる。
・Rubyで記述された種々のライブラリ、拡張ライブラリが存在するので、利用の幅が広い。たとえば、oracle, postgreSQLデータベースアクセス、XML パーサ、OpenGL 等。

以上の点からRuby言語でGUI 環境を記述することとした。また、再利用可能なコードは共通化してライブラリとして提供することとした。

### 3．3　グラフィックライブラリ

UNIXワークステーション上でGUI 実行環境を実現するという点から、X Window System に対応したグラフィックライブラリが必要である。X Window System 上でグラフィカルなアプリケーションを開発する際には、最も基本となるライブラリXlibを直接使用することはほとんどなく、通常Xlibをベースとしたライブラリを使用する。ここでは、Gtk+グラフィックライブラリを用いることとした。Gtk+は、X Window System 上で用いられるライブラリの中で最もよく使用されているものの一つであり、次のような特徴がある。

・GNU LGPLライセンスに従うため本アプリケーションの配布に問題がない。
・オープンソースであり、開発する上で情報が得やすい。よく利用されているため安定している。
・ソースコードの他に特別なファイルなどを使わないため特別な開発環境を必要としない。C コンパイラがあればよい。
・C 上のライブラリであるが、他の言語からも使いやすい設計になっている。Perl、Rubyなどいろいろな言語から使用できる。
・オブジェクト指向の考え方でライブラリが設計されているので習得し易い。ボタンやテキスト入力領域などのGUI 部品がクラス継承により定義されている。

### 3．4　実行環境

GUI を実行するためには以下のソフトウェアが必要である。

①Rubyインタプリタ

Web から入手可能である。現在の安定版である1.6 系バージョンを使用する。Linux であればほとんどのディストリビューションではパッケージとして含まれている。

②Gtk+

Web から入手可能である。Linux であればほとんどのディストリビューションではパッケージとして含まれている。

③GUI プログラムソース

Rubyが必要である。

④GUI プログラム用拡張ライブラリ

C で記述されているのでC 言語コンパイラが必要である。

3．1　GUI 環境の実装

GUI 環境の実装の方法として、次のようなものが考えられる。

①アプリケーションとしてのGUI 実行システム

GUI システムを通常のアプリケーションとしてインストールした計算機にログインして実行する形式をいう。 X Window System のアプリケーションであるので、ネットワーク経由で他の計算機から実行する場合は通常通りの使用方法でよい。GUI システムがインストールされていない他の計算機で実行したい場合は、「AFMESH」，「NSFOIL」がインストールできること、およびGUI システムに必要なソフトウエアがインストールできることが必要である。

②ブラウザを使用した実行システム

ソルバーを実行する計算機で CGI（ Common Gateway Interface)を動かし、ブラウザを通して実行する方法である。

GUI 環境の実装を①の方法を用いて行なう。理由は以下のとおりである。利用ソフトウェアの知識がない人も実行可能である、すなわち、ソルバーの知識がなくても利用可能とするためには、ブラウザを使用することによる制約がユーザー支援を行なうということを難しくすると考えられるからである。①の方法を試みた後、②の方法ではどの程度可能かを検討する方がよいと考えられる。

また、他のGUI 環境の例としてUPACS(Unified Platform Aerospace Computational Simulation) があげられる。UPACS では、pdbEditor (parameter database editor) でパラメータの設定を行っている。UPACS の前処理を図２に示す。図中、定義ファイルは入力させる項目の名前、入力の形式（数値、文字列、選択）、項目間の依存関係を指定するもので、ソルバーに機能追加した場合などは必要に応じてその定義ファイルへの追加・変更を行う。pdbEditor は、定義ファイルを読み、GUI 画面を構成し、ユーザーからの入力に応じてパラメータファイルを生成する。ソルバーはそのパラメータファイルを読み、指定されたスキームやパラメータの値を用いて計算を行う。

UPACS ツールは前処理のみであるが、本GUI 環境では前処理だけでなく後処理も行う。これを図３に示す。前処理については、対象とするソルバーを変更可能とするためにはソルバーの入力形式を限定することができない。Fortran のnamelist記述文が生成できればよいというわけではない。

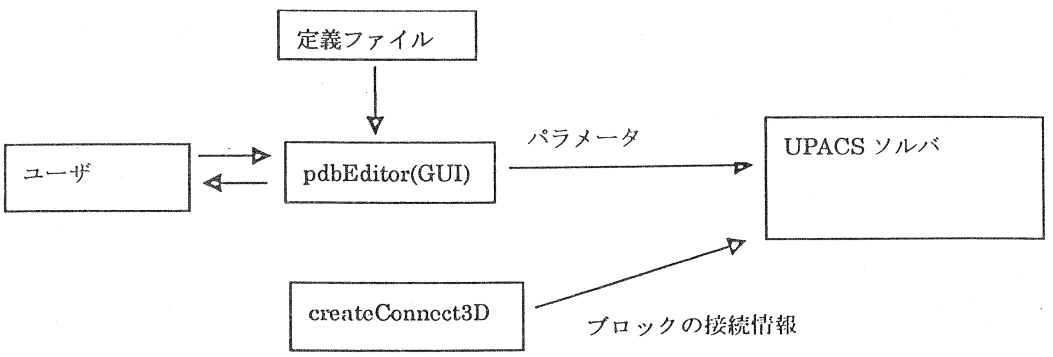


図２　UPACS の前処理

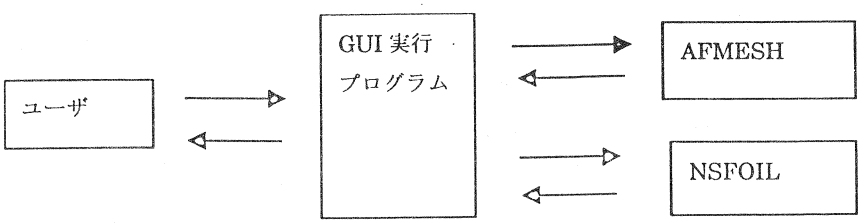


図３　GUI 環境

たとえば、翼型指定などの入力形式もソルバー毎に異なり、個別に対応する必要があるものが考えられ、単なる入力変数の並びだけでは済まない。

このように、pdbEditor のような定義ファイルを設計するだけでは対応できないと考えられるので、前処理部のGUI の構成をプログラム言語で記述することとした。言語を用いることで次のような利点がある。

UPACS ツールのpdbEditor に対応する部分（パラメータなどの入力）と、ジョブ実行や後処理など他の部分を単一のプログラムとして構築することができるので、過去に実行したジョブのパラメータを入力部に呼び出すなどの連携した処理を行ないやすい。また、一つの入力項目（マッハ数、最大反復回数など）あたり2,3 行程度で画面の構築が実現可能である。

### 3．2　使用言語

Ruby言語を用いて作成することとした。Rubyの特徴およびGUI 環境に用いる利点を以下に示す。

・オブジェクト指向言語として設計された言語である。オブジェクト指向言語としての継承、差分プログラミングなどの概念によって、機能の追加、変更を行うことができる。
・簡易なスクリプト言語であり、コンパイルが不要である。変数の宣言、ポインタ、メモリ管理の必要がない。
・インタプリタのため速度は遅いが、GUI 環境としてはあまり問題にはならないと考えられる。C 言語などで記述したライブラリを動的にリンクする（拡張ライブラリと呼ぶ）ことにより速度を速くすることもできる。
・拡張ライブラリは約束事が少ないので、簡単に記述することができる。また、既存のライブラリを結合することもできる。画面表示は後述のGtk グラフィックライブラリをRubyから使うためにRuby/Gtk拡張ライブラリを用いる。
・Rubyで記述された種々のライブラリ、拡張ライブラリが存在するので、利用の幅が広い。たとえば、oracle, postgreSQLデータベースアクセス、XML パーサ、OpenGL 等。

以上の点からRuby言語でGUI 環境を記述することとした。また、再利用可能なコードは共通化してライブラリとして提供することとした。

### 3．3　グラフィックライブラリ

UNIXワークステーション上でGUI 実行環境を実現するという点から、X Window System に対応したグラフィックライブラリが必要である。X Window System 上でグラフィカルなアプリケーションを開発する際には、最も基本となるライブラリXlibを直接使用することはほとんどなく、通常Xlibをベースとしたライブラリを使用する。ここでは、Gtk+グラフィックライブラリを用いることとした。Gtk+は、X Window System 上で用いられるライブラリの中で最もよく使用されているものの一つであり、次のような特徴がある。

・GNU LGPLライセンスに従うため本アプリケーションの配布に問題がない。
・オープンソースであり、開発する上で情報が得やすい。よく利用されているため安定している。
・ソースコードの他に特別なファイルなどを使わないため特別な開発環境を必要としない。C コンパイラがあればよい。
・C 上のライブラリであるが、他の言語からも使いやすい設計になっている。Perl、Rubyなどいろいろな言語から使用できる。
・オブジェクト指向の考え方でライブラリが設計されているので習得し易い。ボタンやテキスト入力領域などのGUI 部品がクラス継承により定義されている。

### 3．4　実行環境

GUI を実行するためには以下のソフトウェアが必要である。

①Rubyインタプリタ

Web から入手可能である。現在の安定版である1.6 系バージョンを使用する。Linux であればほとんどのディストリビューションではパッケージとして含まれている。

②Gtk+

Web から入手可能である。Linux であればほとんどのディストリビューションではパッケージとして含まれている。

③GUI プログラムソース

Rubyが必要である。

④GUI プログラム用拡張ライブラリ

C で記述されているのでC 言語コンパイラが必要である。

⑤ソルバーソースコード

Fortran コンパイラが必要である。GNU g77 ではコンパイルできない。Linux 環境では富士通Fortran が利用可能である。ただし、富士通Fortran は、新しいC ライブラリ glibc 2.2 系ではなく、glibc 2.1 系が必要である。

現在、「AFMESH」及び「NSFOIL」は、Alps( hp kayaku xu800 7/1000,Linux) システムで利用できるがexcelsior (SunEnter-prize4500,SunOS5.6)，kaidou(SunOS5.4)システムでも上記のプログラムがインストールされていることが確認されており、利用可能と考えられる。

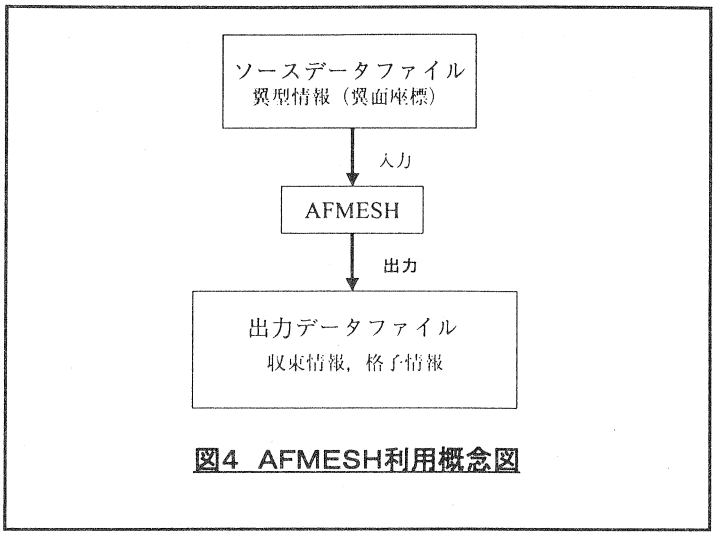


図4 AFMESH利用概念図

3．5 AFMESH-GUI

GUI 環境の整備はAFMESHおよびNSFOILプログラムの２本についてプログラムの入出力の見直しを行い、GUI プログラム作成によりユーザーのパラメータ入力や出力結果を扱い易いものに整えたが、紙面の関係で、本報告ではAFMESH-GUIについて説明を行なう。

図４にAFMESHプログラムの利用概念図を示す。また、図５にAFMESH-GUIの構成図を示す。図５の黒枠内がGUI 部分であり、その外側はAFMESHプログラムの構成（図４に相当）である。GUI 部分のユーザーインターフェースを用いてデフォルト値及び入力値からインプットファイルを作成する。GUI で作成されたインプットファイルはソルバーに入力され、格子生成が行なわれ、格子生成が終了すると、GUI はユーザーが選択した結果をユーザーインターフェースに出力する。このように入力データファイル作成及び出力結果表示に関してAFMESHプログラム利用時のユーザーの負担の軽減をはかっている。

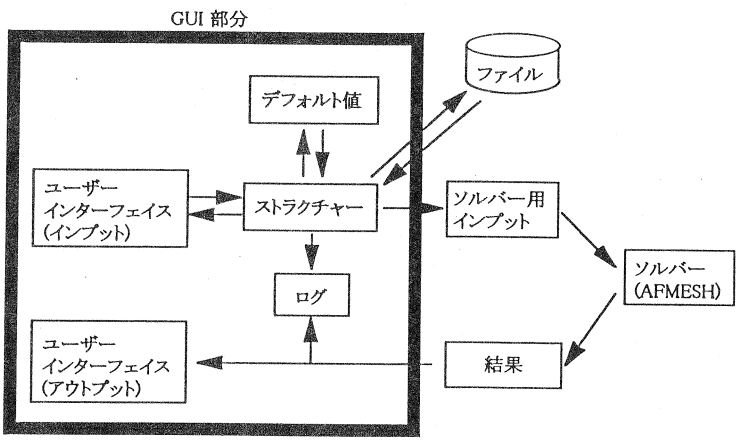


図５ ＡＦＭＥＳＨ－ＧＵＩ構成図

AFMESHプログラムは単独翼用、風洞用、翼列用（コンプレッサー用、タービン用）の格子を生成することができる。これらの入力パラメータは対象とする格子毎に多少異なること及びデフォルト値も異なることから、対象とする格子を指定することにより各々の格子のデフォルト値が設定され、利用者は最低限必要な入力パラメータを設定しAFMESHを利用することができる。これにより約100 変数の入力パラメータを簡単に利用する場合には10個程度の入力で利用できるようにした。

図６～図８にAFMESHプログラムをGUI 環境で利用した時の表示例を示す。図６において左半分は入力パラメータ設定部である。予めデフォルト値が設定されており、ユーザ

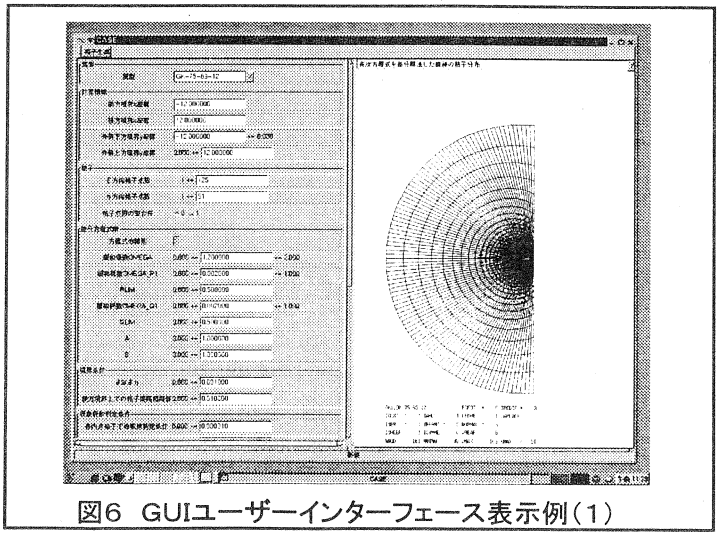

図6 GUIユーザーインターフェース表示例(1)

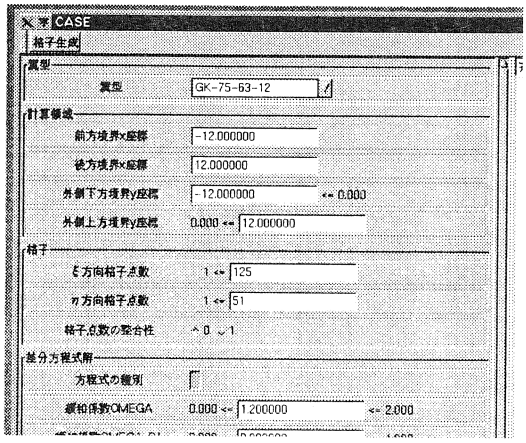


図7 GUIユーザーインターフェース表示例(2)

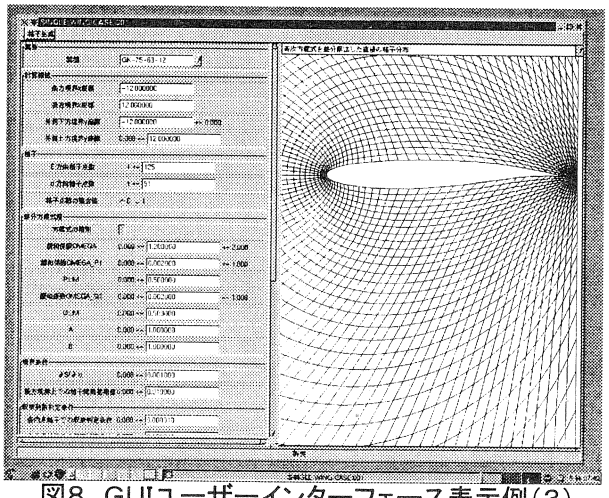

図8 GUIユーザーインターフェース表示例(3)

ーは適宜、設定変更できる。右半分は計算結果の格子網である。図7は、図6の入力設定部を拡大表示したものである。簡単にはここに示す計算領域、格子点数などの10変数程度の入力で実行できる。更に詳細入力を行なう場合は画面を変えて入力を行なう。図8は図6の計算結果の翼型を拡大表示したものである。

この他、NSFOILについてはパラメトリックスタディを考慮して複数の実行条件での結果をログにとっておき、これらの実行結果から、迎角毎のCL、CDなどの作図出力などを行なうことができる。

## 4．おわりに

航技研の開発したプログラム成果の共用促進を目的として、所内利用プログラムとして登録された 2本のプログラムのAFMESHプログラム及びNSFOILプログラムのUNIXシステムへの移植およびGUI 整備を行なった。プロトタイプであり、本プログラムの利用促進を行いながら、利用者の要望をフィードバックし、改良を重ねて利用しやすいものにレベルアップさせることを今後の課題と考えている。

最後に、ソフトウェア製作にあたりご協力をいただいた㈱計算力学研究センターの児玉勇司氏に感謝の意を表します。

参考文献

・航空宇宙技術研究所汎用コードAFMESH使用手引書、廣瀬直樹他、航空宇宙技術研究所、1987年10月
・航空宇宙技術研究所汎用コードNSFOIL使用手引書、河合伸担、廣瀬直樹他、航空宇宙技術研究所、1987年7 月
・オブジェクト指向スクリプト言語Ruby.
まつもとゆきひろ／石塚圭樹共著、アスキー出版、1999年10月