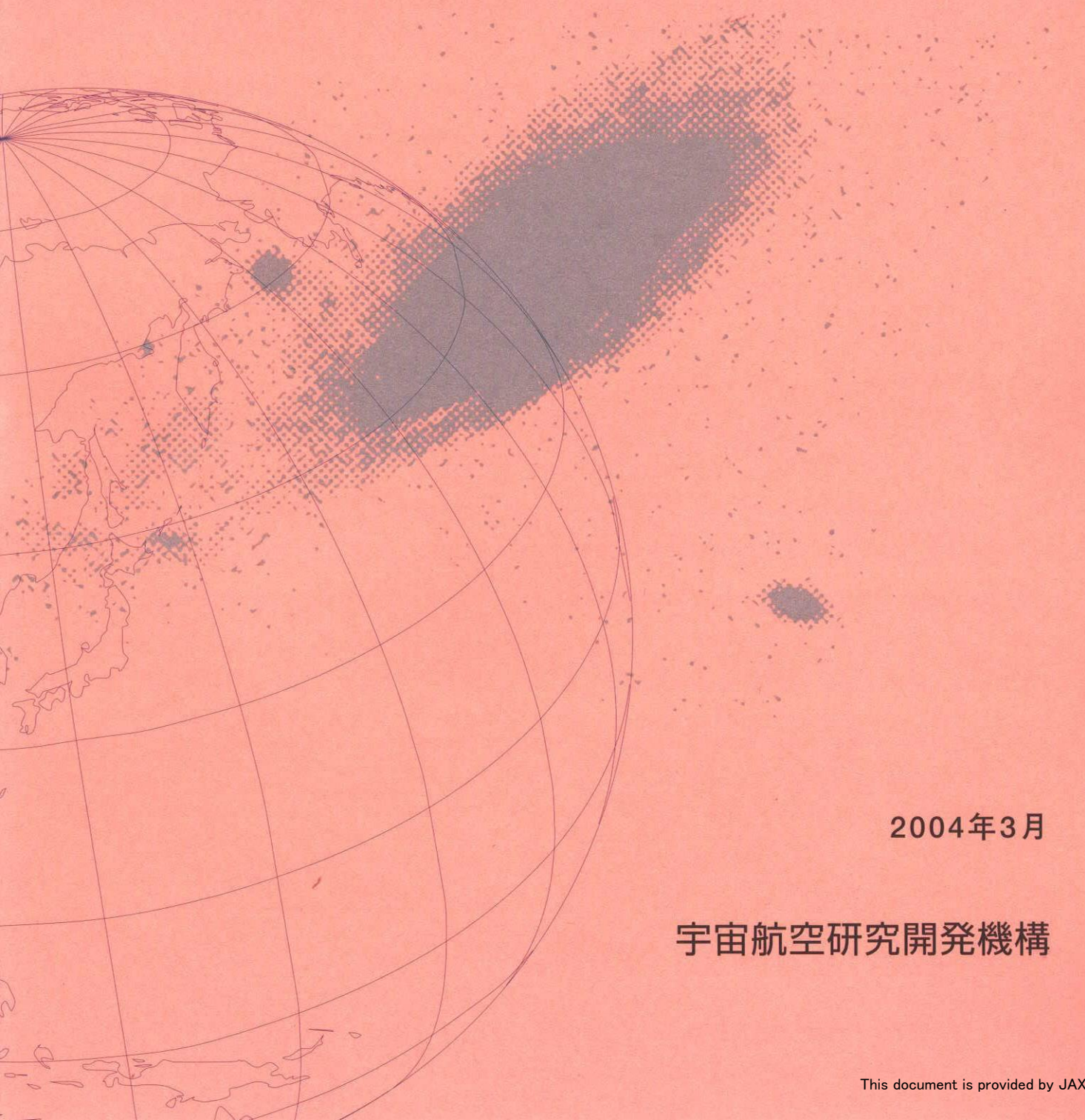


宇宙航空研究開発機構研究開発資料

数学モデル・データ生成システム (CIRAM)



2004年3月

宇宙航空研究開発機構

宇宙航空研究開発機構研究開発資料
JAXA Research and Development Memorandum

数学モデル・データ生成システム (CIRAM)

Construction, Inspection and Reconstruction of Aircraft Model (CIRAM)

岡田 典秋*1、穂積 弘毅*2

Noriaki OKADA, Koki HOZUMI

- *1: 総合技術研究本部 航空安全技術開発センター
Air Safety Technology Center
Institute of Space Technology and Aeronautics
- *2: 総合技術研究本部 飛行試験技術開発センター
Flight Test and Simulation Technology Center
Institute of Space Technology and Aeronautics

2004年3月

March 2004

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

目次

1. はじめに	1
2. 開発および実行環境	2
3. 変数名の定義	2
4. 標準 CIRAM フォーマット	2
5. 機能と使用法	3
5.1 警告パネルと確認パネル	3
5.2 起動と初期画面	4
5.3 FILE プルダウンメニュー	5
5.4 プロパティ設定 (PROPERTY)	6
5.5 グラフ出力仕様の設定 (GRAPH)	6
5.6 フォートラン関数文の生成 (FUNCTION)	9
5.7 CIRAM データの合成 (COMPOSE)	9
6. 拡張 CIRAM フォーマット	13
7. バッチ処理の方法	15
8. おわりに	15
付録 A CIRAM の実行例	16
付録 B フォートラン関数文に含まれるサブルーチン	19

数学モデル・データ生成システム(CIRAM)*

岡田典秋*1、穂積弘毅*2

Construction, Inspection and Reconstruction of Aircraft Model (CIRAM)*

Noriaki OKADA*1 and Koki HOZUMI*2

Abstract

The following software packages related with flight tests and flight simulation tests, which have been developed for experimental powered-lift STOL aircraft ASKA, have been improved:

- CIRAM (Construction, Inspection and Reconstruction of Aircraft Model)
- DHS (Data Handling System)
- BAC (Basic Aircraft Characteristics)

The main functions of CIRAM are to create the Fortran function statement from numeric data provided by the tabular format. Other functions include displays, correction, printing functions of the graph, etc.. This report describes the functions and usage of CIRAM

Keywords : software system, mathematical model

概 要

宇宙航空研究開発機構では、飛行実験及び飛行シミュレーション試験を実施する上で必要なソフトウェア・システムを低騒音STOL実験機「飛鳥プロジェクト」以来整備、運用してきた。ソフトウェア・システムは数学モデル・データ生成システム (CIRAM)、基本飛行特性解析システム (BAC)、飛行実験データ処理システム (DHS) 等から構成される。CIRAMは航空機の数学モデルを作成するソフトウェア・システムであり、風洞試験等によって表形式で与えられる数値データから、空力特性等を算出する補間機能を有するフォートラン関数文を生成する機能の他に、グラフの表示、修正、印刷機能等を有する。本報告書では、CIRAMの機能と使用方法について記述する。

1. はじめに

宇宙航空研究開発機構では、飛行実験及び飛行シミュレーション試験を実施する上で必要なソフトウェア・システム (ARTISAN Systemsと称している、図1.1) が低騒音STOL実験機「飛鳥」プロジェクトの過程で開発され、その後各種の飛行実験及び飛行シミュレーション試験において運用されてきた。ARTISAN Systemsは数学モデル・データ生成システム (CIRAM : Construction, Inspection and Reconstruction of Aircraft Model)、基本飛行特性解析システム (BAC : Basic Aircraft Characteristics)、飛行実験データ処理システム (DHS : Data Handling System) 等から構成されている。開発当時の計算機のOSはGUI機能を含んでいなかったため、これらソフトウェア・システムのユーザ・インタフェース部はキャラクタベースのディスプレイであった。その後、UNIXの普及に伴い操作性向上を目的として、ユーザ・

インタフェース部のGUI化を図ると同時に各ソフトウェア・システムの機能増強を行ってきた。

航空機的设计・開発過程においては、風洞試験結果やエンジンのカスタマーズ・デックを基に多くの計算・解析が繰り返される。設計の進捗・変更に伴って、種々のデータを迅速に評価すると共に、計算・解析に供されなければならない。CIRAMは実験終了からそのデータを計算・解析担当者へ供給する時間を極力短縮することを主眼として開発されたソフトウェア・システムである。

CIRAMは、指定されたフォーマット (CIRAMフォーマット) で記述された3次元までのデータファイル (テキスト形式) に対し、以下の処理を行うことができるソフトウェア・システムである。

- 補間機能を有するフォートラン関数文を生成する。
- ディスプレイ上にグラフを表示する。
- ディスプレイ上でグラフを修正する。

* 平成15年6月20日 受付 (received 20 June, 2003)

*1 総合技術研究本部 航空安全技術開発センター (Air Safety Technology Center, Institute of Space Technology and Aeronautics)

*2 総合技術研究本部 飛行試験技術開発センター (Flight Test and Simulation Technology Center, Institute of Space Technology and Aeronautics)

- ・グラフ及び数値テーブルを印刷する。
- ・2つのCIRAM データを合成する。
- ・グラフをPostScriptファイルで保存する。

特に、フォートラン関数文の生成がCIRAMの特徴的な機能である。更に、グラフをPostScriptファイルで出力する機能を有するため、パソコンの描画ソフトを用いてグラフを編集できるので、報告書の挿入図や発表用スライドの作成、データ・ブックの編集に便利である。また、上記の機能は対話型で実行できると同時に、グラフの修正機能を除いてはバッチ・モードによる一括処理が可能である。CIRAMは対話型とバッチ処理型のいずれかで実行できるが、対話型の場合には、マウスによるボタンのクリックかドラッグ、もしくはキーボードからの入力によって所望の指示を与えることができる。

本報告書は、現状におけるCIRAMの機能について使用方法を中心に述べたものである。

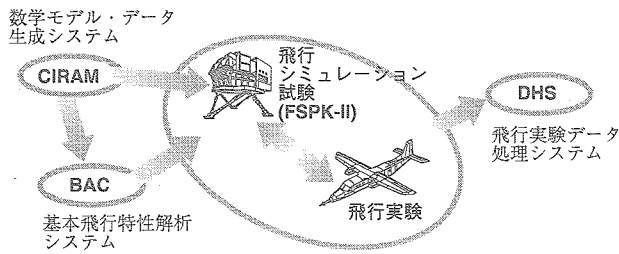


図1.1 ARTISAN Systems

2. 開発及び実行環境

CIRMAの開発は、日本サン・マイクロシステムズ株式会社製のワークステーションを用いて行った。OSはUNIX SVR4をベースとしたSolaris2.Xであり、開発言語はC及びFOTRAN 77を用いた。X Window SystemによるGUI環境は、他機種への移植性を考慮し、現在UNIXで標準となっているMotifを採用した。また、グラフのディスプレイ上への表示はフリーソフトであるghostviewを用いている。

CIRAMを実行させるために必要なハードウェア構成は、

- ・Solaris 2.3以上が稼働する計算機
- ・PostScript対応プリンタ

であり、環境変数として以下の設定が必要である。

PATH : CIRAMの実行ファイルが存在するディレクトリを指定

LD_LIBRARY_PATH : CIRAMに必要な動的ライブラリが存在するディレクトリを指定

また、CIRAMはグラフをPostScriptファイルで出力する機能を有している。PostScriptファイルはAdobe

Illustrator 1.1フォーマットであり、PostScript対応の描画ソフトで編集可能であるとともに、PostScript対応のビューア（ghostview等）やプリンタで出力可能である。

3. 変数名の定義

CIRAMは3次元までの多次元データを取り扱うことができる。生成されるフォートラン関数文においては、引数の左から順に第1変数、第2変数、第3変数と呼ばれる。グラフとして出力された場合には、図3.1に示すように軸変数（Axis Parameter）、線変数（Line Parameter）、図変数（Graph Parameter）と呼ばれる。特別な指定がない限り、第1変数＝軸変数、第2変数＝線変数、第3変数＝図変数の組合せが標準である。

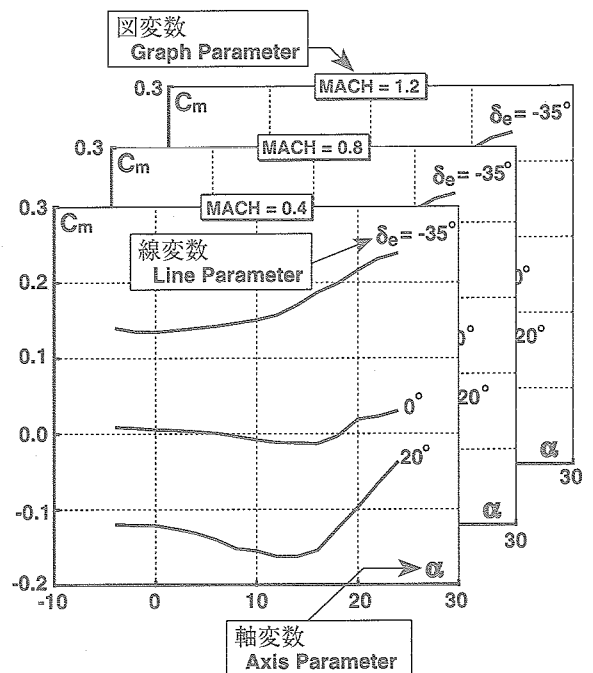


図3.1 変数名の定義

4. 標準CIRAMフォーマット

CIRAMを利用するためには、定められた書式に基づいて作成されたデータファイルを用意する必要がある。その書式には標準CIRAMフォーマットと拡張CIRAMフォーマットの二つがある。拡張CIRAMフォーマットは標準CIRAMフォーマットを拡張したものであり、詳細は第6章で記述される。

標準CIRAMフォーマットはコメント部とそれに続くデータ構成情報及びデータ部からなり、1次元から3次元までの標準CIRAMフォーマット構成を図4.1に示す。

コメント部は2行からなり、任意のコメントを記述できるが、以下の制約条件がある。

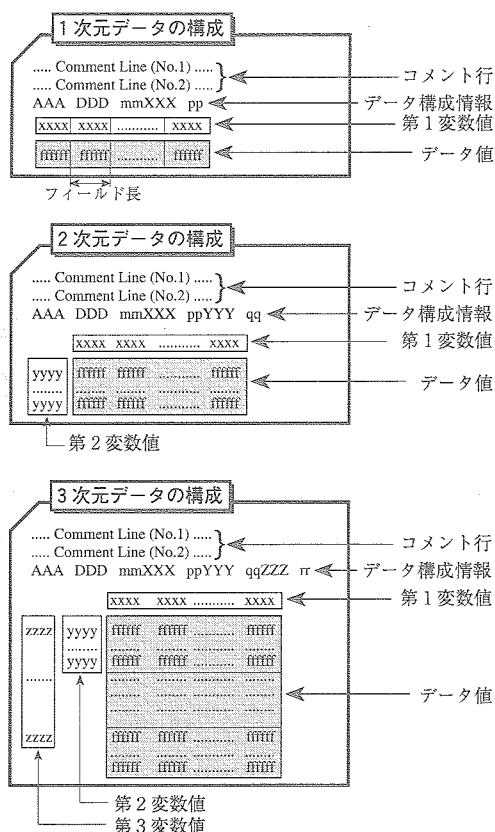
- ・各行最大文字数は68である。

- ・最初の文字は英数字に限られ、特殊文字は使用できない。
- ・空白行は無視される。

データ構成情報部は1行からなり、機体名、データ名及び変数名等を定義する。これらの名称は、フォートラン関数文の生成、グラフ出力時に使われる。

データ部はデータ構成情報部のフィールド長で指定されたコラム数で記述される。フィールド長が指定されていない場合は標準値の6が適用される。3次元データの場合、各第3変数値において第1変数値及び第2変数値は同一でなければならない。

CIRAMのデータファイルの記述例を付録Aに示す。



データ構成情報部詳細			
	コラム	内容	位置
AAA	1~8	機体名	左詰め
	9~10	空白	
DDD	11~17	データ名	左詰め
	18	空白	
mm	19~20	フィールド長	右詰め
XXX	21~28	第1変数名	左詰め
	29~30	第1変数名の個数	右詰め
YYY	31~38	第2変数名	左詰め
	39~40	第2変数名の個数	右詰め
ZZZ	41~48	第3変数名	左詰め
	49~50	第3変数名の個数	右詰め

注) 機体名、データ名及び変数名の第1文字は英文字であること。第1文字以外は英数字で特殊文字は使用できない。

図4.1 標準CIRAMフォーマットの構成

5. 機能と使用法

CIRAMは対話型とバッチ処理型何れかで実行できるが、本章は対話型での機能と使用法について述べる。なお、バッチ処理型については第7章で記述する。対話型の場合には、ユーザインタフェース部がGUI化されているため、マウスによる左ボタンのクリックかドラッグ、もしくはキーボードからの入力によって所望の指示を与えることができる。

以下、CIRAMの機能及び使用法の詳細について述べる。

5.1 警告パネルと確認パネル

CIRAM操作中に誤った指示等を与えた場合、警告パネルまたは確認パネルが表示される。警告パネル及び確認パネルの一例をそれぞれ図5.1, 図5.2に示す。

図5.1はデータファイル読み込み時に誤ってCIRAMのデータファイルでないファイルを指定した場合に表示される警告パネルであり、パネルに記載された事項を確認した後、[Confirm] ボタンをクリックし、操作をやり直すことができる。

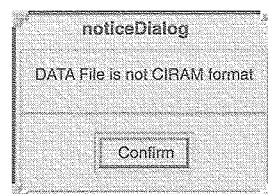


図5.1 警告パネルの一例 (データファイル読込時)

図5.2はCIRAM終了時に表示される確認パネルであり、本当に終了する場合は[QUIT] ボタン、終了しない場合は[CANCEL] ボタンをクリックする。確認パネルのボタン名称は確認内容に応じて異なる。

警告パネル及び確認パネルに関しては、それに表示されるコメントによって内容が理解できるので、ここではその詳細については省略する。

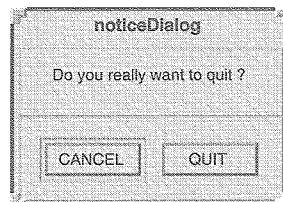


図5.2 確認パネルの一例 (CIRAM終了時)

5.2 起動と初期画面

第2章で述べたCIRAMの環境変数が正しく設定されていれば、以下のコマンド・ラインの入力により、CIRAMが起動され初期画面（図5.3）が表示される。

% ciram &

または

% ciram <DataFileName> &

ここで%はプロンプトであり、<DataFileName>は標準または拡張CIRAMフォーマットで作成されたデータファイルの名称である。&はバックグラウンドジョブとして実行させるために付けてあり必須ではない。ciramの代わりにciram-mで起動すると「グラフ修正画面」（図5.10）をひとまわり小さく表示する。ノート・パソコン等の小さな表示画面上でCIRAMを利用する場合に便利である。

コマンド・ラインからの入力時にデータファイル名を指定しなかった場合の初期画面を図5.3(a)に示す。データファイルのオープンには [FILE] をクリックし表示されるFILEプルダウンメニュー内のopenまたはopen..を選択して行う（5.3節参照）。

データファイルがロードされた後の初期画面（図5.3(b)）は、上端に配置されたCIRAMの種々の機能を指定実行する5つのボタン（FILE, PROPERTY, GRAPH, FUNCTION, COMPOSE）、データファイル名の表示または入力するフィールド（File name）、データを選択するフィールド（Data number）及びコメント部とデータ構成情報の表示領域で構成される。データファイル中に複数個のデータが納められている場合のデータ選択は、Data numberのフィールドにデータ番号を入力しリターンキーを押すか、フィールドの右にあるスライダーを動かすことによって行われる。データ番号を変更するとそれに応じてコメント部とデータ構成情報の表示領域が書き換えられる。図5.3(b)におけるデータ構成情報の表示内容は付録Aの付図A1に示されたデータファイルを読み込んだものである。

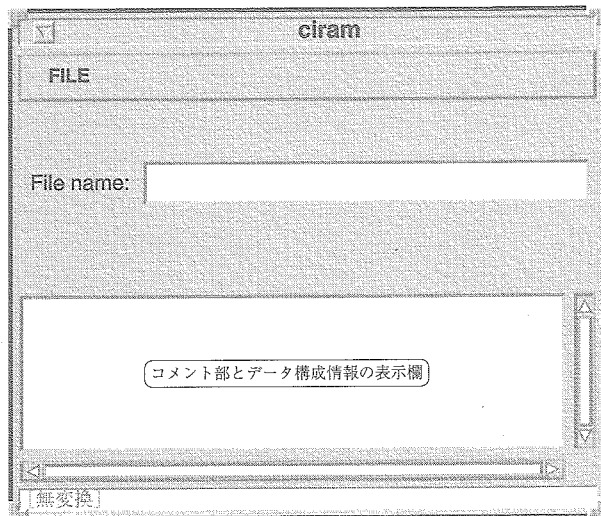
CIRAMデータファイルの記述内容に誤りがあった場合、データファイル読込時に図5.1に示す警告画面が表示されるとともに誤りの推定内容と該当箇所を示す行番号がCIRAMを起動したウインドウ内に表示される。例えば、2行必要なコメント部が1行しかなかった場合、下記のエラーメッセージが表示され、読込が中断される。

```
***Error : VehicleName FunctionName
Param1Name ...must write line.
```

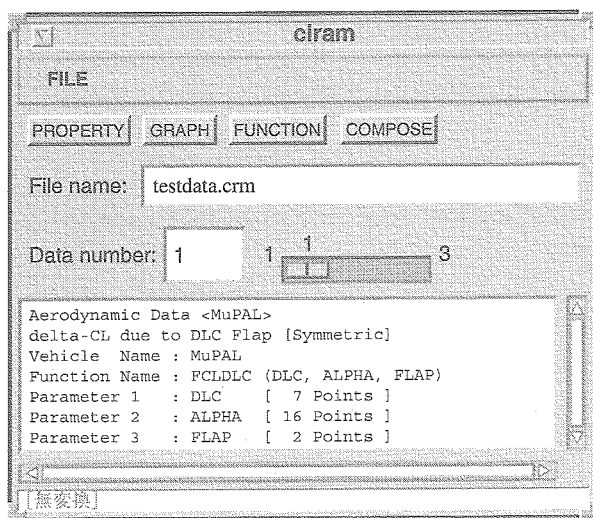
```
Line 21: -30 -20 -10 0 10 20 25
```

必ずしも表示されるエラー内容と該当箇所を示す行番号が実際とは一致しないが、誤りの発見と修正の手助けとなる。

以下、初期画面の上端に表示される5つのボタンの機能の詳細について述べる。



(a) ファイル名を指定しないで起動した場合



画面上的指示とその意味

File name	データファイル名の記入, 表示
Data number	データ選択

ボタンの意味

FILE	ファイルのオープン, CIRAM の終了等の指示
PROPERTY	CIRAM 全体の出力形式の設定
GRAPH	グラフ出力仕様の設定
FUNCTION	フォートラン関数文の生成
COMPOSE	CIRAM データの合成

(b) データファイル読込後

図5.3 初期画面

5.3 FILE プルダウンメニュー

初期画面で [FILE] をクリックするとファイルのオープン、CIRAMの終了等を指示するFILEプルダウンメニュー(図5.4)が表示される。各メニューの機能を以下に示す。

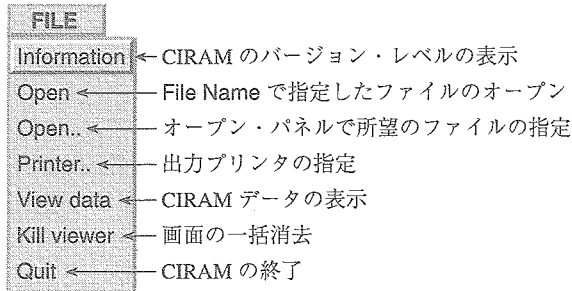


図5.4 FILEプルダウンメニュー

Information

FILEプルダウンメニューで [Information] をクリックすると、CIRAMのバージョン・レベルが表示される。

Open

FILEプルダウンメニューで [Open] をクリックすると、初期画面(図5.3)のFile nameフィールドに入力されたCIRAMデータファイルをオープンする。指定したファイルが存在しない等の異常があると警告画面が表示される。

Open..

FILEプルダウンメニューで [Open..] をクリックすると、オープン・パネル(図5.5)が表示され、マウス操作で所望のCIRAMデータファイルを指定しオープンすることができる。

CIRAMデータファイルの指定は表示画面のリスト・ボックスを操作することによって行われる。左側のリスト・ボックスで指定(クリック)されたディレクトリの内容が、右側のリスト・ボックスに表示される。[◀][▶] ボタンをクリックすることによって、ディレクトリをルートに近づけたり遠ざけたりすることができる。目標のファイルを選択(クリック)し、それを開く場合は [OK] ボタンをクリックする。[OK] をクリックすると、初期画面(図5.3)のFilenameとコメント部/データ構成情報が対応して書き換えられ、オープン・パネルは消える。また、ディレクトリ名の指定はリスト・ボックスを用いるほかに、Directoryフィールドに直接ディレクトリ名を入力しリターンキーを押すことによっても、そのディレクトリに移動することができる。

Filterフィールドにワイルド・カードを利用してフィルター条件を設定することができる。フィルターの設定例を以下に示す。

- ・ 拡張子が `crm` のファイルだけを選択したい場合。
Filter : `*.crm`
- ・ 頭文字が `T` で5文字のファイルだけを選択したい場合。
Filter : `T????`

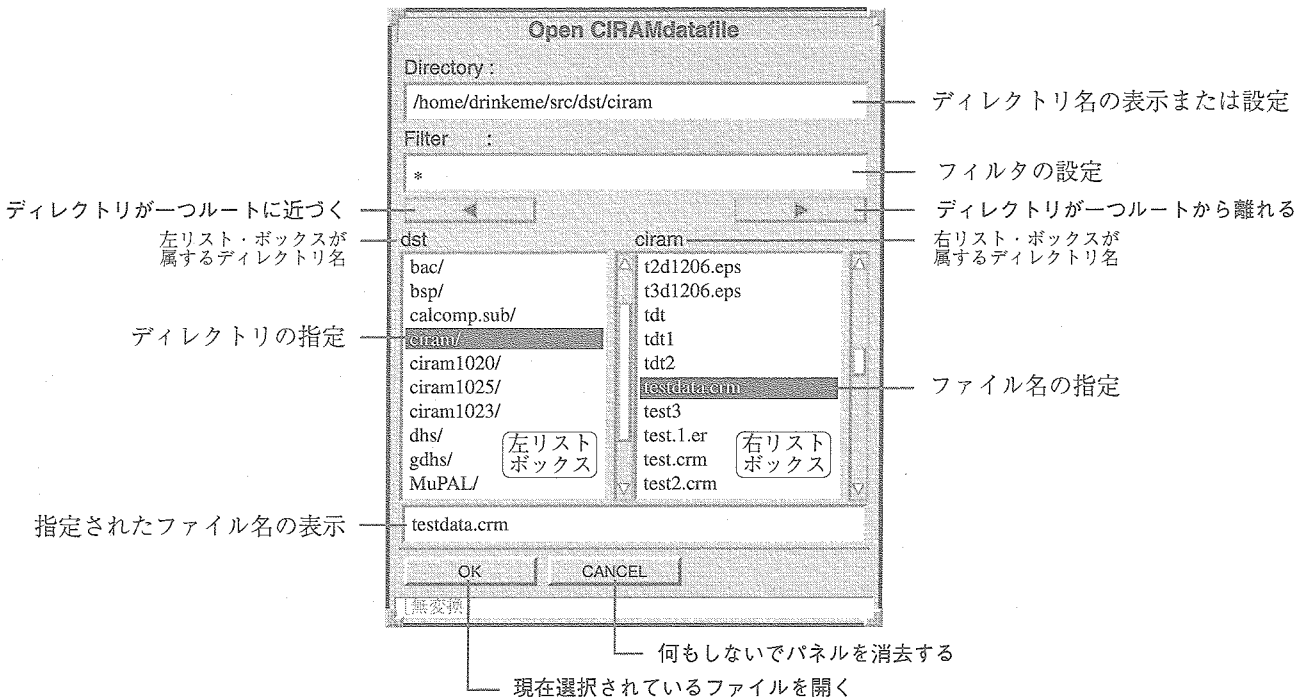
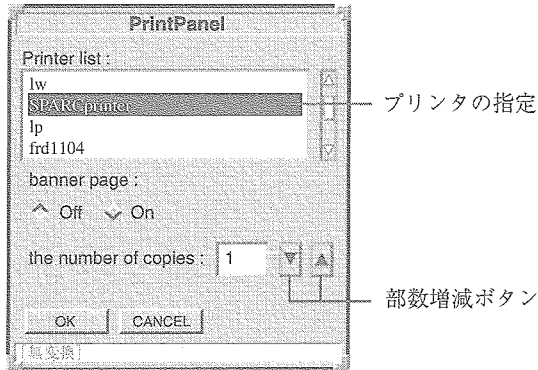


図5.5 ファイル・オープン・パネル

Printer..

FILEプルダウンメニューで [Printer..] をクリックすると、プリンタ指定画面 (図5.6) が表示され、任意の出力プリンタ、バナーページの出力の有無及び印刷部数を指定できる。



画面上の指示とその意味	
Printer list	使用可能プリンター一覧
banner page	バナー・ページの出力選択
the number of copies	プリントする部数

ボタンの意味	
OK	パネル上の設定を実行する。
CANCEL	何もしないでパネルを消去する。

図5.6 プリンタ指定画面

View data

FILEメニューで [View data] をクリックすると、CIRAMデータのコメント部/データ構成情報部に続いてデータ部を初期画面に表示する。初期画面を拡大するか、画面右及び下に表示されるスクロール・バーを移動 (ドラッグ) することによってデータ部を見ることができる。画面の拡大は、四隅の何れかに矢印のポインタを移動するとポインタが丸に変わるので、そこでドラッグすることによって行われる。

Kill viewer

FILEメニューで [Kill viewer] をクリックすると、CIRAM駆動中にgostviewによって表示された全ての画面を一括して消去することができる。消去を実行するか否かは確認パネルで行われる。

Quit

FILEメニューで [Quit] をクリックすると、終了確認パネルが表示されCIRAMを終了することができる。

5.4 プロパティ設定 (PROPERTY)

初期画面で [PROPERTY] ボタンをクリックすると、

プロパティ設定画面 (図5.7) が現れ、CIRAM全体の出力形式を好みに合わせて設定することができる。

図に示されるように7項目を設定することができ、上部4項目は所望のラジオボタンをクリックすることによって選択する。ラジオボタンは凹んで見えるのが選択されていることを表す。下部3項目は、各項目のフィールドに数値を入力しリターンキーを押すか、フィールドの右にあるスライダーをドラッグすることによって設定する。項目の設定を変更した場合は、必ず [APPLY] ボタンをクリックし設定を実行しなければならない。

5.5 グラフ出力仕様の設定 (GRAPH)

初期画面で [GRAPH] ボタンをクリックすると、グラフ出力仕様設定画面 (図5.8) が現れ、グラフのディスプレイへの表示、プリンタへの出力、PostScriptファイルの保存及びグラフの修正を行うことができる。

図5.8は3次元データの場合の画面表示例を示すが、図中で説明されているように次元数に応じて表示が異なる。第1変数 (Parm.1) / 第2変数 (Parm.2) / 第3変数 (Parm.3) と軸変数 (Axis) / 線変数 (Line) / 図変数 (Graph) の対応は所望のラジオボタンをクリックすることにより変更できる。

3次元データの場合、図変数の次数分だけグラフが出力される。画面の中央に表示されるDraw Start No., End No.は出力する図の範囲を指定するものであり、スライダーをドラッグすることにより設定する。

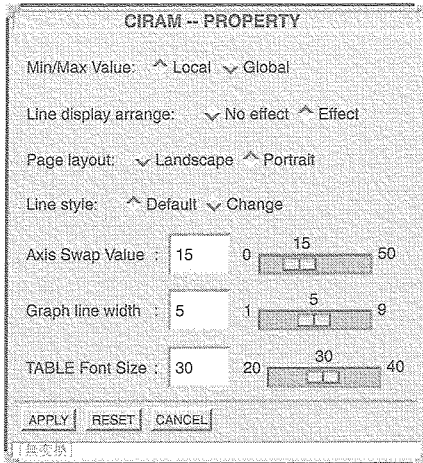
必要な設定が終了したら最下段にある所望のボタンをクリックする。[SAVE] 及び [EDIT] ボタンをクリックすると更にメニュー画面が表示される。

[DISPLAY] による表示では、プロパティ設定で指示された文字サイズ等による画面表示が行われるので、表示に時間がかかる。[EDIT] による表示は瞬時に行われるので、グラフをディスプレイ上で確認したいときなどに便利である。

グラフの出力例を付録Aに示す。

SAVE (PostScriptファイルの保存)

グラフ出力仕様設定画面で [SAVE] ボタンをクリックするとセーブ・パネル (図5.9) が表示されるので、ディレクトリ及びファイル名を指定してPostScriptファイルを保存することができる。Directory及びFilenameフィールドにはデフォルト名が表示されるので、変更する場合はキー・ボードより入力し、セーブ・パネルの [SAVE] ボタンをクリックする。[SAVE] ボタンをクリックするとファイルが保存され、セーブ・パネルは消える。既存ファイルに保存すると無条件で上書きされる。

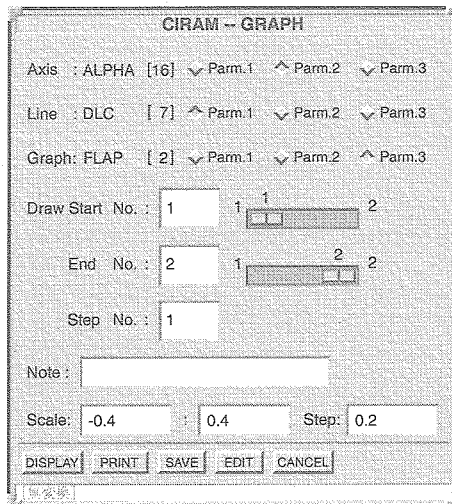


画面上の指示とその意味	デフォルト
Min/Max Value 3次元データで、複数のグラフを表示/印刷する場合にグラフのスケールを共通 (Global) にするか、個別 (Local) にするか の指示。	Local
Line display arrange 2次元以上のグラフの右肩に付ける線変数に引き出し線をつける (Effect) か否 (No Effect) かの選定。	Effect
Page layout A4用紙の方向を横 (Landscape) にするか、縦 (Portrait) にするか の選定。	Portrait
Line style グラフを実線のみ (Default) にするか、異なる7種類の線種 (Change) にするか の選定。	Default
Axis Swap Value 2次元以上のグラフでは、これで指定された数以上のデータ点を有する変数が自動的に軸変数となる。設定値が0の場合は常に第1変数が軸変数である。	15 [0 -- 50]
Graph line width グラフの線幅指定 (1単位は0.1mm)。	5 [1 -- 9]
Table Font Size データ・テーブル印刷時の文字サイズ指定。 40単位: 11pt, 29単位: 9pt, 22単位: 6pt	30 [20 -- 40]

ボタンの意味

APPLY	パネル上の設定を実行する。
RESET	パネル上の設定をデフォルトに戻す。
CANCEL	何もしないでパネルを消去する。

図5.7 プロパティ設定画面

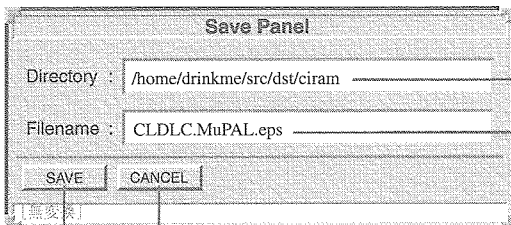


画面上の指示とその意味	備考
Axis	次元数に応じてどの変数を軸変数/線変数/図変数に対応させるかを設定する。
Line	2次元以上で表示
Graph	3次元時のみ表示
Draw Start No.	描画開始の図番
End No.	描画終了の図番
Step No.	1画面に描画する図数 (用紙横方向時のみ有効)
Note	図に記入する注釈
Scale	縦軸表示の最小値/最大値
Step	縦軸の刻み

ボタンの意味

DISPLAY	ディスプレイに表示する。
PRINT	プリンタに出力する。
SAVE	PostScript ファイルを保存する。
EDIT	グラフ修正画面を表示する。
CANCEL	何もしないでパネルを消去する。

図5.8 グラフ出力仕様設定画面



- 保存先のディレクトリの指定 (デフォルト: ciram を起動したディレクトリ)
- 保存時のファイル名の指定 (ファイル名のデフォルト: データ名.機体名.eps)

何もしないでパネルを消去する

PostScript ファイルを保存する

図5.9 PostScriptファイルの保存画面

EDIT (グラフの修正)

グラフ出力仕様設定画面で [EDIT] ボタンをクリックすると、指定されたグラフが表示され修正が可能になる。修正手順を以下に示す。

1) グラフ修正画面のオープン

グラフ出力仕様設定画面 (図5.8) で [EDIT] ボタンをクリックすることによって、グラフ修正画面 (図5.10) をオープンする。

最初に表示されるグラフは初期画面のData numberで指定されたデータであるが、グラフ修正画面を表示したままで初期画面のData numberを変更することにより、表示されるグラフがそれに応じて切り替わるので、所望のデータを簡単に選択することが可能である。また同様に、グラフ出力仕様設定画面の軸変数/線変数/図変数の切換も瞬時にグラフ修正画面に反映される。3次元データの図変数の切換は、グラフ出力仕様設定画面のDraw Start No.を変えることによって行われる。

2) 修正対象線変数の指定

2次元以上のグラフでは所望の線変数をダブル・クリックするか、グラフ修正画面の左下にある [上下ボタン] によって線変数値を現表示値より繰上げ/繰下げすることによって指定する。指定されるとその線変数が赤色になるとともにデータ点が■で表示される。1次元グラフでは常に修正対象グラフが指定された状態になっている。

3) 修正データ点の指定

修正したいデータ点の■をクリックするか、グラフ修正画面の左下にある [左右ボタン] で移動すると茶色の十字カーソルが現れ、その点が修正対象点となる。同時にデータ値が Input valueフィールドに表示される。

4) データの修正

データは以下の2つの方法で修正することができる。

a) 修正したいデータ点の■を上下にドラッグし所望の位置まで移動する。その間データ値は■の動きに対応して Input valueフィールドに表示される。

b) Input valueフィールドに所望のデータ値を打ち込みリターンキーを押すことによって修正する。

修正作業においては [ZOOM] ボタンをクリックし、拡大率を選択すると、修正点周辺が拡大され、より精密な修正を行うことができる。

5) 修正値の設定

一つのデータ点について修正操作が完了したら、[SET] ボタンをクリックすることによって修正値が設定され、右下に undo が表示される。

現在表示中のグラフを更に修正する場合は2)~5)を繰り返す。また、新たなグラフを修正したい場合は1)に戻り、初期画面のData numberを変更することにより所望のグラフを選定し修正作業を行う。

設定した修正を解除するには [UNDO] ボタンをクリックする。設定解除は256回前までの設定値に対して可能である。解除可能な場合には、グラフ画面右下に undo と表示されている。

6) 修正結果の保存

グラフ修正の結果は [SAVE] ボタンをクリックすることによってセーブ・パネル (図5.11) を開きディレクトリと新たなファイル名を指定して保存する。リスト・ボックスの操作法はファイルのオープン・パネルと同様である (5.3節参照)。セーブ・ファイル名が現在読み込んでいるファイル名と同一であると、警告画面が表示され保存は拒否される。

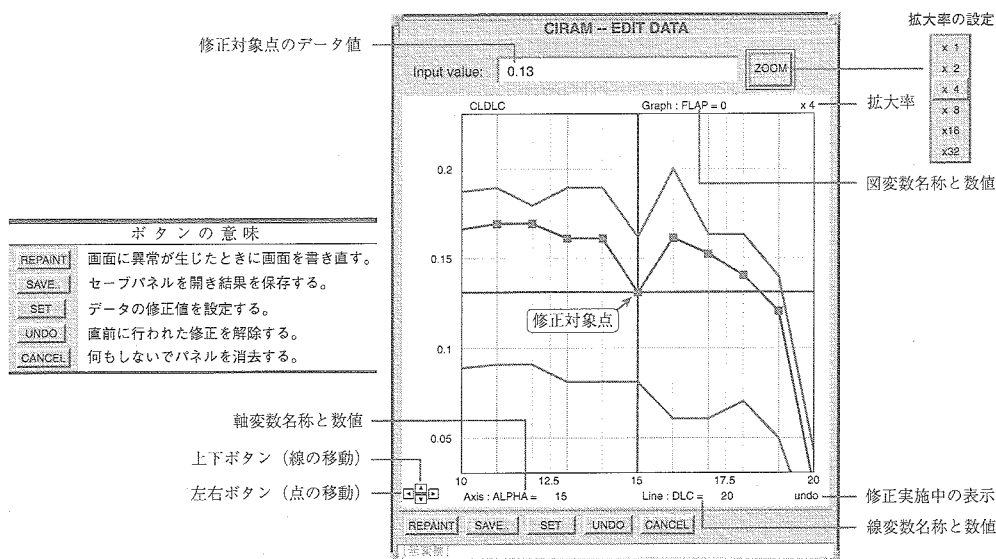


図5.10 グラフ修正機能

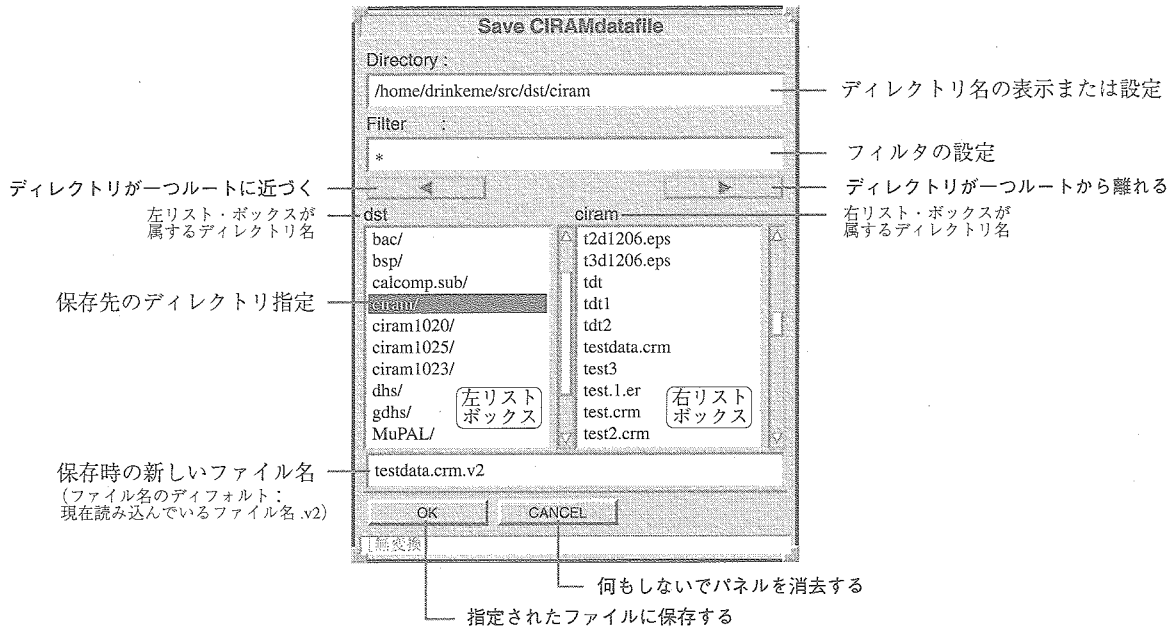


図5.11 修正結果の保存画面

5.6 フォートラン関数文の生成 (FUNCTION)

初期画面で [FUNCTION] ボタンをクリックすると、現在選択されているデータのフォートラン関数文生成設定画面 (図5.12) が表示される。

CIRAMで生成されるフォートラン関数文の補間方式は直線補間 (図5.13) であるが、関数文内のデータ文の記述形式によって、勾配/切片 (Conversion) 型とデータ値 (Direct) 型を選定できる。勾配/切片型を選択した場合は、不等間隔補間 (Unequaly) と等間隔補間 (Regular) を更に選定できる。

勾配/切片型は、CIRAM開発時の計算機性能が低かったために、計算機の負荷を極力少なくし、実時間シミュレーションを実現するために考案された。図5.13に示すようにデータから勾配と切片を計算し、関数文生成時にそれらをデータ文で記述することによって計算機の負荷をを少なくしている。これによって補間時に必要な、加減算9回、乗算2回、除算2回の演算が節約される。等間隔補間でフォートラン関数文を生成すると、補間区間の探索が瞬時に行われるので、更に計算機への負荷が減少する。

計算機の発達は著しく、最近では勾配/切片型のような工夫を施さなくとも実時間シミュレーションが実現可能となってきている。そこで、データ値をデータ文にそのまま記述し、図5.13に示される式を直接解く関数文作成機能 (Direct) を追加した。勾配/切片型で作成される関数文内のデータ文からはその関数の物理的意味は全く読み取れないが、データ文を直接記述することによってこ

の欠点が解消される。

等間隔補間の場合は上記の他に、各データ値を直線補間でなく平滑化して実施する機能と図5.14に示す情報を表示すると同時に等間隔補間に必要な数値を設定するフィールドがある。平滑化を行う場合は [spline] をクリックする。

フォートラン関数文生成設定画面において所望の補間方法を選定し、関数文の格納先ディレクトリ及びファイル名を指定し、[GENERATE] ボタンをクリックすると関数文が生成される。既存ファイルに保存すると無条件で上書きされる。成功裡にフォートラン関数文が生成されると図5.15に示すパネルが表示されるので [Confirm] ボタンをクリックする。何らかの理由で関数文が生成されなかった場合は、図5.15のパネルは表示されず、CIRAMを起動したウィンドウ内に理由が表示される。

勾配/切片 (Conversion) 型の不等間隔補間 (Unequaly) で生成された関数文は、リンク時に補間用サブルーチン Z_SRCHを、またデータ値 (Direct) 型は次元数に応じて補間用サブルーチン LINEAR1 (1次元), LINEAR2 (2次元), LINEAR3 (3次元) を必要とする。各サブルーチンのプログラムリストを付録Bに示す。

5.7 CIRAM データの合成 (COMPOSE)

風洞試験では、例えばマッハ数を一定にして迎角スイープを実施する。別のマッハ数で迎角スイープを実施することによって、迎角を変数とした1次元データがマッハ数に関して複数個得られる。しかし、運動計算等ではマ

ッハ数と迎角の2次元データとして取り扱う方が効率的である。このような場合にデータ合成機能を用いると容易に新たな高次元データ・テーブルを作成することができる。規定された条件を満たしていれば図5.16に示す合成が可能である。

初期画面で [COMPOSE] ボタンをクリックするとコンボウズ画面 (図5.17) が表示され、CIRAM データの合成が可能になる。合成手順を以下に示す。

1) 合成対象ファイルのオープン

コンボウズ画面で [Open..] をクリックすると、オープン画面 (図5.18) が表示され、現在呼び出されているデータに合成すべき2番目のデータを指定することができる。[OK] をクリックすると、オープン画面が消え、データファイルが読み込まれ、コンボウズ画面 (図5.17) のFilenameとコメント部/データ構成情報が対応して書き換えられる。リスト・ボックスの操作はFILEプルダウンメニューのオープン・パネルの場合と同様である (5.3節参照)。

2) 合成結果の保存先ファイル名の指定

コンボウズ画面で [Save..] ボタンをクリックすると、セーブ画面 (図5.19) が表示され合成結果の保存先ファイル名を指定することができる。合成後のデータを現在呼び出しているファイルに保存しようとする、警告画

面が表示され保存を拒否される。合成後のデータを既存ファイルに保存する場合は、上書きを実行するか否かの確認を求められる。保存時のファイル名を指定し、[OK] ボタンをクリックするとセーブ画面が消え、保存先ファイル名がコンボウズ画面下段のCompose nameフィールドに表示される。リスト・ボックスの操作はFILEプルダウンメニューのオープン・パネルの場合と同様である (5.3節参照)。

3) データ合成の実行

コンボウズ画面のコメント部とデータ構成情報欄の下に合成に必要なパラメータの記入フィールドが表示されるので、必要事項を入力しなければならない。図5.17のコンボウズ画面は、合成対象次元数「1次元+1次元」、合成結果次元数「2次元」(図5.16, ②) の場合を例示している。この表示内容は合成対象次元数及び合成結果次元数の次数によって変化する。必要なパラメータを設定した後、[COMPOSE] ボタンをクリックするとデータの合成が実行され、指定された保存先ファイルに書き込まれる。条件を満たさない場合には、警告画面が表示され合成は行われぬ。引き続き合成を実行すると、指定された保存先ファイルに追記 (APPEND) される。保存先ファイル名を変更する場合は、上記2)を実行した後、合成を行う。

画面上の指示とその意味	
Output type	関数文における DATA 文の形式を指定。勾配/切片 (Conversion) / データ値 (Direct) の指示。
Space type	等間隔 (Regular) / 不等間隔 (Unequaly) 補間の指示。
Function Dir Name	関数文の格納先ディレクトリを指定。デフォルトは ciram を起動したディレクトリである。
Function File Name	関数文のファイル名を指定。ファイル名のデフォルトは F.データ名.機種名.f である。
ボタンの意味	
GENERATE	現在の設定でフォートラン関数文を生成する。
RESET	パネル上の設定をデフォルトに戻す。
CANCEL	何もしないでパネルを消去する。

図5.12 フォートラン関数文生成設定画面

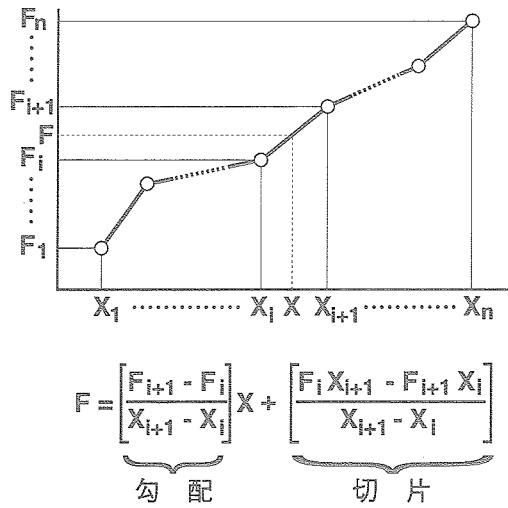


図5.13 補間方法

CIRAM -- FUNCTION Generation

Space type : Regular Unequaly

spline

Parameter	Name	Number	Min	Max	Interval
1 Axis	DLC	7	-30	25	9.17
2 Line	ALPHA	16	-10	23	2.20
3 Graph	FLAP	2	0	5	5.00

Function Dir name :

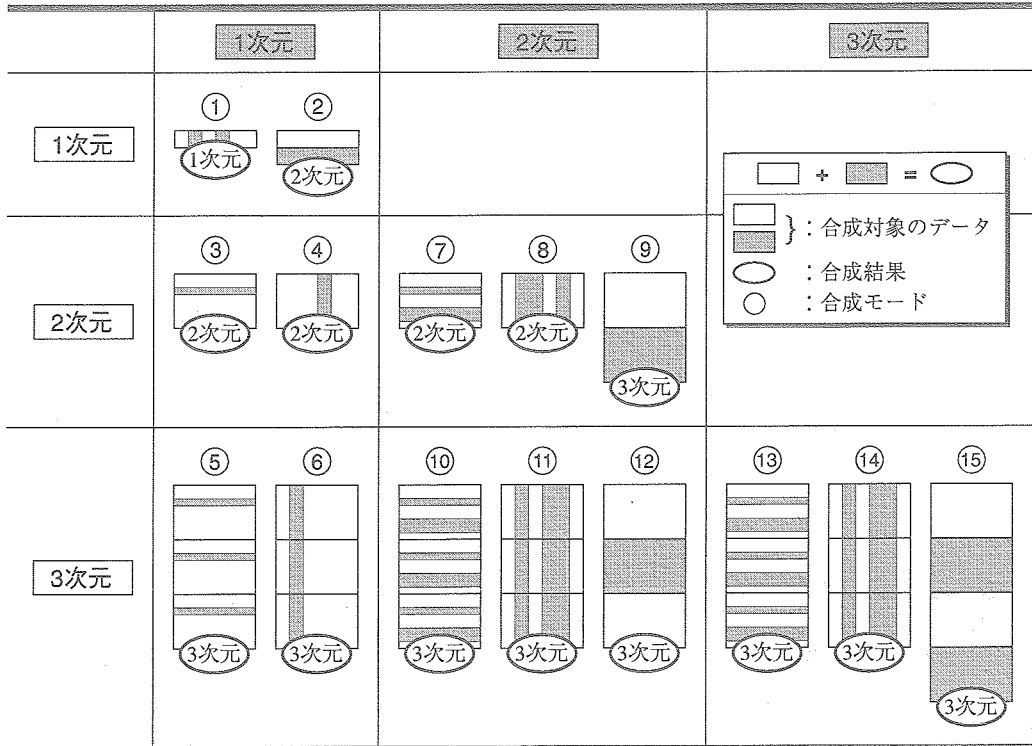
Function File name :

画面上の指示とその意味					
Space type 等間隔 (Regular) / 不等間隔 (Unequaly) 補間の指示。					
spline データ値平滑化の指示。					
Parameter					
Axis (常に表示)	触変数名	データ数	最小値	最大値	データ間隔値の指定
Line (2次元以上で表示)	線変数名				
Graph (3次元で表示)	図変数名				
Function Dir Name 関数文の格納先ディレクトリを指定。					
Function File Name 関数文のファイル名を指定。ファイル名のデフォルトは F.データ名.機種名.f である。					
ボタンの意味					
<input type="button" value="GENERATE"/>	現在の設定でフォートラン関数文を生成する。				
<input type="button" value="RESET"/>	パネル上の設定をデフォルトに戻す。				
<input type="button" value="CANCEL"/>	何もしないでパネルを消去する。				

図5.14 フォートラン関数文生成設定画面 (等間隔補間の場合)

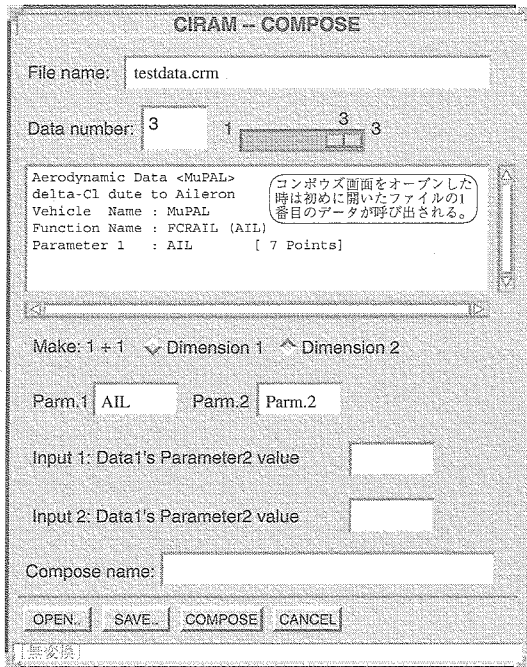


図5.15 関数文生成確認パネル



合成対象次元数	合成結果次元数	合 成 条 件	合 成 モード
1次元+1次元	1次元	両者の第1変数値に同一値が存在しない。	①
	2次元	両者の第1変数値が完全に一致している。	②
2次元+1次元	2次元	両者の第1変数値が完全に一致している。	③
		2次元データの第2変数値と1次元データの第1変数値が完全に一致している。	④
3次元+1次元	3次元	両者の第1変数値が完全に一致している。	⑤
		3次元データの第2変数値と1次元データの第1変数値が完全に一致している。	⑥
2次元+2次元	2次元	両者の第1変数値が完全に一致し、かつ両者の第2変数値に同一値が存在しない。	⑦
		両者の第2変数値が完全に一致し、かつ両者の第1変数値に同一値が存在しない。	⑧
	3次元	両者の第1変数値と第2変数値が完全に一致している。	⑨
3次元+2次元	3次元	両者の第1変数値が完全に一致し、かつ両者の第2変数値に同一値が存在しない。	⑩
		両者の第2変数値が完全に一致し、かつ両者の第1変数値に同一値が存在しない。	⑪
		両者の第1変数値と第2変数値が完全に一致している。	⑫
3次元+3次元	3次元	両者の第1変数値と第3変数値が完全に一致し、かつ両者の第2変数値に同一値が存在しない。	⑬
		両者の第2変数値と第3変数値が完全に一致し、かつ両者の第1変数値に同一値が存在しない。	⑭
		両者の第1変数値と第2変数値が完全に一致し、かつ両者の第3変数値に同一値が存在しない。	⑮

図5.16 CIRAMデータの合成条件



画面上の指示とその意味	
File name	合成対象のファイル名の表示
Data Number	データ選択
Make	各合成対象と合成結果の次元数の表示
Param.1	第1変数名
Param.2	第2変数名
Input 1	次元数が上がったことによって必要となった新たな変数値の指定
Input 2	次元数が上がったことによって必要となった新たな変数値の指定
Compose name	合成結果の保存ファイル名の表示

ボタンの意味	
OPEN..	オープン・パネルを表示し合成対象ファイルを指定する。
SAVE..	セーブ・パネルを表示して合成結果の保存ファイル名を指定する。
COMPOSE	データ合成を実行する。
CANCEL	何もしないでパネルを消去する。

図5.17 コンポズ画面

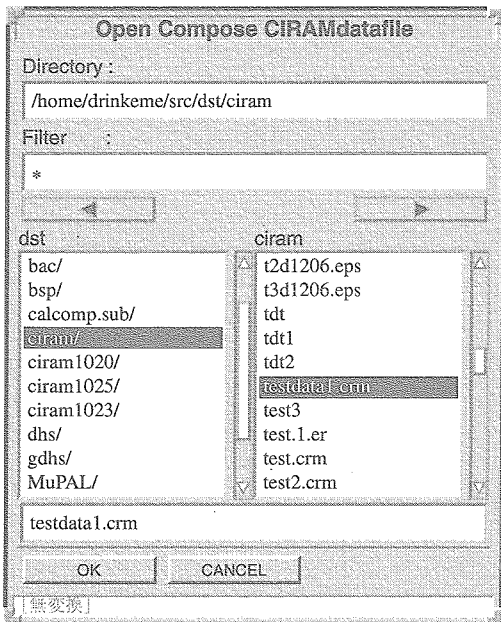


図 5.18 コンポズのファイル・オープン画面

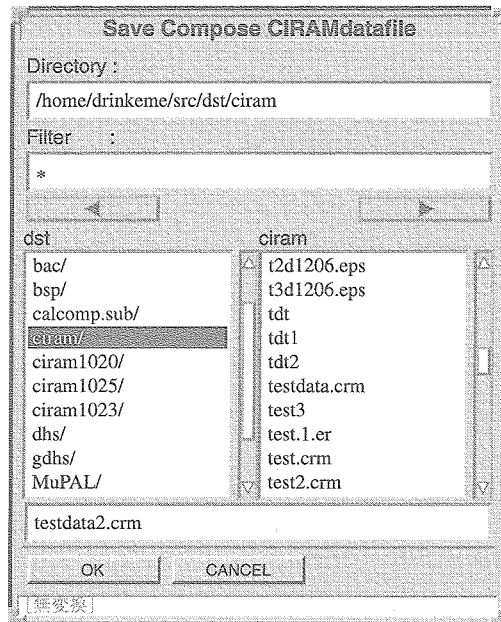


図 5.19 コンポズのファイル・セーブ画面

6. 拡張CIRAMフォーマット

拡張CIRAMフォーマットは標準CIRAMフォーマットに対し、コメントとコマンドを付加して処理・指定等をファイルに記述しておくことができる。「データファイルに自由にコメントを記入する」、「横軸にどの変数を割り当てる」、「生成した関数文の格納ディレクトリを指定す

る」等をあらかじめデータファイルに記入することが可能である。拡張CIRAMフォーマットのコマンドを使用することにより、特に次章で述べるバッチ処理を実行する時に、細かい設定をあらかじめ指定することができるので非常に便利である。

拡張CIRAMフォーマットで記述する際の基本文法を

以下に示す。

- ・行頭が#の行はコメントとみなす。
- ・行頭が&の行はコマンドとみなす。
- ・コマンド文の大文字/小文字は区別しない。
- ・コマンド行中にスペース/タブは任意に入れてよい。
- ・空白行があってはならない。

コマンド系は、共通コマンド(表6.1)と個別コマンド(表6.2)から構成される。共通コマンドはファイル中の

全データに影響するコマンドであり、標準CIRAMフォーマットの最初のコメント行より前に&&で始まる行で記述する。標準CIRAMフォーマットの最初のコメント行以降に記述された&&で始まる行は無視される。個別コマンドは1つのデータにのみ影響するコマンドであり、標準CIRAMフォーマットのコメント行に続いて&で始まる行で記述する。コマンド文と設定値は等号(=)で結ぶ。拡張CIRAMフォーマットの記述例を付録Aに示す。

表6.1 共通コマンド

コマンド	コマンドの意味	デフォルト
DataLineLength	データ記述行の最大コラム数の指定。数値(50-256)で設定。	72
MinMaxValue	変数軸のスケーリング指定 3次元グラフで縦軸スケールを全体(Global)で行うか個別(Local)で行うかを指定。	Local
LineParamArrange	線変数の記述指定 線変数値を引出線を付けて記入する(On)か否(Off)かの指定。	On
PageLayout	出力方向の指定 紙の方向を縦(Portrait)にするか横(Landscape)かの指定。	Portrait
AxisSwapValue	グラフ出力時の軸変数指定 指定した値を越える個数を持つ変数を軸変数とする。0の場合は第1変数が軸変数軸変数となる。数値(0-256)で設定。	15
LineWidth	グラフの線幅の指定 指示値1が0.1mm幅。数値(1-9)で設定。	5
TableFontSize	数表の文字サイズ指定 指示値1が0.275pt。数値(20-40)で設定。	30
SortTableParam	数表出力をグラフの縦・横軸に合わせるか(On)か否(Off)かの指定 Off: データ構成(図)で指定した通りに出力する On: グラフ出力の軸・線・図変数に合わせる	Off
SaveDirName	印刷ファイルの格納先ディレクトリ指定(文字列でパスを指定)	現在いるディレクトリ
FuncDirName	関数文の格納先ディレクトリ指定(文字列でパスを指定)	現在いるディレクトリ
FunctionStyle	生成される関数文のファイル名形式指定 0="F.データ名.機体名.f" 1="F.データ名.f" 2="Fデータ名.f"	0
FuncFormat	関数文の生成方法の指定 データから勾配と切片を計算し、生成時にデータ文に記述(Conversion) データ構成そのままをデータ文に記述(Direct)	Conversion
PrinterName	印刷するプリンタ名指定(文字列でプリンタ名を指定)	計算機のデフォルト
NoteDefaults	ノート(出力時左下に書かれる)の表示形式指定 None : ノート設定なし FileName : ファイル名 DataNo : データ番号 FileName&No : ファイル名とデータ番号	None
PageStep	図の出力方向が[横(PageLayout=Landscape)]の場合1ページに描くグラフの個数指定。数値(1-4)で設定。	1
TimeDefaults	タイム・スタンプ(出力時中央下に書かれる)の指定 Date&Time : 日付と時間 Date : 日付	Date&Time

表6.2 個別コマンド

コマンド	コマンドの意味	デフォルト
AxisParam	グラフの軸変数指定	v1 または p1
LineParam	グラフの線変数指定	v2 または p2
GraphParam	グラフの関変数指定 各変数の記号 第1変数 : v1 または p1 第2変数 : v2 または p2 第3変数 : v3 または p3	v3 または p3
SaveName	印刷ファイル時のファイル名 (文字列でファイル名を指定)	データ名.機体名.eps

7. バッチ処理の方法

データファイルの全てのデータに対し、一括して処理を行うバッチ処理モードは下記に関して有効である。

- ・フォートラン関数文生成と保存
- ・PostScriptファイルの保存
- ・印刷

これらは以下のコマンド・ライン入力によってそれぞれ実行される。

1) フォートラン関数文生成と保存

```
%ciram <DataFileName> -bf または
```

```
%ciram -bf <DataFileName>
```

保存先ディレクトリは共通コマンドFuncDirName(表6.1)で指定することができ、省略時はCIRAM起動時のディレクトリに作成される。保存時の各関数のファイル名は共通コマンドFunctionStyle(表6.1)で指定することができ、省略時のファイル名は「F.データ名.機種名.f」となる。

2) PostScriptファイルの保存

```
%ciram <DataFileName> -bs または
```

```
%ciram -bs <DataFileName>
```

保存先ディレクトリは共通コマンドSaveDirName(表6.1)で指定することができ、省略時はCIRAM起動時のディレクトリに作成される。保存時の各関数のファイル名は個別コマンドSaveName(表6.2)で指定することができ、省略時のファイル名は「データ名.機体名.eps」となる。

3) 関数文生成・保存とPostScriptファイルの保存(組合せ指定)

```
%ciram DataFileName -bfs または
```

```
%ciram -bfs <DataFileName>
```

4) 印刷

```
%ciram <DataFileName> -bp または
```

```
%ciram -bp <DataFileName>
```

プリンター名は共通コマンドPrinterName(表6.1)で正しく指定されていないなければならない。

8. おわりに

飛行シミュレーション用数学モデルを作成する際に必要な数学モデル・データ生成システム(CIRAM)の機能と使用法について述べた。

CIRAMの原型は、航空宇宙技術研究所で行われた低騒音STOL実験機「飛鳥」の飛行シミュレーション試験実施時に坂東俊夫氏(元飛行システム研究領域長)が開発したソフトウエア・システムである。本システムは、飛鳥以降に行われたプロジェクト及び研究(小型着陸実験機「ALFLEX」、高速飛行実証フェーズII「HSFD2」、固定翼型多目的実証実験機「MuPAL- α 」等)の数学モデル作成の際にも活用されてきた。また、CIRAMの機能増強とGUI化に伴うプログラム開発は株式会社渡辺技術研究所の森雅秀氏及び渡辺篤氏の協力を頂いたことを付記する。

付録 A CIRAM の実行例

1～3次元データのCIRAMデータファイルの拡張CIRAMフォーマットの記述例を付図A1に示す。図中の網掛け部分（コメント部、共通コマンド、個別コマンド）を取り除いたものが標準CIRAMフォーマットとなる。

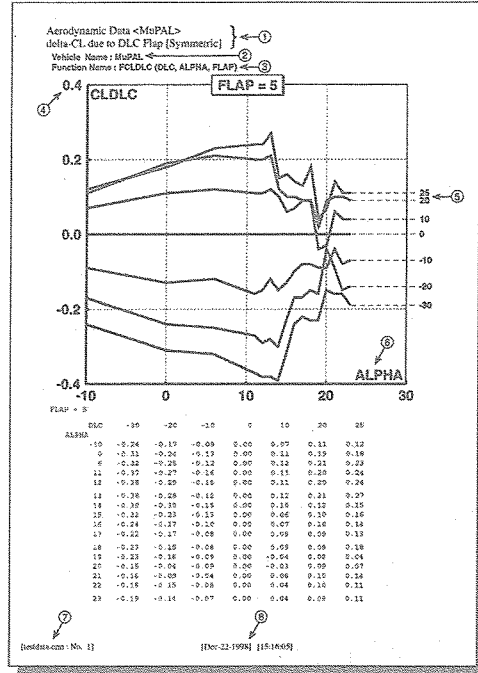
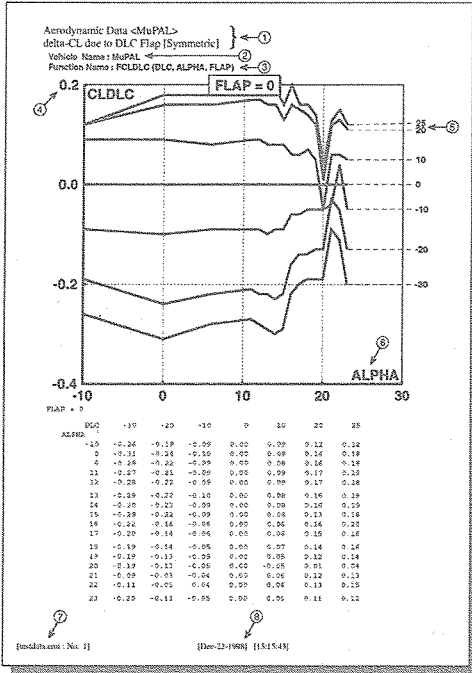
付図A1のファイルを読み込んで実施した1～3次元データのCIRAMの出力例を付図A2～付図A6にそれぞれ示す。付図A2, A3は同一の3次元データをプロパティ設定画面（5.4節参照）または共通コマンド（6章参照）で「データ軸目盛のスケール指定（MinMaxValue）」のパラメータ（Local（個別）とGlobal（全体））を変えて示してある。縦軸スケールはデータの最小最大を探索し自動的に設定されるが、その設定をLocalは図変数毎に、Globalは3次元データ全体について行う。従って縦軸スケールは、Localの場合は図変数毎に異なるのに対して、Globalの場合は図変数全てで共通になる。付図A6は紙の出力方向を横（PageLayoutをLandscapeに指定）にした場合の3次元データの出力例である。

```

=====
# Sample of CIRAM Text File
=====
#
# 共通コマンド記述開始
&& SaveDirName = ../savedir
&& FuncDirName = /fuodir
&& DatLineLength = 80
&& FunctionStyle = 2
&& AxisSwapValue = 11
&& NoteDefaults = FileName&No
# 共通コマンド記述終了
#
# 3次元データ
Aerodynamic Data <MuPAL>
delta-CL due to DLC FLAP [Symmetric]
&& SaveName = MuPAL_CDDLC
MuPAL CDDLC DLC 7ALPHA 16FLAP 2
MuPAL -30 -20 -10 0 10 20 25
0 -10 -.26 -.19 -.09 .00 .09 .12 .12
0 -.31 -.24 -.10 .00 .09 .16 .18
6 -.28 -.22 -.09 .00 .08 .16 .18
11 -.27 -.21 -.09 .00 .09 .17 .19
12 -.28 -.22 -.09 .00 .09 .17 .18
13 -.29 -.22 -.10 .00 .08 .16 .19
14 -.30 -.23 -.09 .00 .08 .16 .19
15 -.29 -.22 -.09 .00 .08 .13 .16
16 -.22 -.16 -.06 .00 .06 .16 .20
17 -.20 -.14 -.06 .00 .06 .15 .16
18 -.19 -.14 -.05 .00 .07 .14 .16
19 -.19 -.13 -.05 .00 .05 .12 .14
20 -.19 -.13 -.05 .00 -.05 .01 .04
21 -.09 -.03 -.04 .00 .06 .12 .13
22 -.11 -.05 .04 .00 .06 .13 .15
23 -.20 -.13 -.05 .00 .05 .11 .12
5 -10 -.24 -.17 -.09 .00 .07 .11 .12
0 -.31 -.24 -.13 .00 .11 .19 .18
6 -.32 -.25 -.12 .00 .12 .21 .23
11 -.37 -.27 -.16 .00 .11 .20 .24
12 -.38 -.29 -.15 .00 .11 .20 .24
13 -.38 -.28 -.12 .00 .12 .21 .27
14 -.39 -.30 -.15 .00 .10 .12 .15
15 -.32 -.23 -.13 .00 .06 .10 .16
16 -.24 -.17 -.10 .00 .07 .10 .14
17 -.22 -.17 -.08 .00 .09 .09 .13
18 -.23 -.15 -.08 .00 .09 .09 .18
19 -.23 -.16 -.09 .00 -.04 .02 .04
20 -.15 -.04 -.09 .00 -.03 .09 .07
21 -.16 -.09 -.04 .00 .06 .10 .14
22 -.16 -.15 -.08 .00 .04 .10 .11
23 -.19 -.14 -.07 .00 .04 .09 .11
#
# 2次元データ
#
Aerodynamic Data <MuPAL>
delta-CD due to DLC Flap [Symmetric, FLAP=5deg]
MuPAL CDDLC 8DLC 7ALPHA 16
-10 .0381 .0231 .0110 .0000 -.0032 -.0020 -.0010
0 .0028 -.0005 -.0009 .0000 .0066 .0147 .0171
6 -.0086 -.0109 -.0069 .0000 .0132 .0261 .0291
11 -.0225 -.0204 -.0144 .0000 .0162 .0333 .0399
12 -.0246 -.0222 -.0135 .0000 .0172 .0390 .0464
13 -.0302 -.0234 -.0122 .0000 .0192 .0375 .0473
14 -.0313 -.0273 -.0123 .0000 .0175 .0779 .0926
15 -.0811 -.0739 -.0283 .0000 .0236 .0551 .0622
16 -.0730 -.0551 -.0253 .0000 .0244 .0489 .0653
17 -.0715 -.0450 -.0233 .0000 .0240 .0511 .0647
18 -.0741 -.0517 -.0307 .0000 .0157 .0508 .0498
19 -.0726 -.0542 -.0330 .0000 .0437 .0734 .0941
20 -.0944 -.0775 -.0360 .0000 .0506 .0554 .0953
21 -.1010 -.0821 -.0554 .0000 .0326 .0555 .0794
22 -.1014 -.0687 -.0357 .0000 .0327 .0628 .0786
23 -.0857 -.0716 -.0335 .0000 .0310 .0585 .0760
#
# 1次元データ
#
Aerodynamic Data <MuPAL>
delta-Cl due to Aileron
MuPAL CRAIL 7AIL 7
-25 -20 -10 0 10 20 25
-.036 -.031 -.016 0 .018 .028 .031
=====

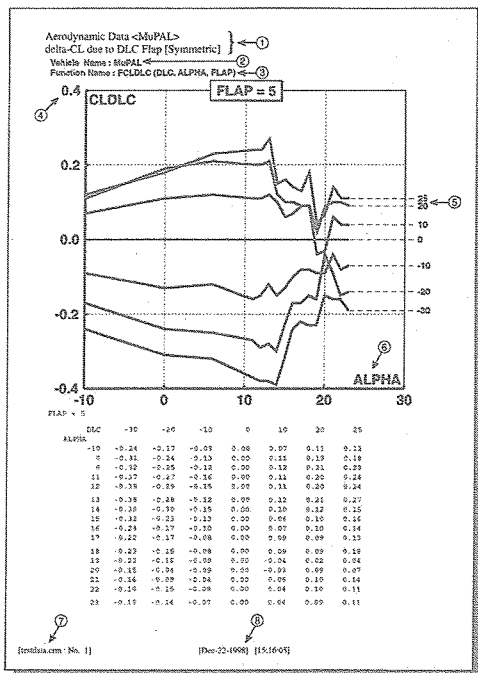
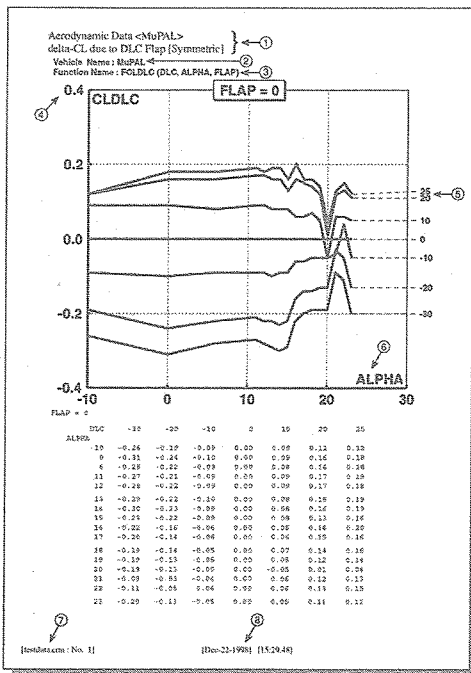
```

付図A1 CIRAMデータファイルの記述例



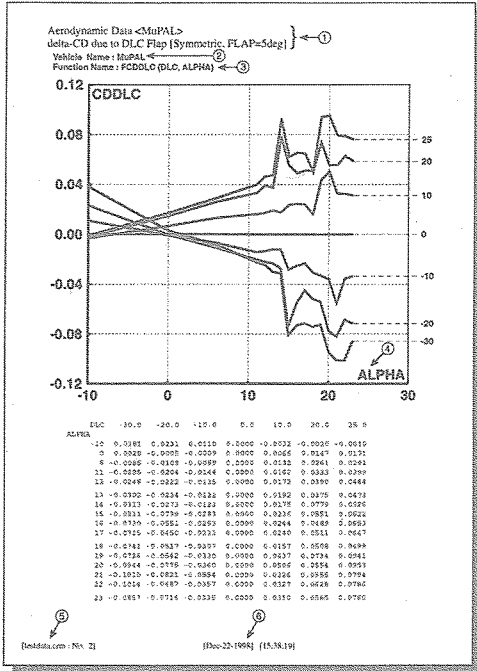
- ① コメント行 (2行)
- ② 機種名
- ③ FORTRAN 関数文 (名称と引数)
- ④ データ軸目盛のスケールリング [MinMaxValue=Local (デフォルト)]
 (目盛最大最小値毎に因変数毎に異なる)
- ⑤ 線変数が重ならない印刷 [LineParamArrange=On (デフォルト)]
- ⑥ 軸変数が第2変数 [AxisSwapValue=1] (第2変数のデータ点数16)
- ⑦ ノートはファイル名とデータ番号 [NoteDefaults=FileName&No]
- ⑧ タイム・スタンプは日付と時刻 [TimeDefaults=Date&Time (デフォルト)]

付図A2 CIRAM出力例(3次元, MinMaxValue=Local)



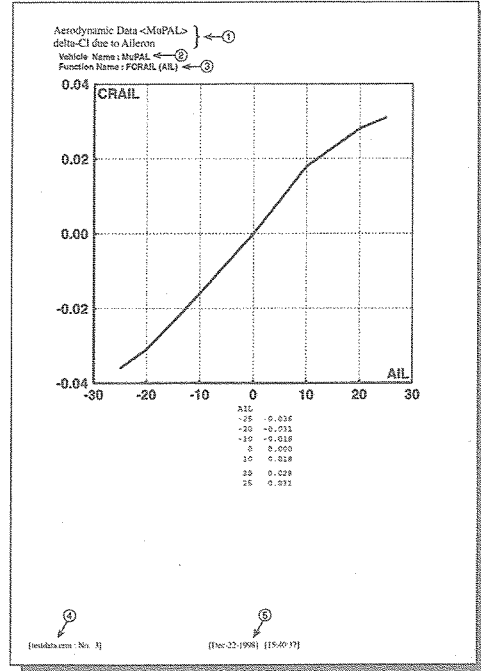
- ① コメント行 (2行)
- ② 機種名
- ③ FORTRAN 関数文 (名称と引数)
- ④ データ軸目盛のスケールリング [MinMaxValue=Global]
 (目盛最大最小値が因変数毎に共通)
- ⑤ 線変数が重ならない印刷 [LineParamArrange=On (デフォルト)]
- ⑥ 軸変数が第2変数 [AxisSwapValue=1] (第2変数のデータ点数16)
- ⑦ ノートはファイル名とデータ番号 [NoteDefaults=FileName&No]
- ⑧ タイム・スタンプは日付と時刻 [TimeDefaults=Date&Time (デフォルト)]

付図A3 CIRAM出力例 (3次元, MinMaxValue=Global)



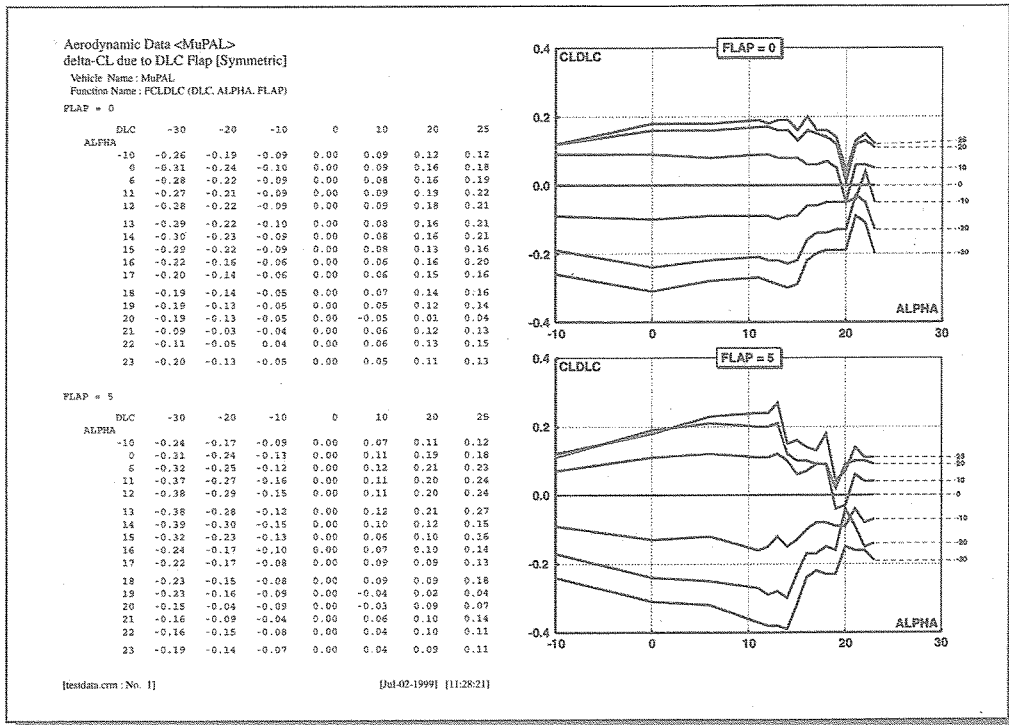
- ① コメント行 (2行)
- ② 機種名
- ③ FORTRAN 関数文 (名称と引数)
- ④ 第2変数が第1変数 [AxisSuppValue=1] (第2変数のデータ点数 16)
- ⑤ ノートはファイル名とデータ番号 [NoteDefault=FileName&No]
- ⑥ タイム・スタンプは日付と時刻 [TimeDefault=Date&Time (デフォルト)]

付図A4 CIRAM出力例 (2次元)



- ① コメント行 (2行)
- ② 機種名
- ③ FORTRAN 関数文 (名称と引数)
- ④ ノートはファイル名とデータ番号 [NoteDefault=FileName&No]
- ⑤ タイム・スタンプは日付と時刻 [TimeDefault=Date&Time (デフォルト)]

付図A5 CIRAM出力例 (1次元)



付図A6 CIRAM出力例 (出力方向横 PageLayout = Landscape)

付録B フォートラン関数文に含まれるサブルーチン

5.6節で述べたように、CIRAMによって生成されるフォートラン関数文には、補間に必要なサブルーチンが含まれている。サブルーチンのプログラムリストを以下に示す。

B1 Z_SRCH

勾配/切片型の不等間隔補間に必要なサブルーチンである。

```

C
C
C
C      SUBROUTINE Z_SRCH
C
C      PURPOSE
C          Variable's Interval Search
C
C      USAGE
C          CALL Z_SRCH (VI, VT, N, I)
C
C      Explanation of Arguments
C          VI - Input Variable [R4]
C          VT - Variable's Table [R4]
C          N - Table's Number [I4]
C          I - Interval's Number [I4]
C
C
C      SUBROUTINE Z_SRCH (VI, VT, N, I)
C
C      DIMENSION VT(*)
C
C      SIGN = VI - VT(I)
C      IF (VI-VT(I)) 20, 30, 10
C
C 10 CONTINUE
C      IF (I .EQ. N) GO TO 30
C      II = I
C      do I = II, N-1, 1
C          IF (VI .LT. VT(I+1)) GO TO 30
C      end do
C      I = N
C      GO TO 30
C
C 20 CONTINUE
C      IF (I .EQ. 1) GO TO 30
C      II = I - 1
C      do I = II, 1, -1
C          IF (VI .GE. VT(I)) GO TO 30
C      end do
C      I = 1
C
C 30 CONTINUE
C      RETURN
C      END
    
```

B2 LINEAR1

データ値型の1次元データに必要なサブルーチンである。

```

C
C
C
C      SUBROUTINE LINEAR1
C
C      PURPOSE
C          Linear Interpolation (1 Dimension)
C
C      USAGE
C          CALL LINEAR1 (NMAX, XDATA, YDATA, X, Y)
C
C      Explanation of Arguments
C          NMAX - Data Number [I4]
C          XDATA - No.1 Variable [R4]
C          YDATA - Data Variable [R4]
C          X - Given No.1 Variable [R4]
C          Y - Return Data Variable [R4]
    
```

```

C
C
C      SUBROUTINE LINEAR1 (NMAX, XDATA, YDATA, X, Y)
C
C      DIMENSION XDATA(NMAX), YDATA(NMAX)
C
C      CALL LOCATE(NMAX, XDATA, X, J)
C
C      if (J .EQ. 0) then
C          Y = YDATA(1)
C          RETURN
C      end if
C
C      if (J .EQ. NMAX) then
C          Y = YDATA(NMAX)
C          RETURN
C      end if
C
C      AVAR = (XDATA(J+1)-X) / (XDATA(J+1)-XDATA(J))
C      BVAR = (X - XDATA(J)) / (XDATA(J+1)-XDATA(J))
C      Y = AVAR*YDATA(J) + BVAR*YDATA(J+1)
C
C      RETURN
C      END
    
```

B3 LINEAR2

データ値型の2次元データに必要なサブルーチンである。

```

C
C
C
C      SUBROUTINE LINEAR2
C
C      PURPOSE
C          Linear Interpolation (2 Dimension)
C
C      USAGE
C          CALL LINEAR2 (NMAX1, NMAX2, XDATA1, XDATA2,
C                      YDATA, X1, X2, Y)
C
C      Explanation of Arguments
C          NMAX1 - No.1 Data Number [I4]
C          NMAX2 - No.2 Data Number [I4]
C          XDATA1 - No.1 Variable [R4]
C          XDATA2 - No.2 Variable [R4]
C          YDATA - Data Variable [R4]
C          X1 - Given No.1 Variable [R4]
C          X2 - Given No.2 Variable [R4]
C          Y - Return Data Variable [R4]
C
C      SUBROUTINE LINEAR2 (NMAX1, NMAX2, XDATA1, XDATA2,
C                      & YDATA, X1, X2, Y)
C
C      DIMENSION XDATA1(NMAX1), XDATA2(NMAX2),
C                      & YDATA(NMAX1,NMAX2)
C
C      X1TMP = X1
C      X2TMP = X2
C
C      CALL LOCATE(NMAX1, XDATA1, X1TMP, INDEX1)
C      CALL LOCATE(NMAX2, XDATA2, X2TMP, INDEX2)
C
C      if (INDEX1 .EQ. 0) then
C          INDEX1 = 1
C          X1TMP = XDATA1(1)
C      end if
C      if (INDEX1 .EQ. NMAX1) then
C          INDEX1 = NMAX1-1
C          X1TMP = XDATA1(NMAX1)
C      end if
C
C      if (INDEX2 .EQ. 0) then
C          INDEX2 = 1
C          X2TMP = XDATA2(1)
C      end if
C      if (INDEX2 .EQ. NMAX2) then
C          INDEX2 = NMAX2-1
C          X2TMP = XDATA2(NMAX2)
C      end if
C
C      Y1 = YDATA(INDEX1 , INDEX2)
C      Y2 = YDATA(INDEX1+1, INDEX2)
    
```

```

Y3 = YDATA(INDEX1+1, INDEX2+1)
Y4 = YDATA(INDEX1 , INDEX2+1)
C
TVAR = (X1TMP-XDATA1(INDEX1))
& / (XDATA1(INDEX1+1)-XDATA1(INDEX1))
SVAR = (X2TMP-XDATA2(INDEX2))
& / (XDATA2(INDEX2+1)-XDATA2(INDEX2))
C
Y = (1-TVAR)*(1-SVAR)*Y1 + TBAR*(1-SVAR)*Y2
& + TVAR*SVAR*Y3 + (1-TVAR)*SVAR*Y4
C
RETURN
END

```

B4 LINEAR3

データ値型の3次元データに必要なサブルーチンである。

```

C
C
C
C
SUBROUTINE LINEAR3
C
C
C
PURPOSE
C
Linear Interpolation (3 Dimension)
C
C
USAGE
C
CALL LINEAR3 (NMAX1, NMAX2, NMAX3, XDATA1,
C
XDATA2, XDATA3,
C
YDATA, X1, X2, X3, Y)
C
C
C
Explanation of Arguments
C
NMAX1 - No.1 Data Number [I4]
C
NMAX2 - No.2 Data Number [I4]
C
NMAX3 - No.3 Data Number [I4]
C
XDATA1 - No.1 Variable [R4]
C
XDATA2 - No.2 Variable [R4]
C
XDATA3 - No.3 Variable [R4]
C
YDATA - Data Variable [R4]
C
X1 - Given No.1 Variable [R4]
C
X2 - Given No.2 Variable [R4]
C
X3 - Given No.3 Variable [R4]
C
Y - Return Data Variable [R4]
C
C
C
SUBROUTINE LINEAR3 (NMAX1, NMAX2, NMAX3, XDATA1,
&
XDATA2, XDATA3, YDATA,
&
X1, X2, X3, Y)
C
C
DIMENSION XDATA1(NMAX1), XDATA2(NMAX2),
&
XDATA3(NMAX3), YDATA(NMAX1,NMAX2,NMAX3)
DIMENSION INDEX(3), YVAR(8)
C
C
X1TMP = X1
X2TMP = X2
X3TMP = X3
C
C
CALL LOCATE(NMAX1, XDATA1, X1TMP, INDEX(1))
CALL LOCATE(NMAX2, XDATA2, X2TMP, INDEX(2))
CALL LOCATE(NMAX3, XDATA3, X3TMP, INDEX(3))
C
C
if (INDEX(1) .EQ. 0) then
INDEX(1) = 1
X1TMP = XDATA1(1)
end if
if (INDEX(1) .EQ. NMAX1) then
INDEX(1) = NMAX1-1
X1TMP = XDATA1(NMAX1)
end if
C
if (INDEX(2) .EQ. 0) then
INDEX(2) = 1
X2TMP = XDATA2(1)
end if
if (INDEX(2) .EQ. NMAX2) then
INDEX(2) = NMAX2-1
X2TMP = XDATA2(NMAX2)
end if
C
if (INDEX(3) .EQ. 0) then
INDEX(3) = 1
X3TMP = XDATA3(1)
end if
if (INDEX(3) .EQ. NMAX3) then

```

```

INDEX(3) = NMAX3-1
X3TMP = XDATA3(NMAX2)
end if
C
YVAR(1) = YDATA(INDEX(1) ,INDEX(2) ,INDEX(3) )
YVAR(2) = YDATA(INDEX(1)+1,INDEX(2) ,INDEX(3) )
YVAR(3) = YDATA(INDEX(1) ,INDEX(2)+1,INDEX(3) )
YVAR(4) = YDATA(INDEX(1) ,INDEX(2) ,INDEX(3)+1)
YVAR(5) = YDATA(INDEX(1)+1,INDEX(2)+1,INDEX(3) )
YVAR(6) = YDATA(INDEX(1) ,INDEX(2)+1,INDEX(3)+1)
YVAR(7) = YDATA(INDEX(1)+1,INDEX(2) ,INDEX(3)+1)
YVAR(8) = YDATA(INDEX(1)+1,INDEX(2)+1,INDEX(3)+1)
C
SVAR = (X1TMP-XDATA1(INDEX(1))) /
& (XDATA1(INDEX(1)+1)-XDATA1(INDEX(1)))
TVAR = (X2TMP-XDATA2(INDEX(2))) /
& (XDATA2(INDEX(2)+1)-XDATA2(INDEX(2)))
UVAR = (X3TMP-XDATA3(INDEX(3))) /
& (XDATA3(INDEX(3)+1)-XDATA3(INDEX(3)))
C
Y = (1-SVAR)*(1-TVAR)*(1-UVAR)*YVAR(1)
& + SVAR*(1-TVAR)*(1-UVAR)*YVAR(2)
& + (1-SVAR)* TVAR*(1-UVAR)*YVAR(3)
& + (1-SVAR)*(1-TVAR)* UVAR*YVAR(4)
& + SVAR* TVAR*(1-UVAR)*YVAR(5)
& + (1-SVAR)* TVAR* UVAR*YVAR(6)
& + SVAR*(1-TVAR)* UVAR*YVAR(7)
& + SVAR* TVAR* UVAR*YVAR(8)
C
RETURN
END

```

B5 LOCATE

LINEAR1, LINEAR2, LINEAR3の各サブルーチンは、更にサブルーチンLOCATEを呼び出している。以下にLOCATEのプログラムリストを示す。

```

C
C
C
C
SUBROUTINE LOCATE
C
C
C
PURPOSE
C
Variable's Interval Search
C
C
USAGE
C
CALL LOCATE (NMAX, XDATA, X, INDEX)
C
C
C
Explanation of Arguments
C
NMAX - Data Number [I4]
C
XDATA - Variable [R4]
C
X - Given Variable [R4]
C
INDEX - Table's Number [I4]
C
C
C
SUBROUTINE LOCATE (NMAX, XDATA, X, INDEX)
C
C
DIMENSION XDATA(*)
C
if (X .GE. XDATA(NMAX)) then
INDEX = NMAX
RETURN
end if
C
if (X .LE. XDATA(1)) then
INDEX = 0
RETURN
end if
C
do I = 2, NMAX
if (X .LT. XDATA(I)) then
INDEX = I - 1
RETURN
end if
end do
C
END

```


宇宙航空研究開発機構研究開発資料 JAXA-RM-03-001

発行日 2004年3月25日
編集・発行 独立行政法人宇宙航空研究開発機構
〒182-8522
東京都調布市深大寺東町七丁目4番地1
TEL 0422-40-3000 (代表)
印刷所 株式会社ビー・シー・シー・
東京都港区浜松町2-4-1

©2004 JAXA

※本書(誌)の一部または全部を著作権法の定める範囲を超え、無断で複写、複製、転載、テープ化およびファイル化することを禁じます。

※本書(誌)からの複写、転載等を希望される場合は、下記にご連絡ください。

※本書(誌)中、本文については再生紙を使用しております。

<本資料に関するお問い合わせ先>

独立行政法人宇宙航空研究開発機構 情報化推進部 宇宙航空文献資料センター



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

