

JAERI-Data/Code

JP0250501

2002-019



AVS/Express (Application Visualization System)
利用手引書

2002年9月

加藤 克海*・久米 悅雄

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

本レポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合せは、日本原子力研究所研究情報部研究情報課（〒319-1195 茨城県那珂郡東海村）あて、
お申し越しください。なお、このほかに財団法人原子力弘済会資料センター（〒319-1195 茨城県那珂郡
東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Research Information Division,
Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-
gun, Ibarakiken 319-1195, Japan.

AVS／Express (Application Visualization System)
利用手引書

日本原子力研究所計算科学技術推進センター
加藤 克海*・久米 悅雄

(2002年7月1日受理)

情報システム管理課では、分散処理環境の整備作業の一環として画像処理環境の整備を進めており、画像処理サーバ、画像処理ソフトウェア AVS/Express、立体視表示装置を導入している。本報告書は、整備した画像処理環境において画像処理ソフトウェアを効率的に利用するための情報をとりまとめたものである。

AVS/Express (Application Visualization System)
User's Guide

Katsumi KATO* and Etsuo KUME

Center for Promotion of Computational Science and Engineering
(Tokai Site)
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received July 1, 2002)

Computer and network environment for image processing has been developed and maintained under the course of establishing a distributed processing environment by the information system operating division. We introduced a server for image processing, AVS/Express for image processing software and Stereo viewing system.

This report summarizes the information to use AVS/Express efficiently in the computer environment for image processing.

Keywords : AVS/Express (Application Visualization System), Image Processing,
Visualization, Stereo Viewing.

*RIST (Research Organization for Information Science Technology)

目 次

はじめに.....	1
1. AVS／Express の起動と停止	2
1.1 AVS/Express を利用するための環境	2
1.2 AVS/Express の起動	2
1.2.1 アプリケーション・タイプ	4
1.2.2 ネットワーク・エディタメニュー	5
1.2.3 モジュール構成	7
1.3 AVS/Express の停止	8
2. ネットワーク・エディタの基本操作	9
2.1 モジュールの検索	9
2.2 モジュールのインスタンス	10
2.3 モジュールの消去	11
2.4 モジュール名の変更	11
2.5 モジュールの接続	11
2.6 モジュールの開閉	12
2.7 モジュールの切断	12
2.8 モジュールの移動	12
2.9 ワークスペース上のモジュール操作	13
2.10 モジュールの情報	13
2.11 アプリケーションの保存	14
2.12 アプリケーションの削除	15
2.13 アプリケーションの読み込み	15
3. データビューワ	16
3.1 データビューワの起動画面	16
3.2 コントロールパネル	17
3.2.1 プルダウンメニュー	18
3.2.2 ツールバー	18
3.3 オブジェクトの読み込み	20
3.3.1 "Read Geom"モジュールのインスタンス	20
3.3.2 ファイルの指定	20
3.4 オブジェクトの幾何変換	21
3.4.1 マウスによる方法	21
3.4.2 メニューからの方法	21
3.5 複数のオブジェクトの読み込み	22
3.6 オブジェクトの選択	22
3.7 オブジェクトの属性の変更	23
3.7.1 オブジェクトの表示方法	23
3.7.2 オブジェクトの表面属性（色の指定）	25
3.7.3 オブジェクトの表面属性（不透明度の指定）	26
3.8 カメラの属性の変更	28
3.8.1 投影法の変更方法	28

3.8.2	ビューワ内の正規化.....	28
3.9	ビューの属性の変更.....	29
3.9.1	レンダリング・メソッド.....	29
3.9.2	背景色.....	30
3.9.3	表示ウインドウの追加.....	30
3.9.4	カメラの移動.....	31
3.9.5	表示ウインドウの消去.....	32
3.10	プルダウンメニューのまとめ	32
4.	データファイルフォーマット	33
4.1	フィールドデータ.....	33
4.1.1	ヘッダー部分の説明.....	34
4.1.2	アスキ記述ファイル.....	37
4.1.3	複数ステップのアスキ記述ファイル	39
4.1.4	ネイティブ・フィールドファイル	43
4.2	イメージデータ.....	46
4.2.1	ファイルフォーマット.....	46
4.3	ボリュームデータ.....	47
4.3.1	ファイルフォーマット.....	47
4.4	UCD データ	48
4.4.1	データ構造.....	48
4.4.2	ファイルフォーマット.....	50
4.4.3	複数ステップの UCD ファイルフォーマット	53
5.	モジュール開発.....	57
5.1	V 言語	57
5.1.1	V 言語とは	57
5.1.2	オブジェクトの階層.....	57
5.1.3	オブジェクトのベースタイプ	60
5.1.4	計算オブジェクトの作成.....	62
5.2	ユーザインターフェース (UI)	63
5.2.1	基本操作.....	64
5.2.2	位置やサイズの指定.....	64
5.2.3	自作モジュールの UI	65
5.2.4	コントロールパネルへの登録	65
5.3	モジュール作成.....	66
5.3.1	プロジェクト環境の設定.....	66
5.3.2	計算モジュールの作成.....	66
5.3.3	実行テスト.....	71
5.3.4	モジュールの保存.....	72
5.3.5	プロジェクト環境への登録.....	72
5.4	モジュール作成.....	73
5.4.1	ソースファイルと再コンパイル	73
5.4.2	モジュール定義.....	74
5.4.3	プロセス管理.....	74
5.4.4	プロジェクト環境の再構築.....	75
5.4.5	user プロセスから express プロセスへの変更	75

5.5 Field データを扱うモジュールの開発.....	76
5.5.1 Field データの構造	76
5.5.2 サンプル(メッシュ形状).....	77
5.5.3 プロジェクト環境の設定.....	78
5.5.4 フィールドデータを扱うモジュールの作成	78
5.5.5 モジュールの保存.....	85
5.5.6 プロジェクト環境への登録.....	85
5.5.7 実行テスト.....	86
5.5.8 UI モジュールとの組み合わせ	86
5.5.9 マクロ化.....	87
6. AVS/Express のヘルプ機能.....	88
6.1 オンライン・ドキュメントの表示方法	88
6.1.1 モジュール情報の表示.....	88
6.1.2 モジュール毎のオンライン・ドキュメントの表示	89
6.1.3 全オンライン・ドキュメントの表示	89
6.2 デモ・アプリケーションの起動方法	90
7. AVS/ExpressMPE の操作手順.....	105
7.1 ハードウェアの準備.....	105
7.1.1 画像サーバ (Onyx3200) の設定確認	105
7.1.2 周辺ハードウェアの準備.....	105
7.1.3 センシングデータ取得用デーモン (Trackd) の起動	106
7.2 AVS/ExpressMPE の利用方法	107
7.2.1 AVS/ExpressMPE の起動	107
7.2.2 MPE 専用ビューワモジュールのインスタンス.....	107
7.2.3 ステータス表示の停止.....	108
7.2.4 Wand パラメータの設定変更と起動	108
7.2.5 レンダリングモードの切り替え	109
7.3 AVS/ExpressMPE コントロール方法	109
7.3.1 立体視メガネの装着.....	109
7.3.2 WANDA コントローラの操作方法.....	110
7.4 AVS/ExpressMPE の終了	111
7.4.1 Window の切り替え	111
7.4.2 レンダリングモードの切り替え	111
7.4.3 [WandViewer3D2] モジュールの停止	111
7.4.4 AVS/ExpressMPE の終了	111
7.5 センシングデータ取得用デーモン (Trackd) の終了	111
謝辞	112
参考文献	112
付録 索引	113

Contents

Introduction.....	1
1. Starting and Exiting AVS/Express	2
1.1 Environment for Using AVS/Express	2
1.2 Starting AVS/Express.....	2
1.2.1 Application Type.....	4
1.2.2 Network Editor Menu.....	5
1.2.3 Module Composition.....	7
1.3 Exiting AVS/Express.....	8
2. Basic Operation of a Network Editor	9
2.1 Module is Searched.....	9
2.2 Instance About a Module.....	10
2.3 Module is Eliminated.....	11
2.4 Module Name is Changed.....	11
2.5 Module is Connected.....	11
2.6 Module is Opened and Closed	12
2.7 Module is Separated.....	12
2.8 Module is Moved.....	12
2.9 Module on a Work Space is Operated	13
2.10 Information on Module.....	13
2.11 Application is Saved.....	14
2.12 Application is Deleted.....	15
2.13 Application is Read.....	15
3. Using the DataViewer Applications	16
3.1 Starting Screen of a Data Viewer	16
3.2 Control Panel.....	17
3.2.1 Pull Down Menu.....	18
3.2.2 Tool Bar.....	18
3.3 Object is Read.....	20
3.3.1 Instance About "Read Geom" Module	20
3.3.2 File is Specified.....	20
3.4 Geometric Conversion of an Object	21
3.4.1 Method by the Mouse.....	21
3.4.2 Method from a Menu.....	21
3.5 Reading of Two or More Objects	22
3.6 Selection of an Object.....	22
3.7 Change of the Attribute of an Object	23
3.7.1 Display Method of an Object	23
3.7.2 Surface Attribute of an Object (Specification of a Color)	25
3.7.3 Surface Attribute of an Object (Opaque Mr. Specification)	26
3.8 Change of the Attribute of a Camera	28

3.8.1	Change Method of Projective Technique	28
3.8.2	Made Regular in a Viewer.....	28
3.9	Change of the Attribute of a View	29
3.9.1	Rendering Method.....	29
3.9.2	Background Color.....	30
3.9.3	Addition of a Display Window	30
3.9.4	Movement of a Camera.....	31
3.9.5	Elimination of a Display Window	32
3.10	Conclusion of a Pull Down Menu	32
4.	Data File Format	33
4.1	Field Data.....	33
4.1.1	Explanation of a Header Portion	34
4.1.2	ASCII Description File.....	37
4.1.3	ASCII Description File of a Step Two or More	39
4.1.4	Native Field File.....	43
4.2	Image Data.....	46
4.2.1	File Format.....	46
4.3	Volume Data.....	47
4.3.1	File Format.....	47
4.4	UCD Data.....	48
4.4.1	Data Structure.....	48
4.4.2	File Format.....	50
4.4.3	UCD File Format of Two or More Steps	53
5.	Developing Modules	57
5.1	V Language.....	57
5.1.1	About V Language.....	57
5.1.2	Class of an Object.....	57
5.1.3	Base Type of an Object.....	60
5.1.4	Creation of a Calculation Object	62
5.2	User Interface(UI).....	63
5.2.1	Basic Operation.....	64
5.2.2	Specification of a Position or Size	64
5.2.3	UI of a Its Original Work Module	65
5.2.4	Registration to a Control Panel	65
5.3	Module Creation.....	66
5.3.1	Setup of Project Environment	66
5.3.2	Creation of a Calculation Module	66
5.3.3	Execution Test.....	71
5.3.4	Modular Preservation.....	72
5.3.5	Registration to Project Environment	72
5.4	Module Creation.....	73
5.4.1	Source File and Re-compile	73
5.4.2	Module Definition.....	74
5.4.3	Process Management.....	74
5.4.4	Reconstruction of Project Environment	75

5.4.5	From a User Process Change in an Express Process	75
5.5	Development of the Module Treating Field Data	76
5.5.1	Structure of Field Data.....	76
5.5.2	Sample (Mesh Form)	77
5.5.3	Setup of Project Environment	78
5.5.4	Creation of the Module Treating Field Data	78
5.5.5	Modular Preservation.....	85
5.5.6	Registration to Project Environment	85
5.5.7	Execution Test.....	86
5.5.8	Put Together as UI Module	86
5.5.9	Macroscopic-izing.....	87
6.	The Help Function of AVS/Express	88
6.1	How to Use an On-line Document	88
6.1.1	Presenting of Module Information	88
6.1.2	Display of the On-line Document for Every Module	89
6.1.3	Display of All On-line Documents	89
6.2	Starting Demonstration Application	90
7.	Operation Procedure of AVS/ExpressMPE	105
7.1	Preparation of Hardware.....	105
7.1.1	Setting Check of a Picture Server (Onyx3200)	105
7.1.2	Preparation of Circumference Hardware	105
7.1.3	Starting of the Daemon (Trackd) for Sensing Data Acquisition.....	106
7.2	Usage of AVS/ExpressMPE.....	107
7.2.1	Starting of AVS/ExpressMPE	107
7.2.2	Instance of the Viewer Module Only for MPE	107
7.2.3	Stop of a Status Display.....	108
7.2.4	Setting Change and Starting of a WAND Parameter	108
7.2.5	Change in Rendering Mode.....	109
7.3	AVS/ExpressMPE Control Method	109
7.3.1	Wearing of the Glasses for Seeing in Three Dimensions	109
7.3.2	Operation Method of a WAND Controller	110
7.4	Exiting AVS/ExpressMPE.....	111
7.4.1	Change of Window.....	111
7.4.2	Change in Rendering Mode.....	111
7.4.3	Stop of a [WandViewer3D2] Module	111
7.4.4	Exiting AVS/ExpressMPE.....	111
7.5	End of the Demon (Trackd) for Sensing Data Acquisition	111
Acknowledgements.....		112
References.....		112
Appendix Index.....		113

はじめに

近年、ネットワーク技術の高度化とワークステーションやパソコン用コンピュータの高性能化により、シミュレーションや実験の結果をワークステーション上において可視化ツールやユーティリティを用いて3次元として画像を容易に表示できるようになっています。情報システム管理課では、従来大型計算機で行なっていた画像処理やファイル編集などを、ワークステーションなどの小型の計算機へ移行させ、作業の効率化を図る分散処理環境の整備を進めており、シミュレーション結果の可視化のため画像処理サーバ、カラープリンタなどの整備を行なっています。この画像処理環境の整備作業の一環として、汎用可視化ツール AVS/Express が画像処理サーバに導入されています。AVS/Express は、シミュレーションや実験などの結果（数値データ）を、直接プログラミングすることなしに GUI 操作で可視化することができるソフトウェアです。

本報告書では、分散処理環境の整備作業の一環として整備した画像処理環境のうち、可視化ソフトウェア AVS/Express の利用方法を中心に解説します。合わせて AVS/ExpressMPE を使った立体視表示装置の利用方法について述べています。

以下に本報告書の概要を記述します。

AVS/Express の起動について第1章で、ネットワーク・エディタの基本操作について第2章で、データビューワの操作方法について第3章で、AVS/Express で扱うデータ（主にフィールドデータ）のフォーマットについて第4章で述べます。第5章ではモジュールの開発方法について、第6章ではヘルプ機能について、第7章では AVS/ExpressMPE を用いた立体視表示の方法について述べます。また本報告書の内容については、コンソール及び X サーバソフトウェアを導入した PC で動作を確認しています。

1. AVS/Express の起動と停止

1.1 AVS/Express を利用するための環境

AVS/Express の利用は、原則的に画像サーバである Onyx コンソールにおいて行います。利用に際しては、情報システム管理課 利用支援係において予約を必要とします。画像サーバの利用に関しては、利用申請が必要になりますので、情報システム管理課にお尋ね下さい。また、Onyx にインストールされている AVS/Express は 5 ライセンスあり、ネットワークを介して同時に 5 人のユーザが利用することが可能です。ただし、ネットワークを介して AVS/Express を利用する場合にはネットワークの負荷を意識してご利用下さい。

(1) ハードウェアに関連する環境

■ 東海研画像処理サーバ

ホスト名	:	tgsv
IP アドレス	:	133.53.244.71

■ 那珂研画像処理サーバ

ホスト名	:	naka-ox1
IP アドレス	:	157.111.162.17

(2) AVS/Express の起動に必要な環境設定

以下に、AVS/Express の起動に最低限必要な環境変数を示します。

```
LM_LICENSE_FILE = /usr/local/flexlm/licenses/license.dat
XP_PATH = /appli/Express/express
MACHINE = sg6
LD_LIBRARY_PATH = /appli/Express/express /lib/sg6
```

必要な環境変数は、各ユーザの.cshrc ファイルや.login ファイル等、個人環境設定ファイルに設定されると便利です。

また、雛型となる環境変数設定ファイルが、/etc/cshrc-express にあります。

1.2 AVS/Express の起動

AVS/Express の起動方法は、Onyx にログインした後、UNIX シェル上で行います。ログイン時に UNIX シェルウィンドウが起動されていない場合は、ディスクトップの UNIX シェルで UNIX シェルウィンドウを起動して下さい。コンソール以外で起動する場合はリモートでログインした後、ディスプレイ変数を設定 (setenv DISPLAY xxx.xxx.xxx.xxx:0.0 等) しておきます。

(1) 以下のコマンドを入力して、AVS/Express を起動します。

```
% /appli/Express/express/bin/sg6/xp
```

コンソール以外で起動する場合は、ハードウェアレンダリングを無効とするオプション (-nohw) を付けて起動します。

```
% /appli/Express/express/bin/sg6/xp -nohw
```

ウインドウには以下のメッセージが表示されます。

AVS/Express Developer Edition	-----①
Version: 5.1 fcs sg6	-----②
Project: /appli/Express/express/bin/sg6/.../... (read-only)	-----③

- ① AVS/Express の名称です。
- ② バージョンが表示されます。
- ③ プロジェクトディレクトリが表示されます。

AVS/Express を起動すると、図 1-1 のような起動するアプリケーションのタイプを指定するダイアログが表示されます。

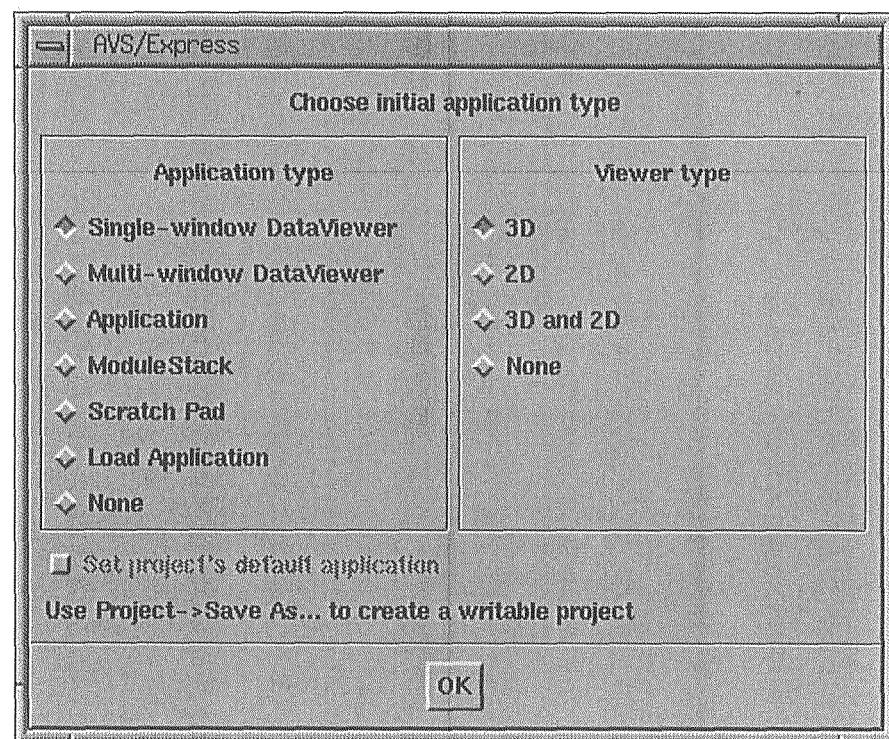


図 1-1 AVS/Express の起動画面

1.2.1 アプリケーション・タイプ

AVS/Expressには、以下のアプリケーション・タイプとビューワ・タイプがあります。

図1-1にて、必要に応じたアプリケーション・タイプとビューワ・タイプを指定したのち、OKボタンをクリックします。

■ アプリケーション・タイプ

- Single-window DataViewer : 表示ウインドウとコントロールパネルが1つのウインドウに組み込まれたデータビューワのワークスペース
- Multi-window DataViewer : (図1-2) 表示ウインドウとコントロールパネルが独立したデータビューワのワークスペース
- Application : データビューワのないワークスペース。表示ウインドウ、コントロールパネル、データ入力など全ての機能を作成するための環境
- ModuleStack : モジュールをインスタンスし、可視化ネットワークを構築するためのワークスペース
- Scratch Pad : モジュールを作成するためのワークスペース
- Load application : 一度保存したアプリケーションを読み込む
- None : 何もない状態

■ ビューワ・タイプ

- 3D : 3次元データを表示するウインドウ
- 2D : 2次元イメージ及びグラフを表示するウインドウ
- 3D and 2D : 両方を表示するウインドウ
- None : ウインドウなし

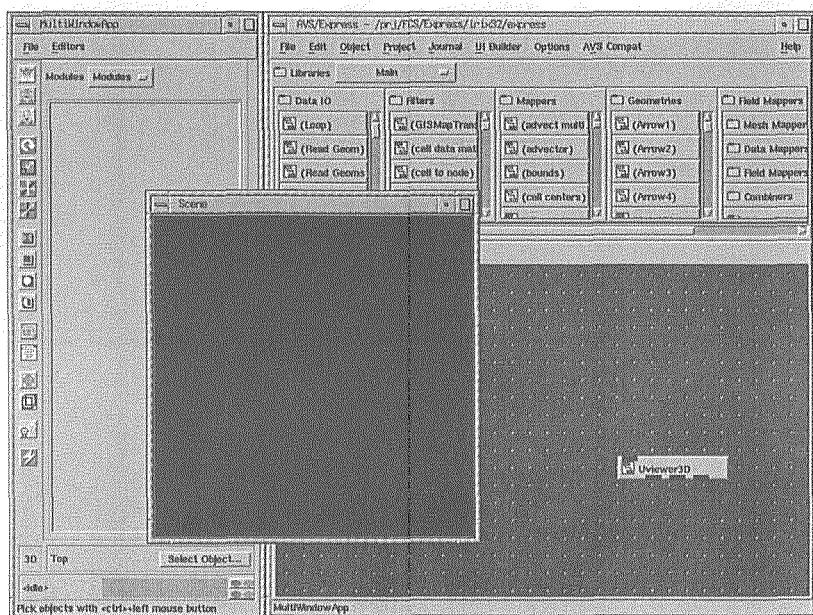


図1-2 AVS/Express 初期画面 (例: Multi-Window Data Viewer)

1.2.2 ネットワーク・エディタメニュー

可視化作業において、ユーザの作成したデータを読み込んで実際に可視化したイメージを表示するまでの過程を大きく(1)から(4)の行程に分けて考えます。

- (1) データを読み込む。
- (2) データを別のデータに変換する。情報を加えたり、減らしたりする。
- (3) 表示用のデータに変換する。
- (4) データを表示する。

AVS/Express では、上にあげた可視化までの1つ1つの行程をモジュールとよばれる部品を組み合わせることによって表現します。

AVS/Express では、この可視化の一連の作業をモジュールの組み合わせで表現したものernetワークと言い、モジュールを組み合わせてネットワークを作成するための道具としてネットワーク・エディタを用います。

AVS/Express 起動時、画面右側に表示されるパネルがネットワーク・エディタです。(図1-3)

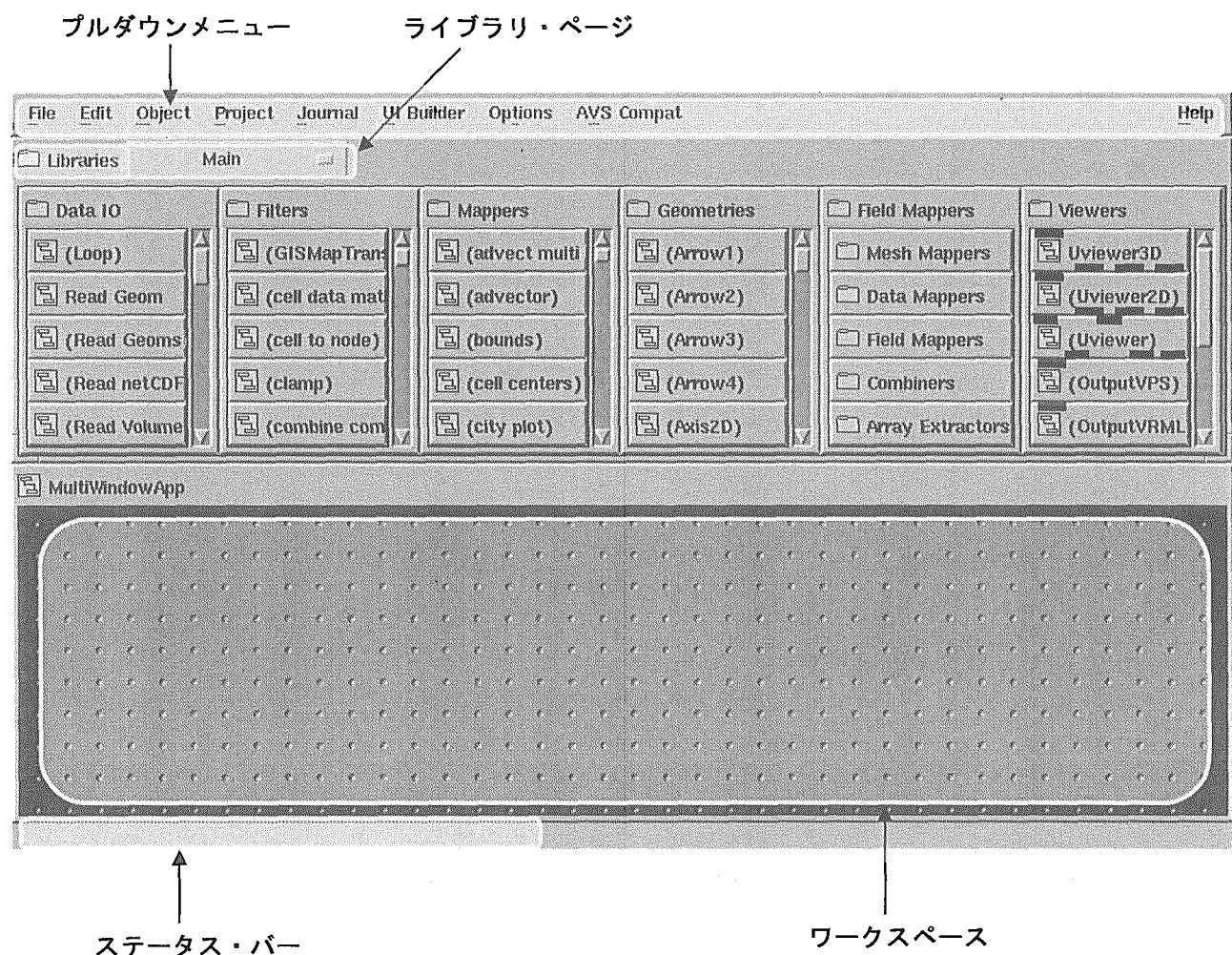


図 1-3 ネットワーク・エディタのパネル

ネットワーク・エディタの機能

ネットワーク・エディタは、以下の4つの機能を持っています。

a) プルダウンメニュー

アプリケーションの保存や読み込み、各オブジェクトに対する処理などが登録されています。

File	: アプリケーション関連の操作を行ないます
Edit	: モジュールやアプリケーションの編集を行ないます
Object	: オブジェクトに関する操作を行ないます
Project	: プロジェクトに関する操作を行ないます
Journal	: 実行手順の保存、再実行を行ないます
UI_Builder	: ユーザインターフェースのレイアウトに関する操作を行ないます
Option	: その他の操作を行ないます
AVS_Compat	: AVS5に関する操作を行ないます
Help	: ヘルプに関する操作を行ないます

b) ライブラリ・ページ

各オブジェクト（モジュール）が登録されています。

■ モジュールライブラリの種類

Main	: 可視化のための部品
Accessories	: 可視化以外の様々な部品
Standard Objects	: 最小構成単位となる部品
User Interface	: ユーザインターフェースを構築するための部品
Graphics Display	: ビューワーを構築するための部品
Annotation Graphing	: グラフやセンターを構築するための部品
Visualization	: 幾何変換処理を行なうための部品
Imaging	: イメージ処理を行なうための部品
AVS5 Modules	: AVS5モジュール
Examples	: サンプルアプリケーション
Library Workspaces	: ユーザ作成モジュールを格納するライブラリ
Templates	: テンプレート部品

■ Main ライブラリのカテゴリ

Data IO	: データの入出力を行なう部品
Filters	: データの加工を行なう部品
Mappers	: 表示データの生成を行なう部品
Geometries	: 矢印、線などの形状部品
Field Mappers	: Fieldデータを作成する部品
Viewers	: 表示部品

c) ワークスペース

オブジェクトの接続によって可視化ネットワークを作成する作業領域です。

d) ステータス・バー

操作に対するヘルプなどの情報が表示されます。

1.2.3 モジュール構成

モジュールをワークエリアに配置すると図 1-4 のように表示されます。(配置方法については 10 ページを参照してください)

AVS/Express を起動した状態のライブラリにあるモジュールは、括弧()で括られていて入出力ポートは付いていないように見えます。このような状態のモジュールは、名前はエントリしていても本体はロードされていないものです。これらは必要なモジュールを必要なときに起動するように工夫されているためです。



図 1-4 モジュール表示例 (Read_Field モジュール)

入出力ポートの説明

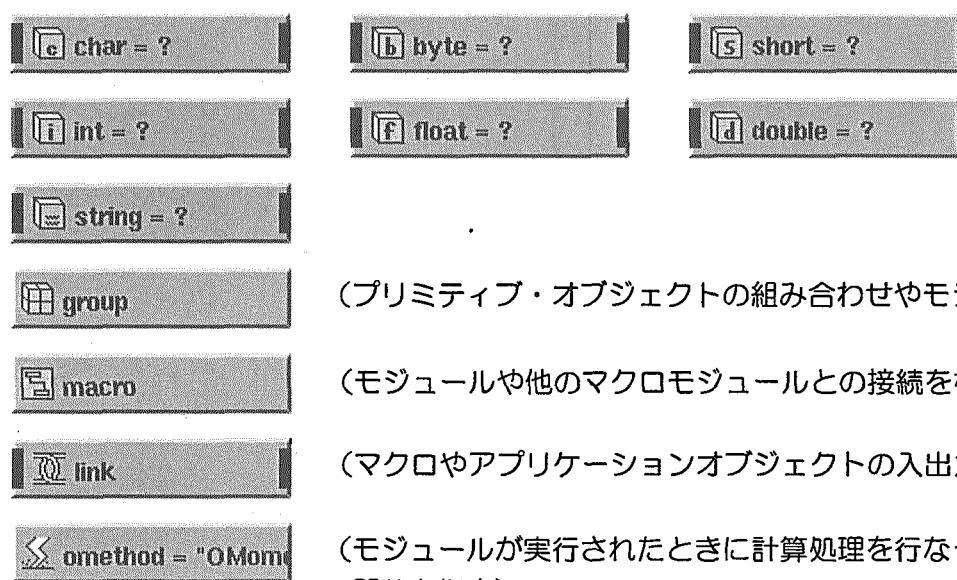
入出力ポートの色は、そのモジュールが受け取ることのできるデータタイプを示しています。つまり、モジュール同士でデータのやり取りを行なう場合は、同じ色のついたポートを接続する必要があります。

ポートの色はモジュール・プロパティを設定することで変更することができます。ここでは、AVS/Express の代表的なポートの色を紹介します。

- | | |
|--------|---------------------------------------|
| 緑 | : UIconnection (ユーザインターフェースの接続を意味します) |
| 赤 | : DefaultObject (3 次元オブジェクトを意味します) |
| マルチカラー | : Field (Field データを意味します) |

モジュール・アイコンの説明

モジュール・アイコンは、AVS/Express で定義されているプリミティブ・オブジェクトとグループやマクロ、メソッド等のオブジェクトによって分類されています。(図 1-5)



(プリミティブ・オブジェクトの組み合わせやモジュールの定義)

(モジュールや他のマクロモジュールとの接続を格納)

(マクロやアプリケーションオブジェクトの入出力の接続)

(モジュールが実行されたときに計算処理を行なう C,C++,FORTRAN の関数を指定)

図 1-5 プリミティブ・オブジェクト

1.3 AVS/Express の停止

ネットワーク・エディタのプルダウンメニューから [File]-[Exit] を選択します。図 1-6 のようなパネルが表示されるので、[OK] ボタンを選択してください。

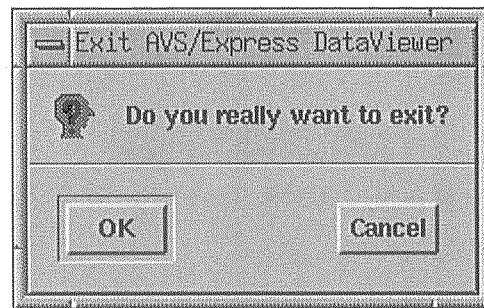


図 1-6 AVS/Express の終了確認パネル

2. ネットワーク・エディタの基本操作

この章では、モジュールの接続、切断などネットワーク・エディタの基本操作について説明します。まず、AVS/Express を起動してください。

"Choose initial application type" パネルから [Multi-window Data Viewer] と [3D and 2D] を選択し、[OK] ボタンを押してください。

"Standard Objects" モジュールライブラリにある単純な部品を使ってネットワーク・エディタの基本操作について説明します。

2.1 モジュールの検索

モジュール名の一部がわかっている場合は、名前をもとに検索し、表示させることができます。

□ 全てのライブラリを検索する場合

プルダウンメニューの [Object]-[Find in All Libraries…] を選択すると、"Object Finder" パネルが表示されます。(図 2-1) この中の [Search Pattern] の部分に検索したい名前またはその一部を入力します。

検索時に入力する名前は大文字、小文字の区別はしません。また、* や ? のワイルドカードが使用できます。

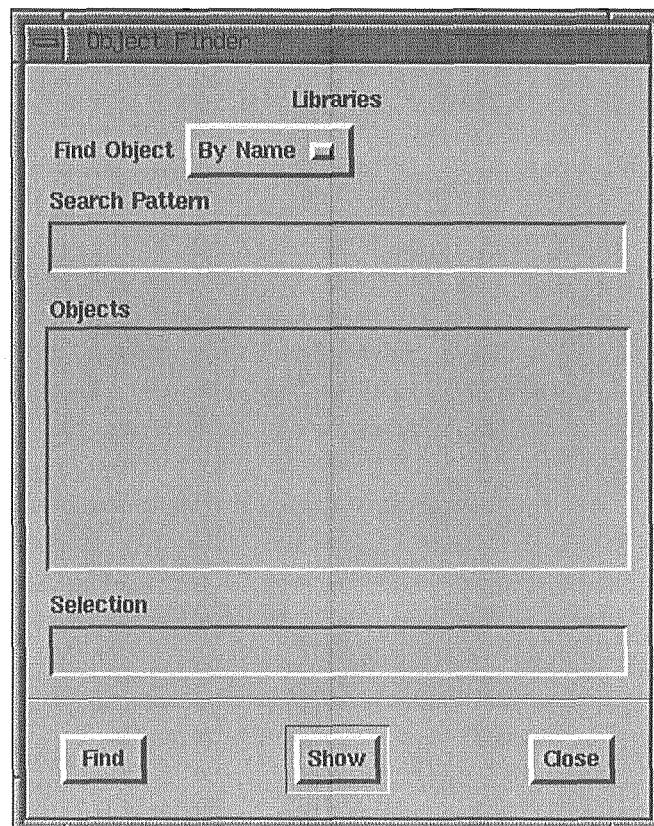


図 2-1 "Object Finder" パネル

□ 特定のライブラリを検索する場合

モジュールライブラリを検索したいライブラリに変更します。
 検索したいライブラリメニュー上でマウス右ボタンを押したままポップアップメニューを表示し、メニューの中の[Find…]を選択すると"Object Finder"パネルが表示されます。
 この中の[Search Pattern]の部分に検索したい名前またはその一部を入力します。

例："int"モジュールの検索

"int"という名前のモジュールを検索します。

1. プルダウンメニューから [Object]-[Find in All Libraries…] を選択してください。
2. [Search Pattern] の領域に"int"と入力しリターンキーを押すと、名前が int であるモジュール・リストが [Objects] の領域に表示されます。この中の "Standard Objects.Parameters.int"を選択して、[Show] ボタンを押してください。"Standard Objects"ライブラリが表示され、Parameters カテゴリに "int"モジュールが選択された状態になります。
3. "Object Finder"パネルの"Close"ボタンを選択し、ウインドウを閉じます。

2.2 モジュールのインスタンス

ライブラリの中にあるモジュールを使用するには、ワークスペース上に移動して利用します（図 2-2）。

モジュールを移動するには、移動したいモジュールアイコン上でマウス左ボタンを押したままワークスペースにドラッグし、マウス左ボタンを離します。

ライブラリの中からワークスペースに移動することを「インスタンスする」と呼ぶこともあります。

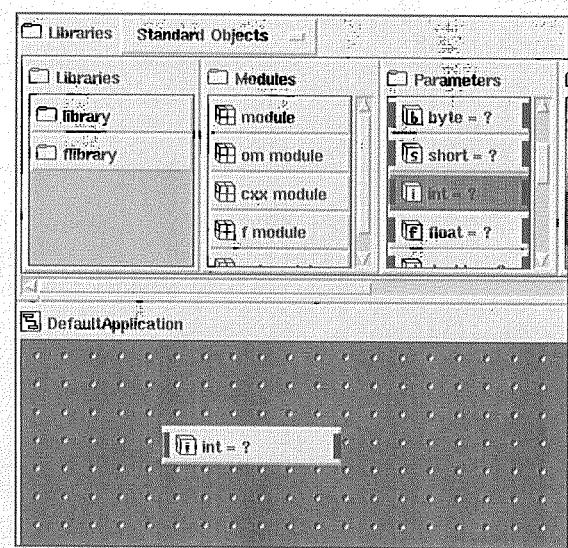


図 2-2 モジュールのインスタンス

同じ名前のモジュールを複数インスタンスした場合は、モジュール名の末尾に #1, #2, #3 が付きます。これは、AVS/Express がモジュールを管理する時に名前で管理しているためです。ただし、同一階層でなければ同じ名前のモジュールが複数存在しても構いません。

2.3 モジュールの消去

□ 単体モジュールの場合

消去したいモジュール上でマウス右ボタンを押したままポップアップメニューを表示し、[Delete]を選択します。(または、消去したいモジュールを選択し、プルダウンメニューの [Edit]-[Delete] を選択します)

□ 複数モジュールの場合

消去したいモジュールをマウス左ボタンを押したまま囲みます。消去したいモジュールが選択されていることを確認し、そのままプルダウンメニューの [Edit]-[Delete] を選択します。

2.4 モジュール名の変更

モジュール名を変更する場合は、名前を変更したいモジュール上でマウス右ボタンを押したままポップアップメニューを開き、その中の [Rename...] を選択します(または、モジュールを任意のマウスボタンで選択し、プルダウンメニューの [Object]-[Rename...] を選択します)。

"Rename Object"パネルが表示されるので、入力領域に名前を入力します。(図 2-3)

デフォルトの状態ではすでに文字が入力されているので、入力されている文字を消去してから名前を入力し、リターンキーを押します。

注：同じ階層で同一名のモジュールを存在させることはできません。また、数字で始まるモジュール名をつけることはできません。

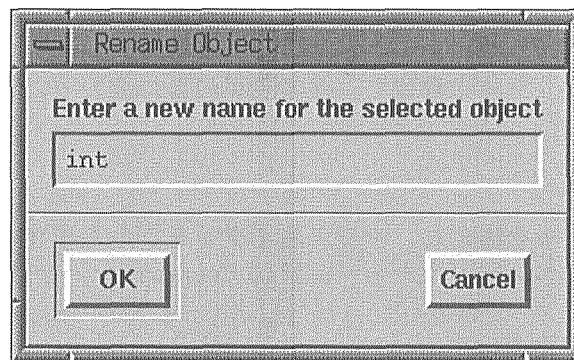


図 2-3 "Rename Object"パネル

2.5 モジュールの接続

□ マウスによる接続方法

カーソルを接続したいモジュールの出力ポート上に移動し、マウス左ボタンを押したままカーソルを移動します。接続可能なモジュールが細い線で表示されるので、接続したいモジュールの方向にマウス左ボタンを押したまま移動し、線の色が白色に変わったところでマウス左ボタンを離します。

□ タイプインによる接続方法

接続したいモジュールの入力側のモジュール上でマウス左ボタンをダブルクリックしてモジュールを開きます。開いた箱の中で以下のように参照したいモジュール名を入力し、リターンキーを押します。

=> .int
モジュール名

2.6 モジュールの開閉

□ 開く方法

モジュール上でマウス左ボタンをダブルクリックします。(または、モジュール上でマウス右ボタンを押したままポップアップメニューを開き、メニューから [Open]を選択します。)

□ 閉じる方法

モジュール上でマウス左ボタンをダブルクリックします。(または、モジュール上でマウス右ボタンを押したままポップアップメニューを開き、メニューから [Close]を選択します。)

2.7 モジュールの切断

□ マウスによる切断方法

マウスによる切断方法には、ポート上で行なう方法と接続ライン上で行なう方法があります。

ポート上で行なう場合：

カーソルを切断したいモジュールの出力ポート上に移動し、マウス左ボタンを押したままカーソルを移動します。切断可能なモジュールが細い線で表示されるので、切断したいモジュールの方に向にマウス左ボタンを押したまま移動し、線の色が白色に変わったところでマウス左ボタンを離します。

接続ライン上で行なう場合：

接続線上にマウスカーソルを移動し、マウス右ボタンを押したままポップアップメニューを表示します。メニューの中から [Delete Connection] を選択します。

□ タイプインによる切断方法

入力側のモジュール上でマウス左ボタンをダブルクリックしてモジュールを開き、そこに記述されている参照の情報(どのモジュールと接続されているかを記述している)をバックスペースキーですべて消去し、リターンキーを押します。

2.8 モジュールの移動

ワークスペース内のモジュールを配置し直す場合は、移動したいモジュール上でマウス左ボタンを押したまま移動します。

複数のモジュールを同時に移動したい場合は、移動したいモジュールを選択します。選択方法には、矩形領域で囲む方法とマウスで選択する方法があります。

矩形領域で囲む場合：

ワークスペース上でマウス左ボタンを押したまま右下方向に移動すると矩形領域が表示されます。この矩形領域で移動したいモジュールを囲みます。

マウスで選択する方法：

Shift キーを押したまま、移動したいモジュール上でマウス左ボタンを押します。

選択(青く表示)されているモジュール上にマウスカーソルを合わせ、マウス左ボタンを押したまま移動します。

青く表示されているモジュール以外のところでマウス左ボタンを押してしまうと、選択状態が解除されてしまいます。

2.9 ワークスペース上のモジュール操作

□ ポップアップメニューによる操作

ワークスペース上でマウス右ボタンを押すとポップアップメニューが表示されます。このメニューには次のような機能があります。

- | | |
|---------------|------------------------------|
| Reset Scaling | モジュールの大きさをリセットします。 |
| Arrange Icons | モジュールの配置を整頓します。 |
| Zoom to Fit | モジュールの大きさをワークスペースの大きさに合わせます。 |

ワークスペース上にモジュールをインスタンスしているのにモジュールが見えなくなってしまった場合は、Arrange Icons を選択するとワークスペースの中心にモジュールが配置されます。(図 2-4)

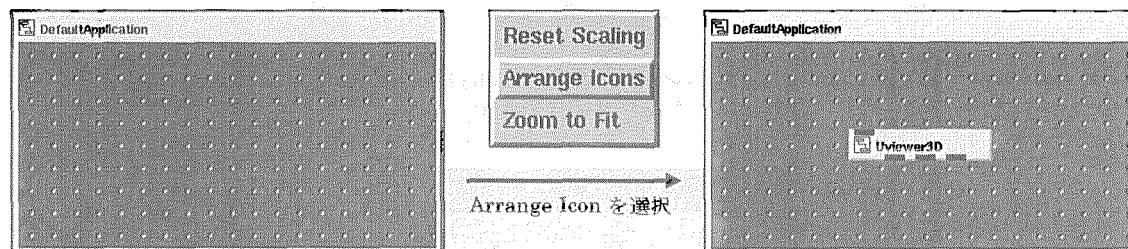


図 2-4 Arrange Icons 機能

□ マウスによるモジュールの移動

Shift キーを押したまま、ワークスペース上でマウス中ボタンを押します。

そのままマウスを移動するとワークスペース上の全てのモジュールが平行移動します。

Ctr1 キーを押したまま、ワークスペース上でマウス中ボタンを押します。

そのままマウスを移動するとワークスペース上の全てのモジュールが拡大または縮小されます。

2.10 モジュールの情報

モジュールの名前を変更した場合、元が何というモジュールだったのか分らなくなる場合があります。

そのような場合は、“Object Info”パネルによって確認することができます。(図 2-5)

“Object Info”パネルを表示するには、モジュール上でマウス右ボタンを押したままポップアップメニューを表示し、メニューの中から [Info…] を選択します。

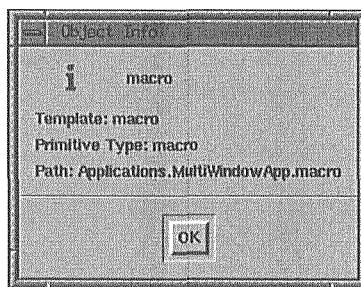


図 2-5 "Object Info" パネル

2.11 アプリケーションの保存

"MultiWindowApp"の名前を変更し、アプリケーションとして保存します(アプリケーションの保存とは、ワークスペースごと保存することを意味します)。

"MultiWindowApp"と書かれているグレー部分上でマウス右ボタンを押したままポップアップメニューを開きます。メニューから [Rename...]を選択し、名前を変更します。

次にプルダウンメニューから [File]-[Save Application...]を選択してください。ワークスペースが閉じられた状態でアプリケーションの保存を行なおうとしても、プルダウンメニューの [File]-[Save Application...] の部分が灰色になっているため保存できません。アプリケーションを保存する場合は、必ず、そのワークスペースを開いた状態にしておく必要があります。

[File]-[Save Application...]を選択すると"Save Application"パネルが現れます。(図 2-6) この中で保存するディレクトリとファイル名を指定します。"Filter"の入力領域に保存するディレクトリ名を入力してください。 /tmp 配下に"test.v"と保存する場合は、"Filter"領域に"/tmp/*.v"と入力します。

"Selection"領域には"test.v"と入力し、リターンキーまたは[OK]ボタンを押します。アプリケーションを保存する場合は、拡張子に ".v" を付けます。

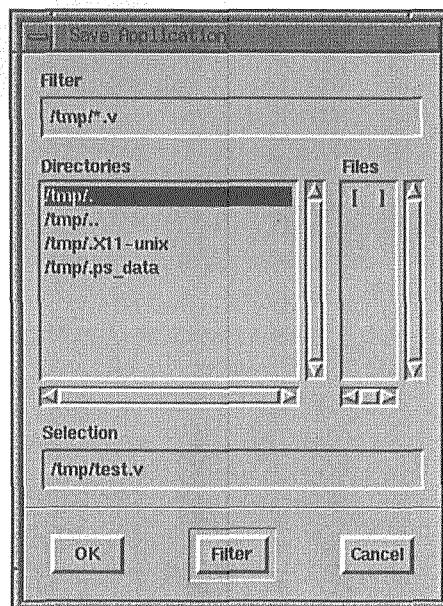


図 2-6 "Save Application" パネル

保存した"test.v"ファイルには次のように記述されています(モジュールを配置した位置によって若干の違いがあります)。

```
APPS. MultiWindowApp test<NEdisplayMode="maximized"> {
    UI {
        Modules {
            IUI {
                mod_panel {
                    x = 0;
                    y = 0;
                };
            };
        };
    };
    GDM. Uviewer Uviewer<NEx=603, NEy=453>;
    int a<NEportLevels=1, NEx=341., NEy=99.>;
    int b<NEportLevels=1, NEx=528., NEy=99.> => .a;
};

```

2.12 アプリケーションの削除

作成したアプリケーション(ワークスペース)を削除します。アプリケーションを削除する場合は、ワークスペースを開いた状態で行います。

プルダウンメニューから[File]-[Delete Application]を選択します。

注：アプリケーションを削除する場合は、削除しても良いか確認するダイアログは表示されませんのでご注意ください。

2.13 アプリケーションの読み込み

保存したアプリケーションを読み込みます。

プルダウンメニューから [File]-[Load Application…] を選択すると、“Load Application”パネルが現れます。 “/tmp/test.v”を読み込む場合は。“Filter”領域に“/tmp/*.v”と入力し、リターンキーを押します。表示された“test.v”を選択した後、[OK]を選択すると“test”ワークスペースの中に先程保存したネットワークが読み込まれます。

注：アプリケーションを読み込む時には、読み込む先を指定する必要はありません。

3. データビューワ

データビューワとは、可視化を行なう場合にモジュールのパラメータの指定やグラフィックスの制御を行なうためのユーザインターフェースのことです。

この章では、データビューワの操作方法について説明します。

3.1 データビューワの起動画面

すでにアプリケーションが読み込まれている場合は、アプリケーションを消去します。
 ネットワーク・エディタのプルダウンメニューから [File]-[New Application...] を選択してください。“New Application”パネルが現れます。
 このパネルの中の [Application type] から [Multi-window DataViewer] を選択し、さらに [Viewer type] から [3D] を選択して [OK] ボタンを押してください。

図 3-1 のようにワークスペースの中には“Uviewer3D”モジュール、それ以外の場所には、表示ウインドウやデータビューワのコントロールパネルが現れます。

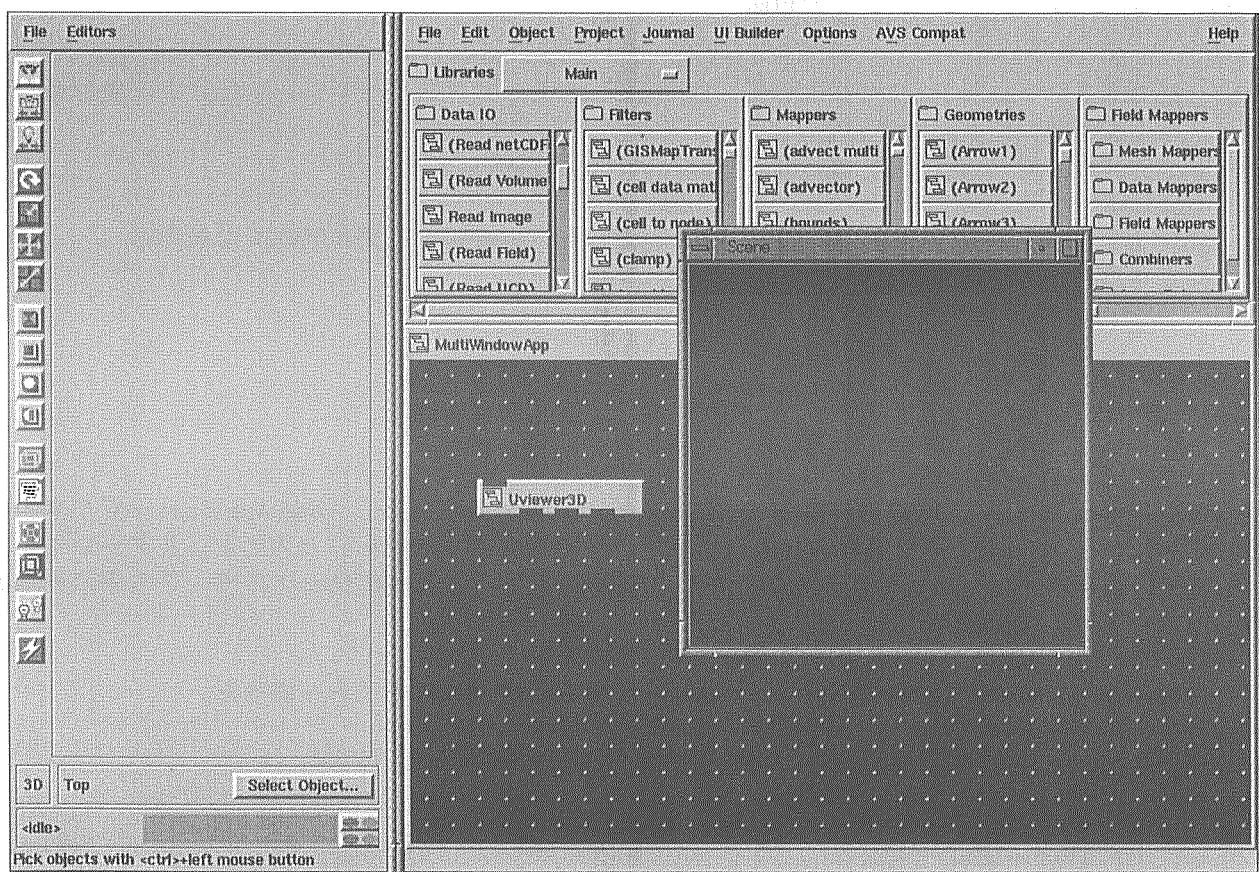


図 3-1 データビューワ

3.2 コントロールパネル

データビューワを起動して表示されたウインドウの左側の部分がコントロールパネルです。(図3-2) プルダウンメニュー、ツールバー、"Select Object..."ボタン、ステータスバーがあり、画像をコントロールすることができます。各項目については3.2.1以降で説明します。

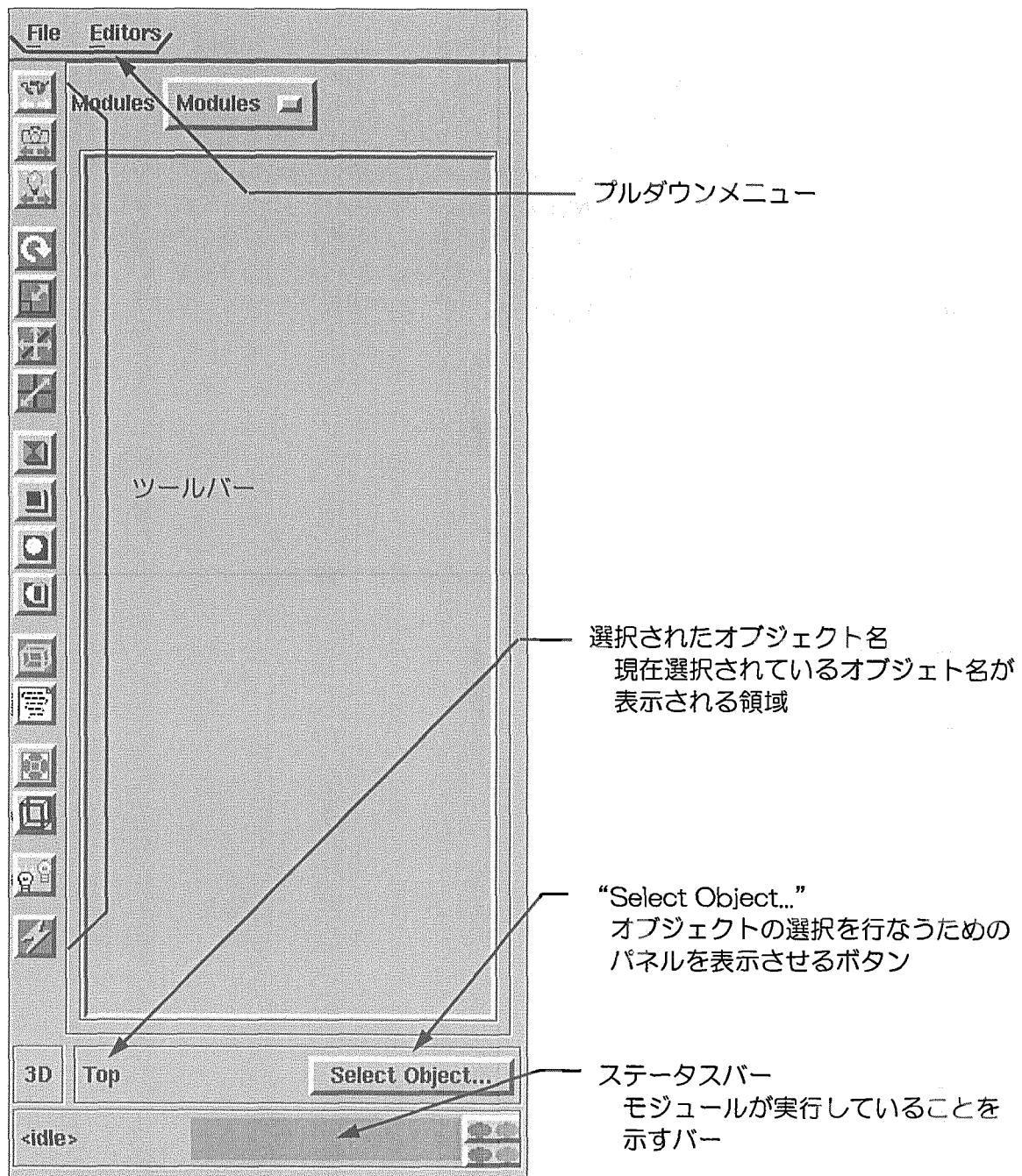


図3-2 コントロールパネル

3.2.1 プルダウンメニュー

コントロールパネルのプルダウンメニューには、以下の機能があります。

File	
Exit	AVS/Express の終了
Editors	
Modules	モジュールに関するメニュー(モジュール・エディタ)を表示
View	ビューに関するメニュー(ビュー・エディタ)を表示
Transform	幾何変換に関するメニュー(トランスフォーム・エディタ)を表示
Light	ライトに関するメニュー(ライト・エディタ)を表示
Camera	カメラに関するメニュー(カメラ・エディタ)を表示
Object	オブジェクトに関するメニュー(オブジェクト・エディタ)を表示
Datamap	データマップに関するメニュー(データマップ・エディタ)を表示
Graph	グラフに関するメニュー(グラフ・エディタ)を表示(2D タイプのみ)
Print	プリントに関するメニュー(プリント・エディタ)を表示
Windows (Single-window DataViewer の時のみ、シーンの切替えを行ないます)	
Scene	1 番目のシーン
Scene#1	2 番目のシーン
Scene#2	3 番目のシーン

3.2.2 ツールバー

表示されている形状(オブジェクト)の操作を行なうためのボタン群です。

■ 幾何変換の対象の選択



オブジェクト



カメラ



ライト

■ マウス左ボタンによる操作対象



回転



拡大/縮小



X Y 平面内でも移動



Z 方向の移動

■ リセット/正規化/中心化



選択されたオブジェクトの位置をリセット



選択されたオブジェクトを表示ウインドウの大きさに応じてスケーリング（正規化）



選択されたオブジェクトの回転中心をオブジェクトの中心に変更（中心化）



選択されたオブジェクトのリセット、正規化、中心化を同時に行なう

■ その他



アルターネートオブジェクトのオン/オフ指定



ジッター機能のオン/オフ指定



オブジェクト表示の自動正規化のオン/オフ指定



透視表示のオン/オフ指定（デフォルトはオフ）



双方向平行光線のオン/オフ指定（デフォルトはオフ）



ハードウェア（オン）/ソフトウェア（オフ）の指定

3.3 オブジェクトの読み込み

データを読み込むには、データファイルを読み込むためのモジュールが必要です。ジオメトリデータを読み込むモジュールは"Read Geom"です。このモジュールを使用してデータを取り込みます。

3.3.1 "Read Geom"モジュールのインスタンス

モジュールライブラリを"Main"に切り替え、左端の"Data IO"カテゴリから"Read Geom"モジュールを探し出し、ワークスペース上に移動してください。

次に、"Read Geom"モジュールの赤色の出力ポートと"Uviewer3D"モジュールの赤色の入力ポートをマウス左ボタンを使って接続してください。(図 3-7)

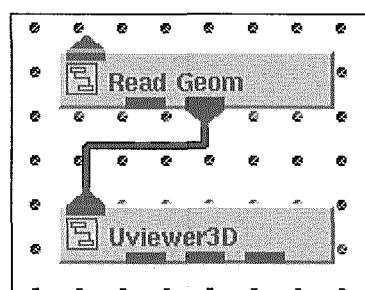


図 3-7 "Read Geom" と "Uviewer3D"

3.3.2 ファイルの指定

"Read Geom"モジュールのパラメータを指定してファイルを読み込みます。
コントロールパネルのプルダウンメニューから [Editors]-[Modules] を選択してください。モジュール・エディタが表示されます。
このモジュール・エディタの中には、インスタンスしたモジュールのパラメータが格納されます。複数のモジュールの中から特定のモジュールのパラメータを表示したい場合は、[Modules]選択メニューから必要なモジュールを選択します。

[Browse...]ボタンを押してください。"Read Geom"のパラメータには、ファイル名を直接入力する領域と、ファイルブラウザを表示するための[Browse...]ボタンがあります。
"Read GEOM Filename"ファイルブラウザが現れます。このブラウザの中の[Files]の領域から"teapot.geo"を選択し、[OK]ボタンを押すと、表示ウインドウにティーポットが表示されます。(図 3-8)

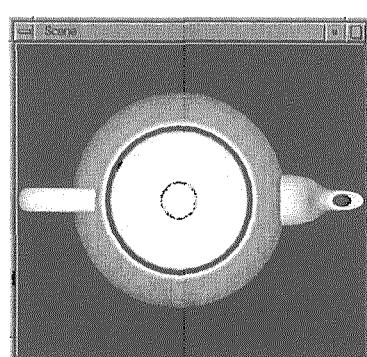


図 3-8 ティーポット・オブジェクト

3.4 オブジェクトの幾何変換

オブジェクトの幾何変換を行なうには、「マウスによる方法」と「メニューから的方法」があります。それぞれの作業中にウインドウ内のオブジェクトが見えなくなってしまった場合は、コントロールパネルの“Select Object...”から元に戻したいオブジェクトを選択し、ツールバーにあるリセットボタン()を押してください。

3.4.1 マウスによる方法

■ マウス左ボタンのみを使う場合

回転か移動かの指定などはツールバーにあるボタン (   ) を選択することで行ないます。

■ マウス中ボタンとキーボードを組合せて使う場合（3ボタンマウスのみ）

回転	中ボタン
拡大/縮小	[Shift] + 中ボタン
移動（XY平面）	[Ctrl] + 中ボタン

3.4.2 メニューからの方法

数値入力、ダイヤルによる値の設定によって幾何変換を行なうこともできます。

コントロールパネルのプルダウンメニューから [Editors]-[Transform] を選択してください。トラン

スマフォーム・エディタのメニューが表示されます。（図3-9）

この中の [Transform] 選択メニューから [Transform] を選択してください。 [Transform] のサブメニ

ューが表示されます。

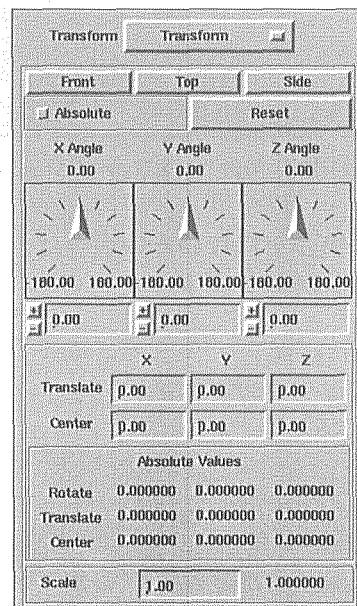


図 3-9 Transform サブメニュー

3.5 複数のオブジェクトの読み込み

複数のオブジェクトを読み込む場合は、読み込みたいオブジェクトの数だけ"Read Geom"モジュールが必要です。"Read Geom"モジュールを先程の手順で新たにもう1つ追加してください。(図 3-10)

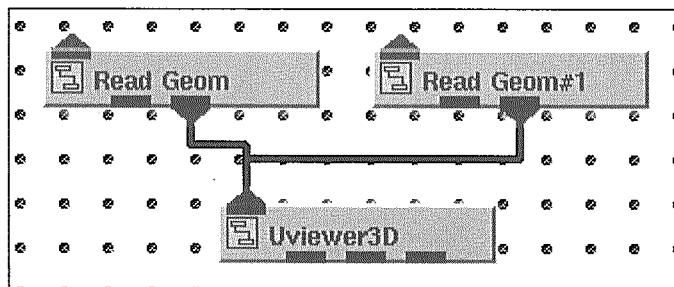


図 3-10 ネットワーク例

コントロールパネルのプルダウンメニューから [Editors]-[Modules] を選択します。

[Modules]選択メニューから2つ目に追加した"Read Geom"モジュールのパラメータである[Read Geom#1]を選択します。ファイルブラウザを表示し、"dodec.geo"を読み込むと、表示ウインドウには「ティーポット」と「12面体」の2つのオブジェクトが表示されます。

3.6 オブジェクトの選択

オブジェクトを複数表示した場合、操作対象は「個々のオブジェクト」と「全てのオブジェクト」に分かれます。「全てのオブジェクト」のことを"Top"と呼びます。"Top"の状態で幾何変換などの操作を行なうと全てのオブジェクト(ここでは teapot と dodec)に反映されます。

現在選択されているオブジェクトの名前は、コントロールパネル下部の[Select Object...]の左側に表示されます。(図 3-11)

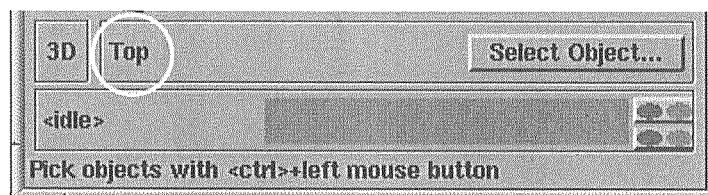


図 3-11 選択されているオブジェクト名

個々のオブジェクトに対して個別に操作を行ないたい場合は、オブジェクトの選択が必要です。オブジェクトの選択方法には、次の2通りがあります。

■ マウスによる方法

選択したいオブジェクト上にマウスカーソルを移動し、[Ctrl]+マウス左ボタンを押すと、そのオブジェクトを選択することができます。

■ 名前の指定による方法

コントロールパネルの下部にある[Select Object...]ボタンを押すと"Object Selector"パネルが現れます。このパネルの中の[Items]の領域に表示されているオブジェクト名をマウス左ボタンでクリックし、[Apply]ボタンを押すと、そのオブジェクトが選択されます。

注：[Items]からオブジェクト名を選択した後は、必ず [Apply] ボタンを選択してください。[Apply] ボタンで確定しなければオブジェクトは選択されません。(図 3-12)

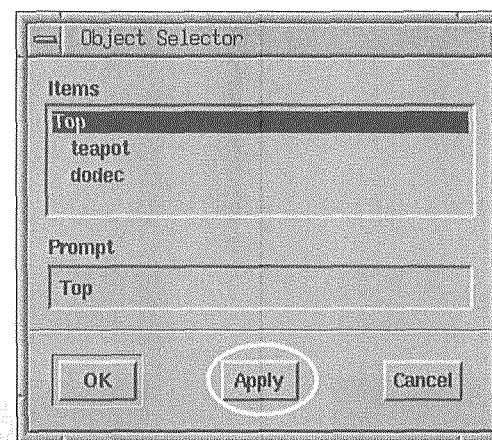


図 3-12 オブジェクトの階層

3.7 オブジェクトの属性の変更

オブジェクトの色や表示方法などを変更します。

3.7.1 オブジェクトの表示方法

オブジェクトをワイヤーフレーム表示やポイント表示にする方法について説明します。コントロールパネルのプルダウンメニューから[Editors]-[Object]を選択してください。オブジェクトエディタ・メニューが現れます。オブジェクトエディタ・メニューを[General]から[Modes]に変更すると、メニューは図3-13のようになります。

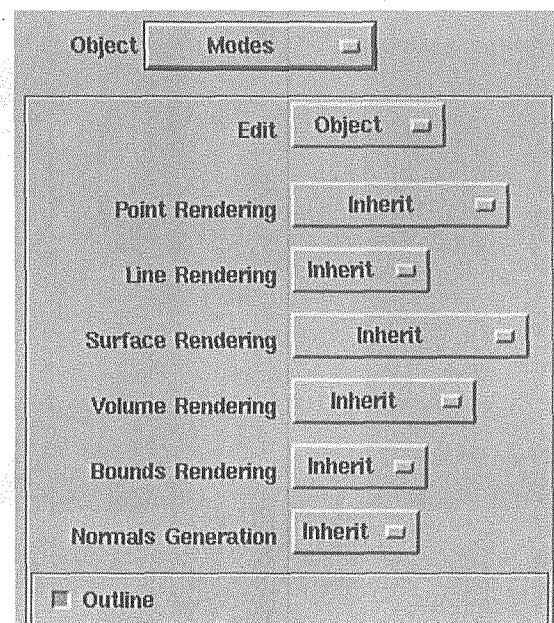


図 3-13 オブジェクト・エディタの"Modes"

「ティーポット」を選択し、"Surface Rendering"を"None"にします。
各レンダリングのデフォルトの状態は "Inherit"になっています。これは、親(ここでは Top)の状態を継承することを意味します。Surface Rendering を Inherit のままにしておくと、他の表示結果が分りにくくなるため"None"に変更します。

以下のように様々な表示ができます。(図 3-14、図 3-15、図 3-16、図 3-17)

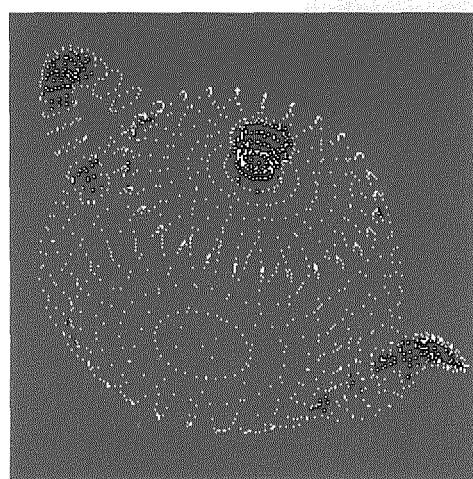


図 3-14 点表示

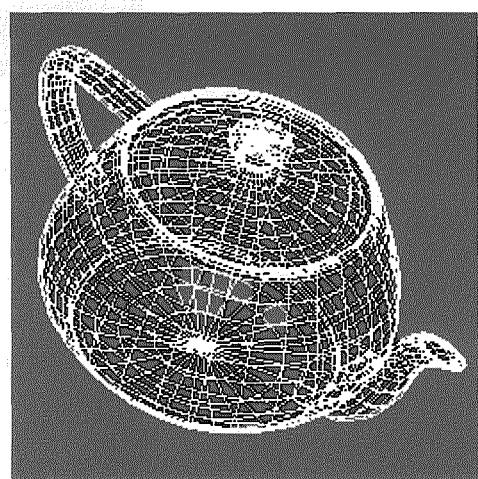
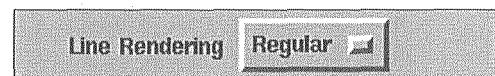


図 3-15 線表示

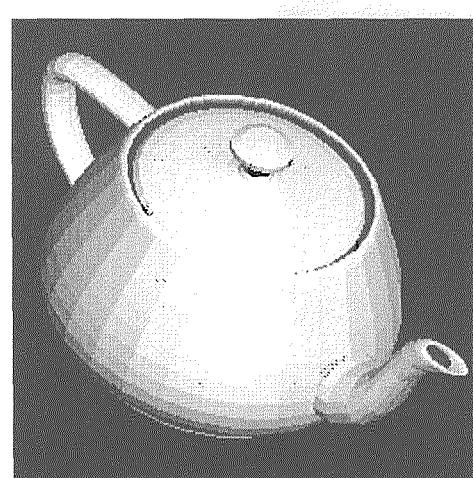


図 3-16 フラットシェーディング

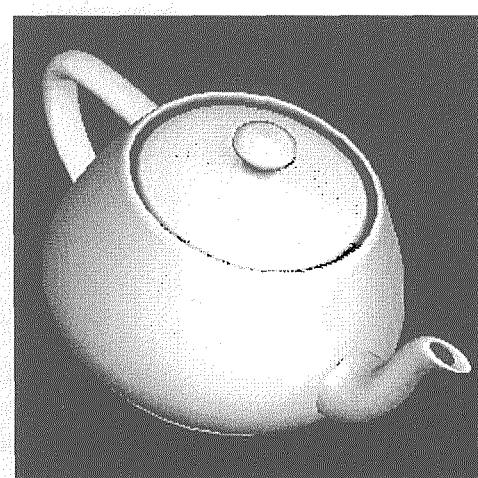
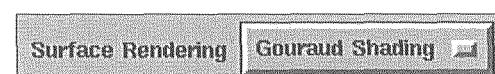


図 3-17 グーローシェーディング

【参考】ラインの幅とタイプの指定

オブジェクトをライン表示した時には、ラインの幅とタイプを指定することができます。オブジェクトエディタ・メニューから [Properties] を選択します(図 3-18 ①)。メニューの中の Type 選択メニューから [Point/Line] を選択します(図 3-18 ②)。メニューは図 3-18 のように変わります。

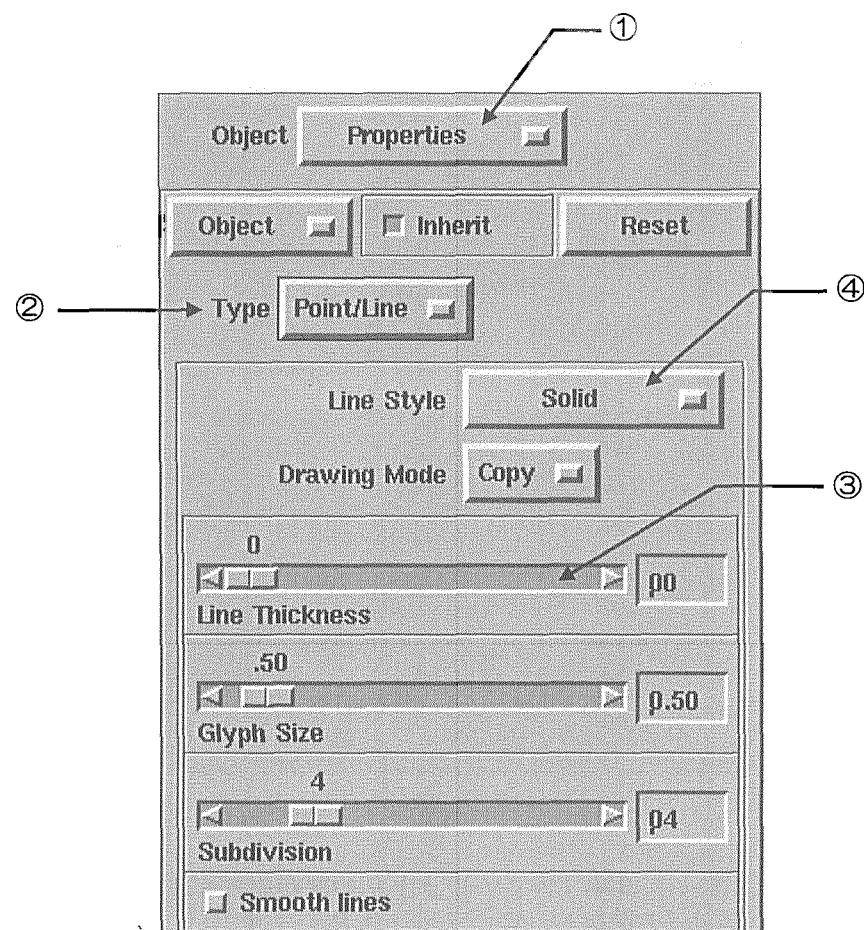


図 3-18 [Point/Line] オブジェクト・エディタ

「ティーポット」を選択し、ライン表示にします。

[Line Thickness]スライダーの値を大きくするとライン幅が太くなります(図 3-18 ③)。

ラインタイプを指定する場合は、[Line Style]から選択します(図 3-18 ④)。

3.7.2 オブジェクトの表面属性（色の指定）

オブジェクトの色を変更する方法について説明します。

オブジェクトエディタ・メニューから [Properties] を選択します(図 3-19 ①)。
 メニューの中の Type 選択メニューから [General] を選択します(図 3-19 ②)。
 メニューは図 3-19 のように変わります。

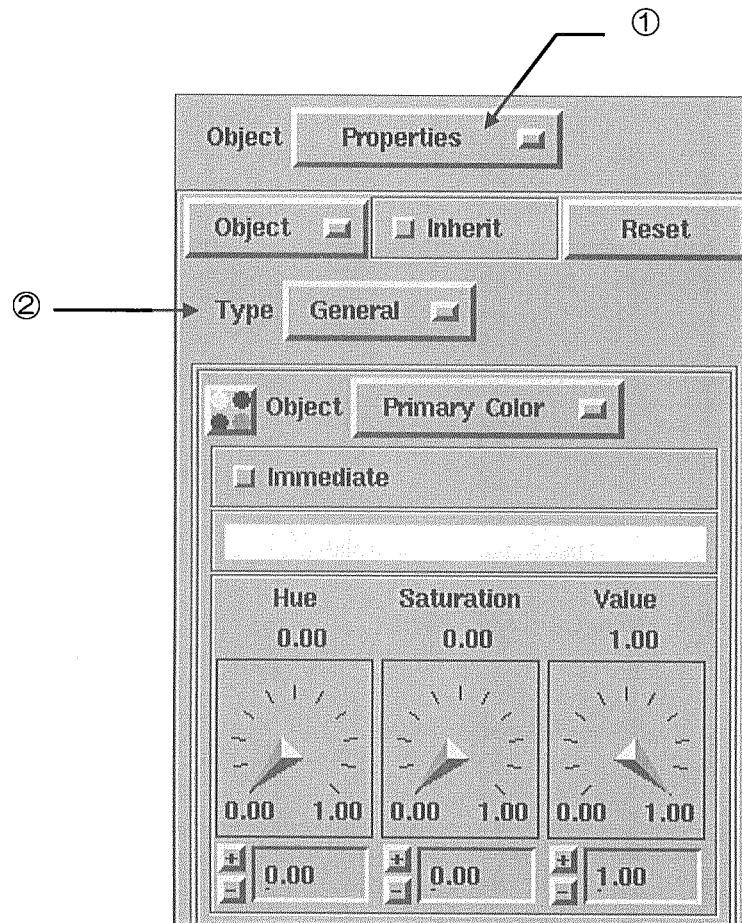


図 3-19 [General] オブジェクトエディタ・メニュー

「ティーポット」を選択します。この時「ティーポット」の表示方法が Line の場合は、Object 選択メニュー(図 3-19 ①)から [Mode] を選択し、全て Inherit にします。

メニューの中の [Hue:色相], [Saturation:彩度], [Value:明度] のダイヤル値を調節することで色を変更することができます。

□Hue, Saturation, Value による色の指定例

	Hue	Sat	Val
白	0.00	0.00	1.00
黒	0.00	0.00	0.00
赤	0.00	1.00	1.00
緑	0.33	1.00	1.00
青	0.67	1.00	1.00
黄	0.17	1.00	1.00

3.7.3 オブジェクトの表面属性 (不透明度の指定)

オブジェクトの不透明度を変更する方法について説明します。

オブジェクトエディタ・メニューから [Properties] を選択します(図 3-20 ①)。
 メニューの中の Type 選択メニューから [Surface] を選択します(図 3-20 ②)。
 メニューは図 3-20 のように変わります。

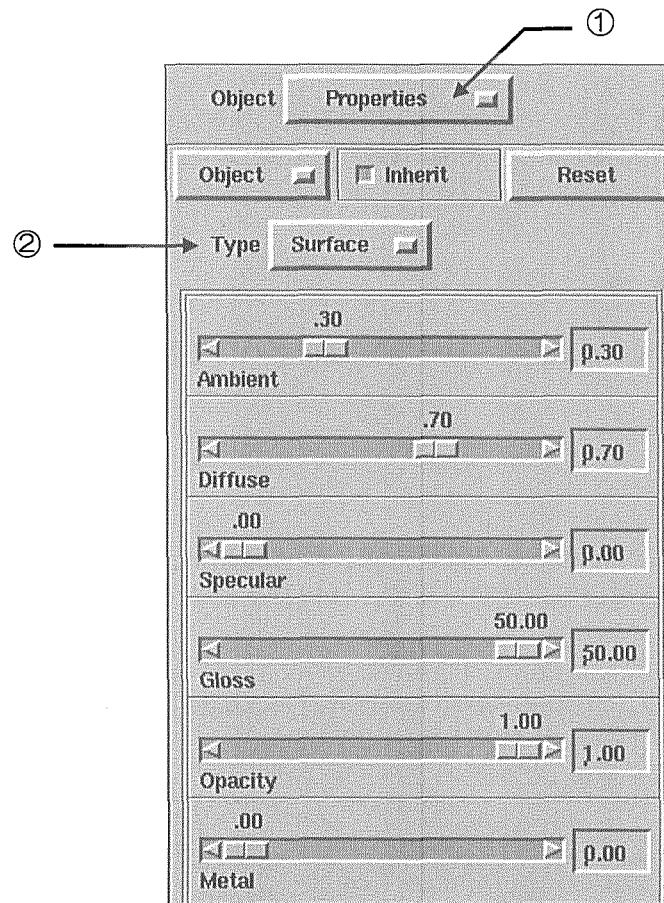


図 3-20 [Surface] オブジェクトエディタ・メニュー

各スライダーには次のような機能があります。

Ambient : アンビエント・ライト(環境光)

オブジェクトが反射するアンビエント・ライトの割合を設定します。アンビエント・ライトとは、方向性がなくサーフェスのどの部分にも同じ影響を与えるライトのことを言います。

Diffuse : ディフューズ・ライト(拡散光)

アンビエント・ライト以外のライト(平行光線、点光線など)が、各方向に反射する割合を設定します。

Specular : スペキュラー・ハイライト

視点の方向とライト(平行光線と点光線)の反射方向が近い場合にスペキュラー・ハイライトが生成されます。Specular は、スペキュラーライトの明るさを設するもので、ライティング計算時のスペキュラー係数に対応しています。

Gloss : グロス

グロスはスペキュラー・ハイライトの鋭さを設定するもので、ライティング計算時のスペキュラー係数に対応しています。値が大きいほどスペキュラー・ハイライトの範囲が小さくなり、シャープになります。

Opacity : オパシティー

オパシティーはオブジェクトの不透明度を設定します。

値が小さくなるほど透明度が増し、オブジェクトの後ろにある別のオブジェクトを透過して見ることができます。

Metal : メタル

メタルは、スペキュラー・ハイライトの色を設定します。

スライダーの値を 0(左端)にし、光源の色に 1(右端)にするとオブジェクトの色に設定されます。

3.8 カメラの属性の変更

投影法の変更方法とビューワ内でのオブジェクトの正規化について説明します。

3.8.1 投影法の変更方法

AVS/Express の投影法には「平行投影」と「透視投影」の 2 種類があります。デフォルトでは平行投影になっているので、これを透視投影に変更します。透視投影にすると、手前が大きく奥に行くほど小さく表示されます。

コントロールパネルのプルダウンメニューから [Editors]-[Camera] を選択してください。
カメラエディタ・メニューから [Lens] を選択します(図 3-21 ①)。
メニューは図 3-21 のように変わります。

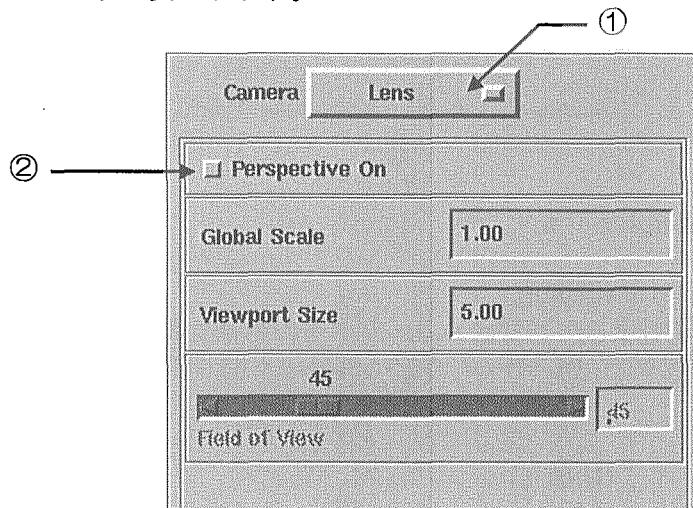


図 3-21 カメラ・エディタの“Lens”

[Perspective On] ボタン(図 3-21 ②)を選択すると透視投影になります。

コントロールパネルのツールバーにある (Perspective) ボタンをオン/オフすることでも指定できます。

3.8.2 ビューワ内の正規化

AVS/Express では、表示ウインドウ内に新たなオブジェクトが読み込まれると、表示ウインドウ内のオブジェクトは自動的にウインドウの大きさに応じて正規化されます。

この機能を解除する方法について説明します。

コントロールパネルのプルダウンメニューから [Editors]-[Camera] を選択してください。
カメラエディタ・メニューから [General] を選択します(図 3-22 ①)。
メニューは図 3-22 のように変わります。

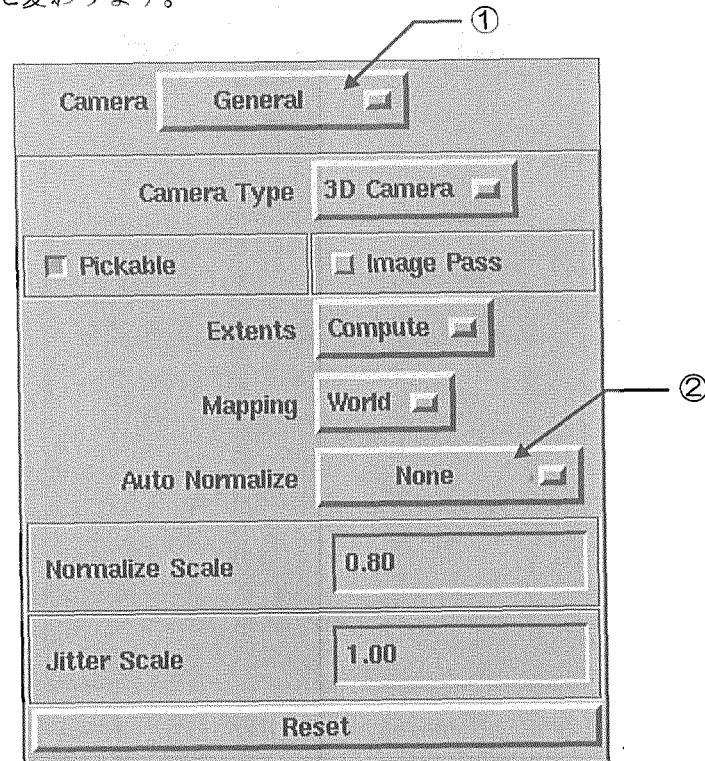


図 3-22 [General] カメラエディタ・メニュー

Auto Normalize 選択メニューから None を選択すると自動正規化が解除されます(図 3-22 ②)。

コントロールパネルのツールバーにある (AutoNormalize) ボタンをオン/オフすることでも指定できます。

3.9 ビューの属性の変更

表示ウインドウの属性を変更します。
コントロールパネルのプルダウンメニューから [Editors]-[View] を選択してください。
デフォルトでは図 3-23 のように [General] が表示されます。

3.9.1 レンダリング・メソッド

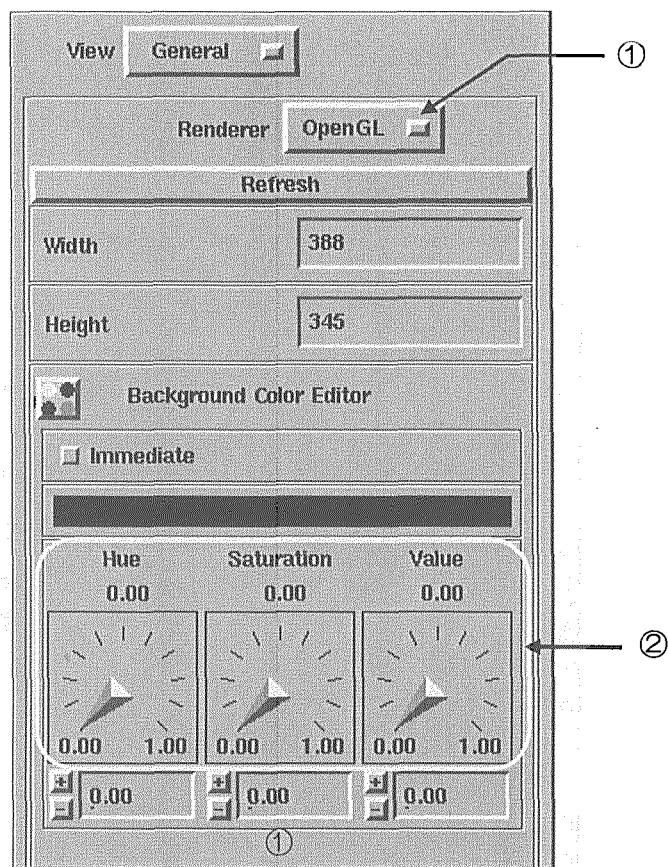


図 3-23 [General] ビューエディタ・メニュー

3次元形状のレンダリングは、「ハードウェアとグラフィックス・ライブラリに依存する方法」と「レンダリング計算をCPUに依存する方法」があります。前者をハードウェア・レンダリング、後者をソフトウェア・レンダリングといいます。

ハードウェア・レンダリングはプラットフォームに依存しており、グラフィックス・ライブラリとしてはOpenGL, XGLなどがあります。ソフトウェアレンダリングは全プラットフォームに共通な機能を提供しており、特別なグラフィックス機能をもたない機種でも3次元表示が行なえるようになっています。

レンダリング・メソッドは、Renderer選択メニュー(図3-23 ①)で指定します。

コントロールパネルのツールバーにある (HW/SW)ボタンをオン/オフすることでも指定できます。

3.9.2 背景色

表示ウインドウの背景色を変更します。

背景色の指定は、メニューの中の [Hue], [Saturation], [Value] のダイヤル(図3-23 ②)の値を調節して指定します。

3.9.3 表示ウインドウの追加

現在表示しているオブジェクトと全く同じものを表示するウインドウを1つ追加します。ビューエディタ・メニューから[Scene]を選択します(図3-24 ①)。メニューは図3-24のようになります。

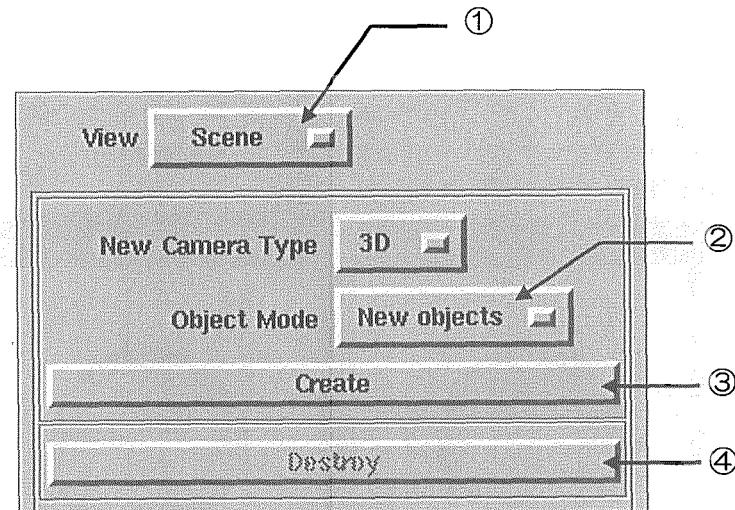


図3-24 [Scene]ビューエディタ・メニュー

[Object Mode]選択メニューから [Same objects] を選択します(図3-24 ②)。[Create]ボタン(図3-24 ③)を押すと、同じオブジェクトを表示したウインドウが新たに生成されます。ただし、カメラ属性やビュー属性はウインドウ毎に設定されるので、透視投影や背景色の変更などを行なっていてもそれらは新しく生成したウインドウには反映されません。

2つの表示ウインドウの大きさをマウスで適当に調節して、横にオブジェクトを回転します。(図3-25)
オブジェクトは2つの表示ウインドウで同じように回転します。これは、オブジェクトは1組だけで、それを写しているカメラが2台あり、同じものが2つのモニターに写し出されているのと同じことです。

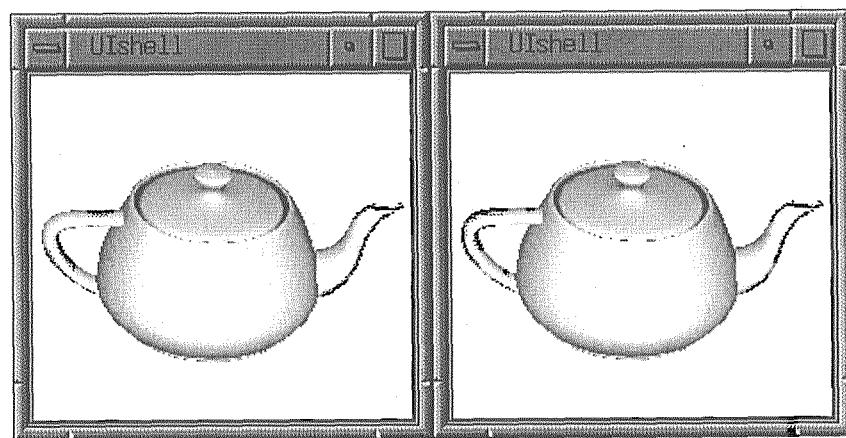


図3-25 オブジェクトを移動した場合の2つの表示ウインドウ

3.9.4 カメラの移動

カメラの位置を変更することにより、同じオブジェクトを異なる視点から見ることができます。

片方のカメラの位置を変更する方法について説明します。

まず、移動するカメラを選択します。2つ並べている表示ウインドウで、右側のウインドウの中でマウス左ボタンをクリックしてください。選択されると表示ウインドウの枠が赤色になります。オブジェクトを移動するモードからカメラを移動するモードに変更します。コントロールパネル

のツールバーから Transform Camera ボタン()を選択してください。黄色のボタンが選択されている状態を表します。

右側の表示ウインドウ内にマウスカーソルを移動し、回転操作を行うと、今度はオブジェクトではなく、カメラが回転します。(図 3-26)

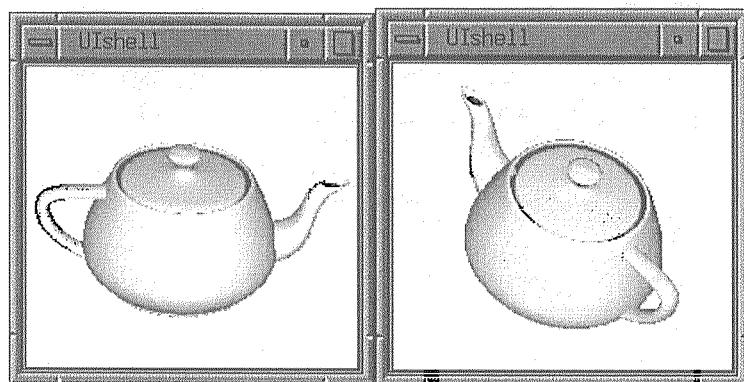


図 3-26 カメラを移動した場合の2つの表示ウインドウ

3.9.5 表示ウインドウの消去

新たに追加した表示ウインドウのみを消去することができます。AVS/Express を起動した時から表示されているウインドウはメニューから消去することはできません。

追加した表示ウインドウを消去します。

表示ウインドウのタイトルが "View#1" のウインドウを選択してください。次にビューエディタ・メニューから [Destroy] ボタン(図 3-24 ④)を押してください。"View#1" 表示ウインドウが消去されます。

3.10 プルダウンメニューのまとめ

幾何変換	[Transform]-[Transform]
表示方法	[Object]-[Modes]
オブジェクトの色	[Object]-[Properties]-[General] : [Hue], [Saturation], [Value]
ライン幅	[Object]-[Properties]-[Point/Line] : [Line Thickness]
オブジェクトの不透明度	[Object]-[Properties]-[Surface] : [Opacity]
自動正規化	[Camera]-[General]-[Auto Normalize]
透視投影	[Camera]-[Lens]
デプス・キューリング	[Camera]-[Depth Cueing]
レンダリングメソッド	[View]-[General]-[Renderer]
背景色	[View]-[General] : [Hue], [Saturation], [Value]
表示ウインドウの生成	[View]-[Scene] : [Create]
表示ウインドウの消去	[View]-[Scene] : [Destroy]

4. データファイルフォーマット

ここでは AVS/Express で取り込むことのできるファイル・フォーマットについて説明します。

AVS/Express データファイル・フォーマットには、大きく分けると 2 種類あります。

規則正しい格子上にデータが分布するタイプ(Field データ)と要素の集まりで構成される非構造格子型タイプ(UCD データ)です。Field データは、Field, Image, Volume の 3 種類のファイルフォーマットに分類されます。

(1) フィールドデータ (Field Data: 構造格子データ)

最も頻繁に使われるデータタイプで、流体解析などで用いられる差分格子のような規則正しく並んでいるデータに利用します。各格子点のデータ成分数、次元数等に特に制限はありません。フィールドデータは、“Read Field”モジュールを用いて AVS/Express にデータを取り込むことができます。ファイルの拡張子は、“.fld”です。

(2) イメージデータ (Image Data: 2 次元画像データ)

フィールドデータの一種で、使用頻度が高いため特別なフォーマットを持たせています。イメージデータの読み込みには、“Read Image”モジュールを用います。

ファイルの拡張子は、“.x”です。

(3) ボリュームデータ (Volume Data: 3 次元スカラーデータ)

フィールドデータの一種で、使用頻度が高いため特別なフォーマットを持たせています。ボリュームデータの読み込みには、“Read Volume”モジュールを用います。

ファイルの拡張子は、“.dat”です。

(4) UCD (Unstructured Cell Data: 非構造格子データ)

非構造格子型のデータタイプを表し、各種セルとノードから構成されます。このデータタイプは、有限要素法の解析結果のデータ等を表現するのに用いられます。UCD データは、“Read UCD”モジュールを用いて AVS/Express にデータを取り込むことができます。

ファイルの拡張子は、“.inp”です。

4.1 フィールドデータ

フィールドデータを格納するファイルです。データ情報を記述するヘッダー部、データ値を記述するデータ部、座標情報を記述する座標情報部から構成されます。ヘッダー部分には次元数、次元のサイズ、要素ごとのデータ数などが記述されています。データ部分には、ヘッダー部で定義されているデータ数だけデータを記述します。座標情報はフィールドのタイプが rectilinear, irregular の場合のみ記述します。

ファイル名には、拡張子”.fld”を付けます。

フィールドデータファイルには、構成によって二種類に分類されます。

1つは図 4-1(a)のようにヘッダー部、データ部、座標情報部を同じファイル内に直接記述するタイプ(ネイティブ・フィールドファイル)です。

もう1つは、図 4-1(b)のようにヘッダー部、データ部、座標情報部をそれぞれ別ファイルに記述し、ヘッダーファイルから他のファイルを参照するタイプ(アスキー記述ファイル)です。

詳細については、4.1.2と4.1.4で説明します。

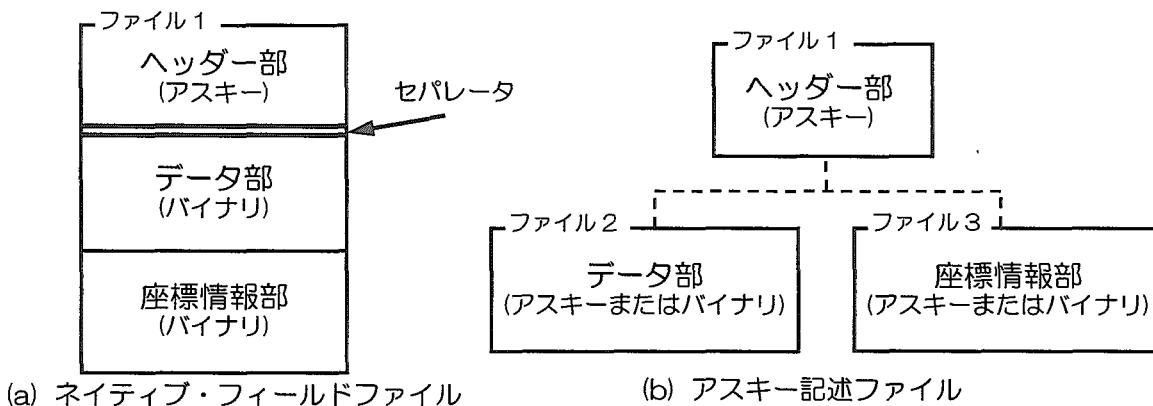


図 4-1 フィールドデータファイルの構成

2つのファイルタイプのヘッダ部は基本的には共通ですが、一部、アスキ記述ファイルにのみ含まれるものがあります。

4.1.1 ヘッダー部分の説明

ヘッダー部には図 4-2 に示すような、コメントや次元数、次元サイズなどの Field データを定義するのに必要な情報をアスキ形式で記述します。

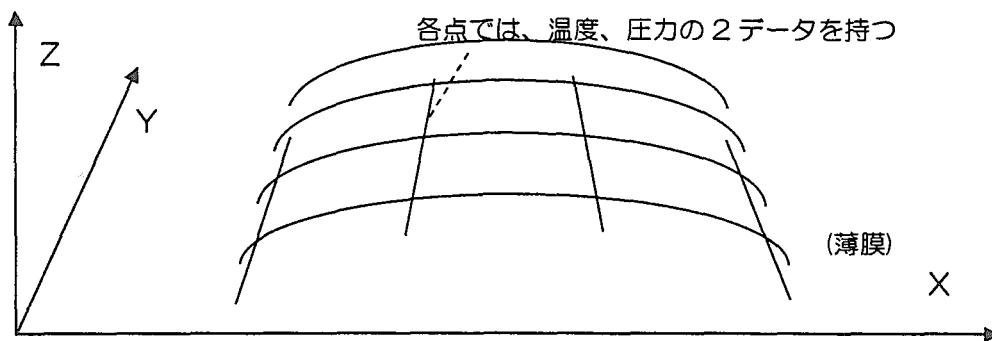
情報を記述するには、[キーワード=数字] というように幾つか用意されたキーワードに値をセットします。

データの区切りには空白かタブを使用します。大文字、小文字の区別はありませんが、先頭行の # AVS だけは大文字で記述します。

```
# AVS ----- (1)
#
# This is a sample for "field 3D 3-vector rectilinear" ----- (2)
#
ndim      = 3 ----- (3)
dim1      = 2 ----- (4)
dim2      = 3
dim3      = 4
nspace    = 3 ----- (5)
veclen    = 3 ----- (6)
data      = float ----- (7)
field     = rectilinear ----- (8)
```

図 4-2 ヘッダー部の記述

- (1) # AVS
Field データファイルの先頭には必ずこの文字列が必要です。（#の次には空白が入ります）
"# AVS"の後ろには任意の文字を書くことができます。
- (2) #
2 行目以降の#で始まる行はコメント行と見なされます。
- (3) ndim=<数値>
計算空間（データ配列）での次元数を定義します（図 4-3 参照）。
離散的にデータが分布し、隣り合うデータに規則性がない場合は 1 次元のデータ（ndim=1）として定義します。
規則性があり、2 次元格子の場合は 2 次元のデータ（ndim=2）となり、3 次元格子の場合は 3 次元のデータ（ndim=3）となります。
イメージデータの場合は 2、ボリュームデータの場合は 3 を記述します。



計算空間：薄膜の各格子点を表現するために必要な次元数（2 次元）
物理空間：薄膜を空間に表現するため必要となる座標次元数（3 次元）
veclen：各点で持つデータ数（2）

図 4-3 計算空間、物理空間と veclen の関係

- (4) dim1=<数値>
dim2=<数値>
dim3=<数値>
各軸方向の格子数を定義します。（dim1 → X 軸、dim2 → Y 軸、dim3 → Z 軸）
- (5) nspace=<数値>
データが存在する物理空間（実空間）の次元数を定義します（図 4-3 参照）。
- (6) veclen=<数値>
格子点に存在するデータ成分の種類の数を定義します。
温度と圧力の 2 つの物理量が存在する場合は、veclen=2 となります（図 4-3 参照）。
- (7) data=<文字列>
データ成分の型を定義します。
<文字列>の部分には、以下の 5 タイプから設定します。
short, byte, integer, float, double
注：データ成分の種類が複数ある場合、すべての成分は同じデータ型になります。

(8) field=<文字列>

座標値に関する情報を定義します。

座標値の持ち方によって<文字列>の部分には、以下の3タイプから設定します。

□uniform (図4-4)

直交等間隔（間隔は1とする）の格子で座標値を持たないタイプ。

□rectilinear (図4-5)

直交不等間隔の格子で、各軸方向の座標値を定義します。

(3次元の場合は、dim1+dim2+dim3個の座標データを持ちます)

□irregular (図4-6)

すべての格子点の座標値を定義します。

(3次元の場合はdim1×dim2×dim3個の座標データを持ちます)

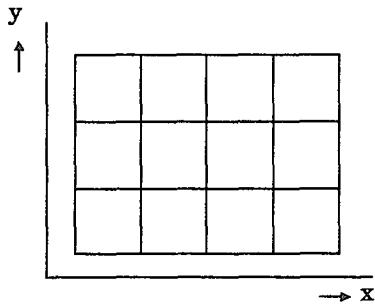


図4-4 uniform

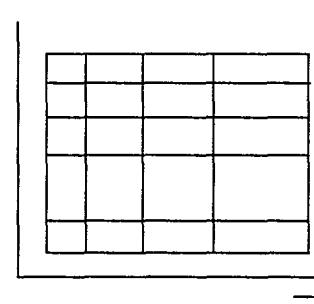


図4-5 rectilinear

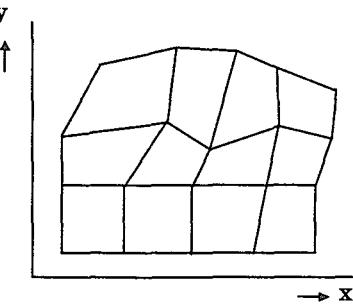


図4-6 irregular

(9) min_ext=<X座標の最小値><Y座標の最小値><Z座標の最小値> (オプション)

(10) max_ext=<X座標の最大値><Y座標の最大値><Z座標の最大値> (オプション)

座標軸の最小値、最大値を指定することにより格子の存在範囲を指定します。Y座標、Z座標を指定する場合はスペースまたはカンマを入れて区切ります。

(11) label=<文字列><文字列>... (オプション)

データ成分に対してラベルを付けます。

データ成分の種類が複数ある場合は<文字列>と<文字列>の間にスペースまたはカンマを入れて区切ります。

(12) unit=<文字列><文字列>... (オプション)

データの成分に対して単位を付けます。

データ成分が複数ある場合は<文字列>と<文字列>の間にスペースまたはカンマを入れて区切ります。

(13) min_val=<数値><数値>... (オプション)

(14) max_val=<数値><数値>... (オプション)

データ成分毎に最小値、最大値を指定します。

データ表示の際に大きさを正規化するときに使用されます。

(15) nstep=<数値> ... (オプション)

ステップ数を指定します。アスキー記述ファイルのときのみ有効です。

時系列（複数）データを定義する場合にのみ有効のため、このキーワードは AVS/Express3.2 以前のバージョンおよび AVS5 ではサポートされていません。

4.1.2 アスキー記述ファイル

ヘッダー部、データ部、座標情報部をそれぞれ別ファイルに記述するフォーマットです。ヘッダー部には、参照するデータ部、座標情報についてどのように利用するのかを記述します。参照するデータ部と座標情報には、アスキー形式とバイナリ形式のどちらでも記述できます。ヘッダー部のファイル名には、拡張子”.fld”を付けなくてはなりません。（図 4-7）

```
-- [ヘッダー部] -----
# AVS
#
#This is a sample for "field 3D 3-vector 3-space rectilinear"
#
ndim      = 3
dim1      = 2
dim2      = 3
dim3      = 4
nspace    = 3
veclen    = 3
data       = float
field     = rectilinear
label     = data1 data2 data3

variable 1 file=./sample6_data filetype=ascii skip=1 offset=0 stride=3  (1)
variable 2 file=./sample6_data filetype=ascii skip=1 offset=1 stride=3
variable 3 file=./sample6_data filetype=ascii skip=1 offset=2 stride=3
coord 1 file=./sample6_coord filetype=ascii skip=1  (2)
coord 2 file=./sample6_coord filetype=ascii skip=4
coord 3 file=./sample6_coord filetype=ascii skip=8

-----
```

[データ部]

sample6_data		
# data1 data2 data3		
1.0	1.0	4.0
1.0	2.0	4.0
1.0	3.0	4.0
:	:	:

[座標情報部]

x_coord	sample6_coord
0.0	
25.0	# y_coord
	0.0
	30.0
	50.0
z_coord	# z_coord
0.0	0.0
	10.0
	15.0
	40.0

図 4-7 アスキー記述ファイルの例

(1) データ部分

variable n file=<文字> filetype=<文字> skip=<数値> offset=<数値> stride=<数値>

□ variable

データに関する情報を示す行は、このキーワードで始まります。
“variable” 行は、“veclen” で記述した数で記述します。

□ n

データ成分を区別する番号で、1 から始まる整数値を指定します。

□ file=<文字>

読み込むファイル名を指定します。絶対パス名、相対パス名のどちらでも構いません。

□ filetype=<文字>

読み込むファイルのデータタイプを指定します。

filetype=ascii

: アスキ

filetype=binary

: バイナリ (C のプログラムで出力)

filetype=unformatted

: バイナリ (FORTRAN の UNFORMATTED で出力)

unformatted は FORTRAN の UNFORMATTED で出力したデータの最初の 2byte をスキップして読み込みます。

□ skip=<数値> ... (オプション)

最初にどれだけデータを読み飛ばすかを指定します。

指定しない場合のデフォルト値は skip=0 です。

ascii

: 最初に読み飛ばす行数

binary, unformatted

: 最初に読み飛ばすバイト数

□ offset=<数値> ... (オプション)

ascii の場合のみ有効です。最初のデータを読み込むときには行の先頭から何カラム読み飛ばすかを指定します。デフォルト値は offset=0 です。

□ stride=<数値> ... (オプション)

データとデータの間を何カラム毎読み込むのかを指定します。

ascii

: 読み飛ばす項目数。

データとデータの間にスペースまたはタブが 1 つ以上ある場合、それを 1 つの区切りと見なします。

binary, unformatted

: 読み飛ばす要素数。

[データサイズ × stride 値] バイトごとにデータを読み込むように指定します。

float 型データが並んでいる場合 stride = 3 とすると
[(float 型データサイズ) × 3] バイトごとにデータを読み込みます。

※ 各データ型(short, byte, float, double, integer)の
バイトサイズはマシン、OS 環境毎に異なります

(2) 座標情報

coord n file=<文字> filetype=<文字> skip=<数値> offset=<数値> stride=<数値>

 coord

座標値に関する情報を示す行は、このキーワードで始まります。

"coord"行は、"n space"で記述した数だけ記述します。

 n

座標軸を区別する番号です。

n=1 : X 座標

n=2 : Y 座標

n=3 : Z 座標

 file=<文字> filetype=<文字> skip=<数値> offset=<数値> stride=<数値>

データ部分のキーワードと同じ意味です。

4.1.3 複数ステップのアスキー記述ファイル

アスキー記述ファイルの場合、複数ステップのフィールドデータを記述することができます。複数ステップのデータを記述する場合は、必要なステップの分だけ variable と coord を続けて記述します。第2ステップ以降の座標値が変化しない場合は、coord を省略することができます。(図 4-8)

```
# AVS field file
#
nstep      = 3 ----- (1)
ndim       = 2
dim1       = 4
dim2       = 3
n space    = 3
veclen     = 1
data        = float
field       = irregular

#time step 1
time file=./sample.data filetype=ascii skip=1 close=0 ----- (2)
variable 1 ...
coord 1 ...
coord 2 ...
coord 3 ...
EOT ----- (3)

#time step 2
time ...
```

図 4-8 複数ステップのアスキー記述の例

```

# AVS field file
#
nstep      = 3          ----- (1)
ndim       = 2
dim1       = 4
dim2       = 3
nspatial   = 3
veclen     = 1
data        = float
field       = irregular

#time step 1
time file=./sample.data filetype=ascii skip=1 close=0 ----- (2)
variable 1 ...
coord 1 ...
coord 2 ...
coord 3 ...
EOT           ----- (3)

DO           ----- (4)
time ...
variable 1 ...
coord 1 ...
coord 2 ...
coord 3 ...
EOT           ----- (4)
ENDDO

```

図 4-9 DO ~ ENDDO を用いた複数ステップのアスキーメモリの例

(1) nstep=<数値> ... (オプション)

ステップ数を指定します。DO ~ ENDDO が記述されている場合は必須です。

DO ~ ENDDO の記述がない場合は nstep が記述されていても無視されます。

(2) time file=<文字> filetype=<文字> skip=<数値> offset=<数値> close=0/1(オプション)

[または time value=<文字>]

ステップに対するコメント(時刻、ステップ番号など)を記述します。

コメントが別のファイルに記述されている場合(time file=...)と直接記述する場合

(time value=...)の二通りがあります。前者の場合、filetype=ascii であれば任意の文字列を読み込むことができますが、filetype=binary または unformatted であれば float(real) に限られます。stride のキーワードはありません。

□ close=0 または 1 ... (オプション)

close=1(デフォルト、省略可)の場合:

そのステップを読み終わったあとにファイルがクローズされます。次のファイルを読み込むときには、skip, stride の記述をデータファイルの先頭から指定する必要があります。

close=0 の場合:

そのステップを読み終わってもファイルはクローズされず、ファイルポインタを保持します。そのため、次のステップが同じデータファイルに記述されている場合は、その位置から相対的に指定することができます。

(3) EOT

各ステップの終わりに必要なキーワードです。最後のステップにも必須です。
ステップ数が 1 の場合に限り省略できます。

(4) DO ~ ENDDO ... (オプション)

DO と ENDDO で囲まれたブロックが最後の nstep ステップまで繰り返されます。

ENDDO の後に別のステップを記述することはできません。また、このキーワードを使用する場合には nstep キーワードが必須になります。(図 4-9)

サンプル・ファイル

以下のサンプルデータを読み込むためのフィールド・ファイルの例を示します。

sample.data

#data	X	Y	Z
1			
5.0	-2.0	-1.0	0.0
10.0	-1.0	-1.0	0.0
20.0	0.0	-1.0	0.0
25.0	1.0	-1.0	0.0
30.0	-2.0	0.0	0.0
40.0	-1.0	0.0	0.0
50.0	0.0	0.0	0.0
60.0	2.0	0.0	0.0
2			
5.0	-2.0	-1.0	0.0
10.0	-1.0	-1.0	0.0
20.0	0.0	-1.0	0.0
25.0	1.0	-1.0	0.0
30.0	-2.0	0.0	0.0
40.0	-1.0	0.0	0.0
50.0	0.0	0.0	0.0
60.0	2.0	0.0	0.0
3			
5.0	-2.0	-1.0	0.0
10.0	-1.0	-1.0	0.0
20.0	0.0	-1.0	0.0
50.0	2.0	-1.0	0.0
35.0	-2.0	-1.0	0.0
45.0	-1.0	0.0	0.0
55.0	0.0	0.0	0.0
95.0	3.0	0.0	0.0

【例 1 : close=1 の場合】

```
# AVS
#
ndim      = 2
dim1      = 4
dim2      = 2
nspacce   = 3
veclen    = 1
data      = float
field     = irregular

time file=./sample.data filetype=ascii skip=1 close=1
variable 1 file=./sample.data filetype=ascii skip=2 offset=0 stride=4 close=1
coord 1 file=./sample.data filetype=ascii skip=2 offset=1 stride=4 close=1
coord 2 file=./sample.data filetype=ascii skip=2 offset=2 stride=4 close=1
coord 3 file=./sample.data filetype=ascii skip=2 offset=3 stride=4 close=1
EOT

time file=./sample.data filetype=ascii skip=10 close=1
variable 1 file=./sample.data filetype=ascii skip=11 offset=0 stride=4
close=1
coord 1 file=./sample.data filetype=ascii skip=11 offset=1 stride=4 close=1
coord 2 file=./sample.data filetype=ascii skip=11 offset=2 stride=4 close=1
coord 3 file=./sample.data filetype=ascii skip=11 offset=3 stride=4 close=1
EOT

time file=./sample.data filetype=ascii skip=19 close=1
variable 1 file=./sample.data filetype=ascii skip=20 offset=0 stride=4
close=1
coord 1 file=./sample.data filetype=ascii skip=20 offset=1 stride=4 close=1
coord 2 file=./sample.data filetype=ascii skip=20 offset=2 stride=4 close=1
coord 3 file=./sample.data filetype=ascii skip=20 offset=3 stride=4 close=1
EOT
```

第1ステップ

第2ステップ

第3ステップ

【例 2 : close=0、DO ~ ENDDO の場合】

```
# AVS
#
nstep     = 3
ndim      = 2
dim1      = 4
dim2      = 2
nspacce   = 3
veclen    = 1
data      = float
field     = irregular

time file=./sample.data filetype=ascii skip=1 close=0
variable 1 file=./sample.data filetype=ascii skip=2 offset=0 stride=4 close=0
coord 1 file=./sample.data filetype=ascii skip=2 offset=1 stride=4 close=0
coord 2 file=./sample.data filetype=ascii skip=2 offset=2 stride=4 close=0
coord 3 file=./sample.data filetype=ascii skip=2 offset=3 stride=4 close=0
EOT

DO
time file=./sample.data filetype=ascii skip=8 close=0
variable 1 file=./sample.data filetype=ascii skip=1 offset=0 stride=4 close=0
coord 1 file=./sample.data filetype=ascii skip=1 offset=1 stride=4 close=0
coord 2 file=./sample.data filetype=ascii skip=1 offset=2 stride=4 close=0
coord 3 file=./sample.data filetype=ascii skip=1 offset=3 stride=4 close=0
EOT
ENDDO
```

第1ステップ

第2ステップ

第3ステップ

4.1.4 ネイティブ・フィールドファイル

ヘッダー部、データ部、座標軸情報部を1つのファイルの中に記述するフォーマットです。

注：ヘッダー部分とデータ部分を区切るためにセパレータ”^L^L”([Ctrl]+L [Ctrl]+L)が必要です。(図 4-10)

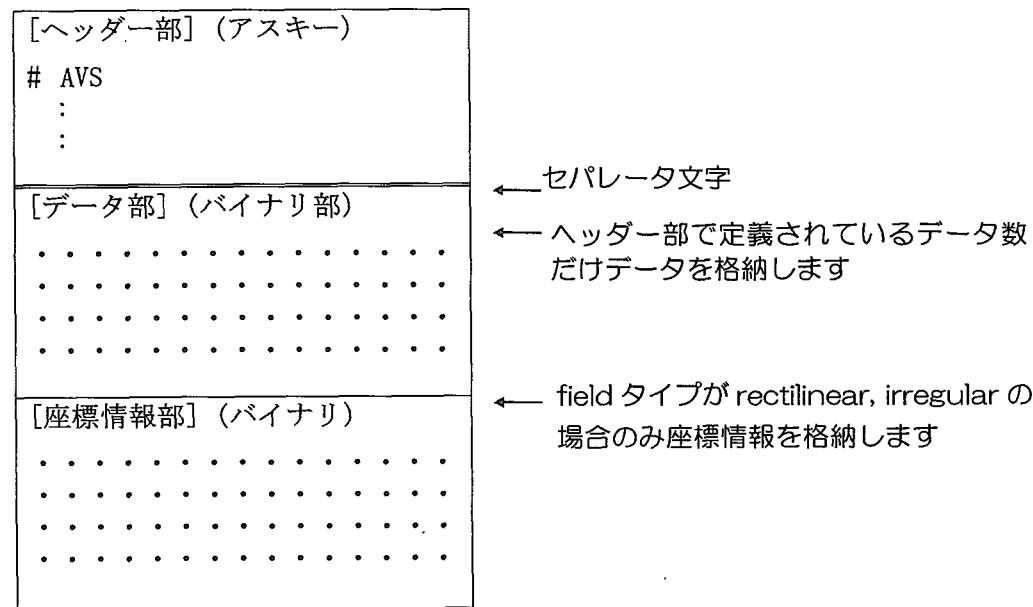


図 4-10 ネイティブ・フィールドファイル

(1) データ部

データ部はバイナリ形式で記述します。配列表現は以下の通りになります。

1) FORTRANでの配列表現

```
DIMENSION DATA(VECLEN, DIM1, DIM2, DIM3)
```

2) Cでの配列表現

```
float data[dim3][dim2][dim1][veclen];
```

図4-11にデータの格納順序を記述します。データ数は $[VECLEN \times DIM1 \times DIM2 \times DIM3]$ だけ記述します。

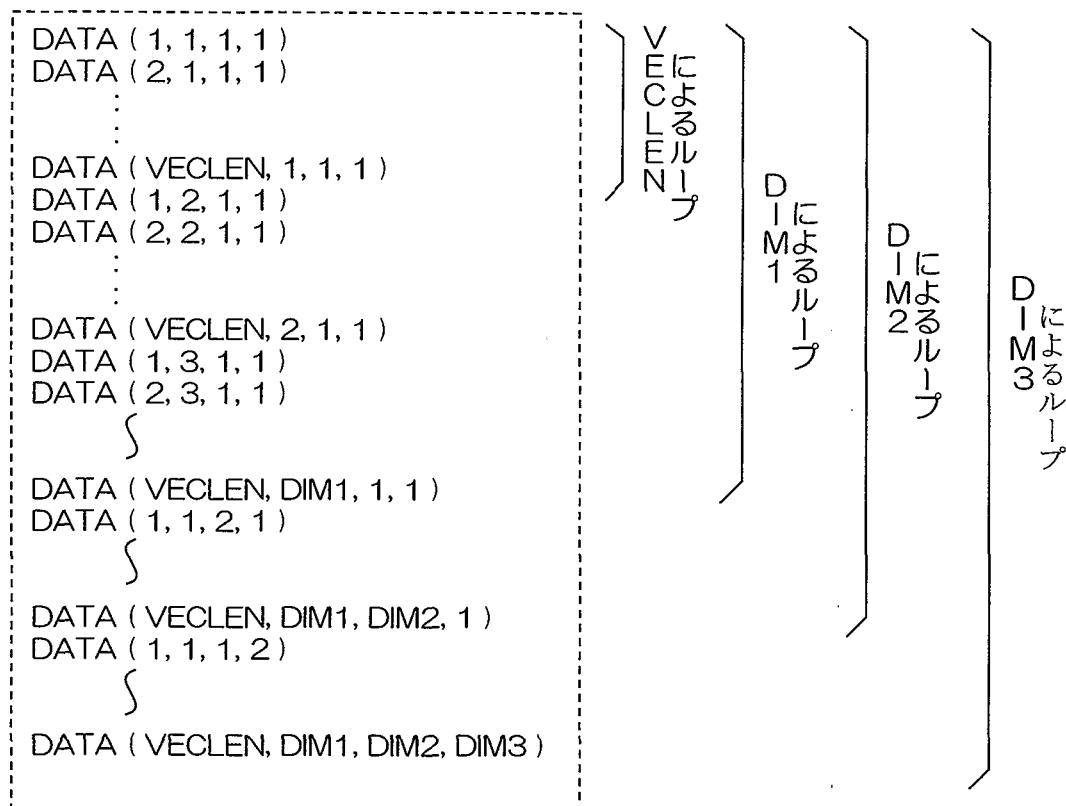


図4-11 データの格納順番

(2) 座標情報部

このフォーマットで座標データを読み込む場合は、座標値のデータの並びに注意しなければなりません。

図 4-12 の(a)にあるように、最初にX座標データを全て記述し、続いてY座標データ、Z座標データという順番に記述します。図 4-12 の(b)のような記述は誤りです。

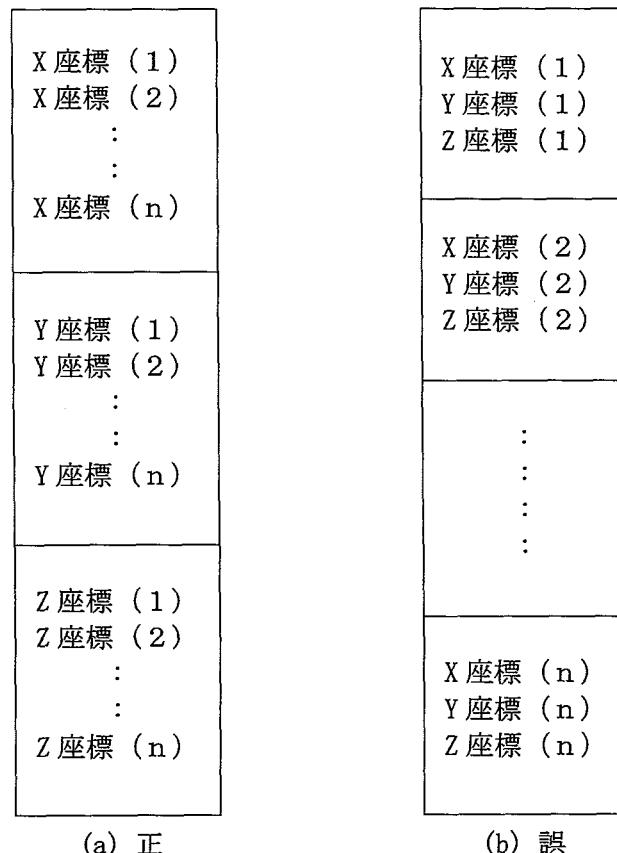


図 4-12 座標情報

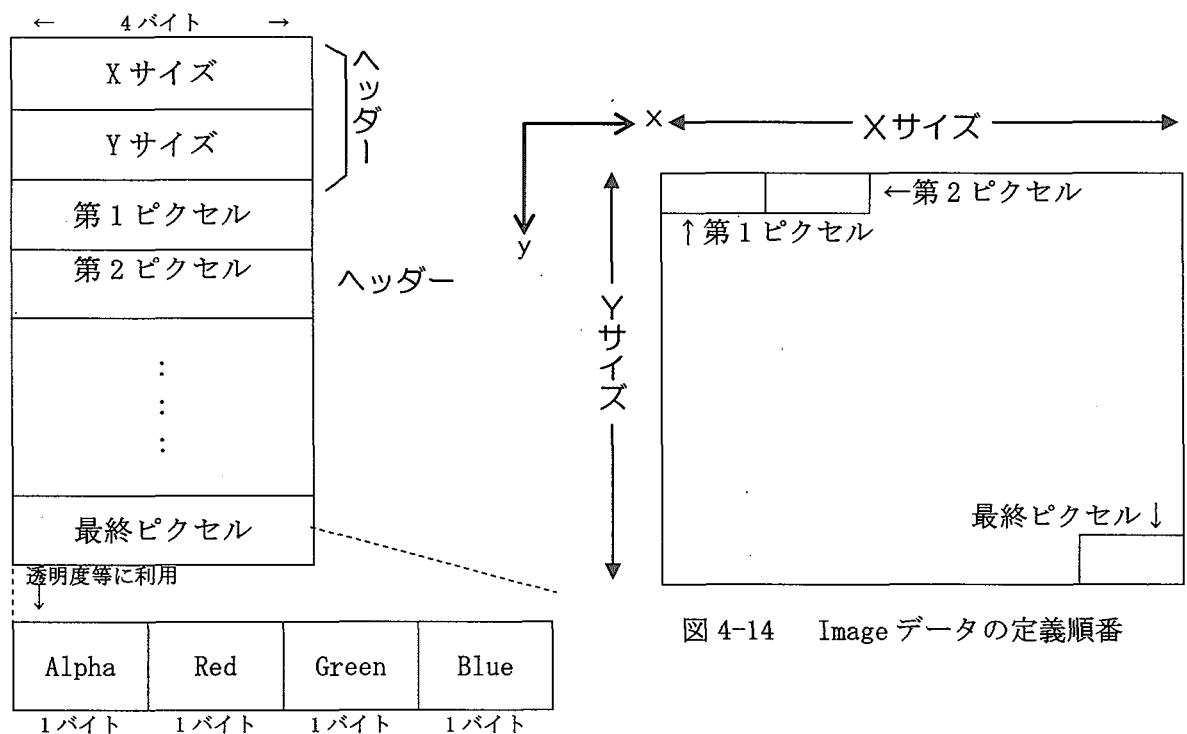
4.2 イメージデータ

イメージデータは、衛星写真やX線写真等の任意の大きさの2次元画像データを表現する際に使用します。ファイル名には、拡張子".x"をつけて作成します。

4.2.1 ファイルフォーマット

イメージデータのファイルは、バイナリで書かれています。

イメージデータのファイルフォーマットは、ファイルの先頭にイメージのサイズを指定した2ワードのヘッダーを記述し、続いて各ピクセルのデータを〔X方向のサイズ×Y方向のサイズ〕分だけ記述します。各ピクセルのデータは4バイトにパックし、赤、緑、青の各成分を1バイト(0~255)づつ記述し、残りの1バイトを透明度等のデータに使用します。(図4-13、図4-14)



例えば、512x512ピクセルのImageデータの場合、 $512 \times 512 \times 4 + 4 \times 2 =$ 約 1Mbyte のファイルの大きさになります。

4.3 ボリュームデータ

任意の大きさのボクセルデータ（0から255に正規化された3次元のスカラーデータ）を表現する際に用います。数値シミュレーション、MRI（Magnetic Resonance Imaging：核磁気共鳴画像化装置）、流体解析等のデータに利用されます。ファイル名には、拡張子".dat"をつけて作成します。

4.3.1 ファイルフォーマット

ボリュームデータのファイルは、バイナリで書かれています。ボリュームデータのファイルフォーマットは、ファイルの先頭にボリュームのサイズを指定した3ワードのヘッダーを記述し、続いて各ボクセルのデータを〔X方向のサイズ×Y方向のサイズ×Z方向のサイズ〕分だけ記述します。（図4-15、図4-16）

データ部分にはバイト型のデータを、FORTRANの3次元配列の書式無し出力の順序で記述します。配列のサイズは各1バイトで定義されているため、ボリュームデータの最大サイズは $255 \times 255 \times 255$ です。

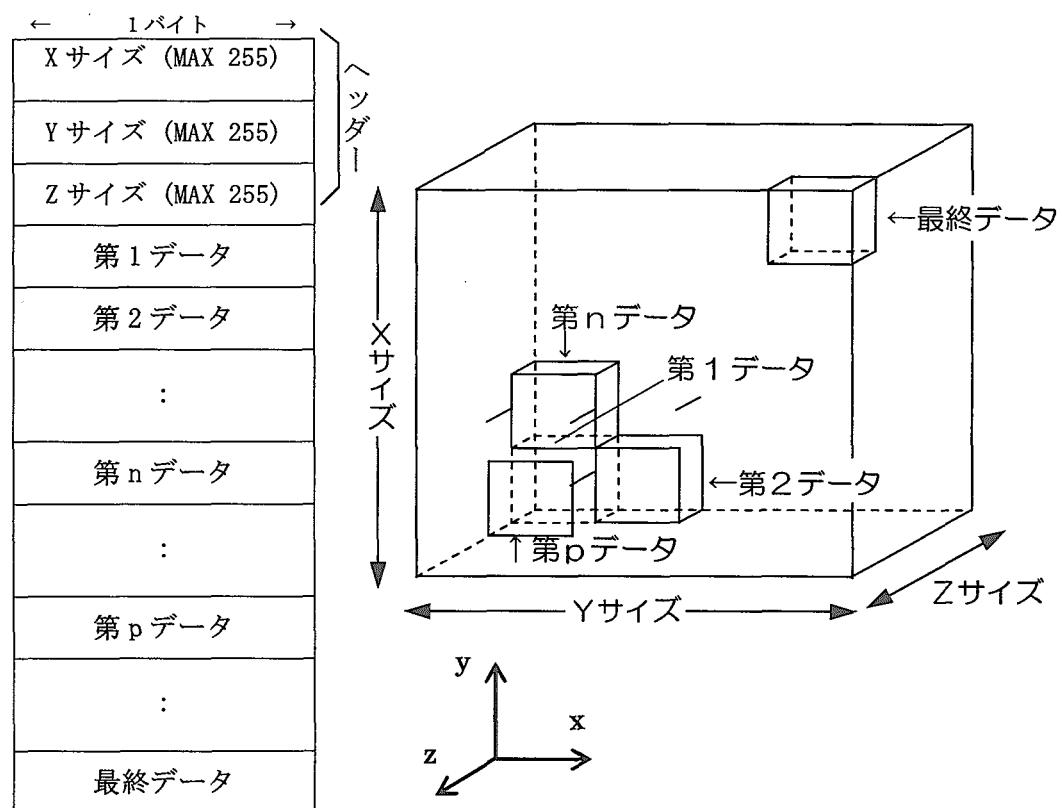


図4-15 Volumeデータのフォーマット

図4-16 ボクセルデータの定義順番

4.4 UCD データ

非構造格子型を表すデータタイプで、主に構造解析や計算流体力学の有限要素法の解析結果を表現する際に用います。ファイルは、拡張子".inp"をつけて作成します。

4.4.1 データ構造

UCD データは、図 4-17 のようにストラクチャー(Structure:構造体)をトップにセル(Cell:要素)、ノード(Node:節点)という階層構造になっています。データは、ストラクチャー全体、各セル、各ノードを持たせることができ、それぞれの成分はスカラーまたはベクトルで表します。UCD でデータ表現する場合には、各節点の座標情報の他に各セルのタイプやどの節点から構成されているかなどの情報も必要になります。

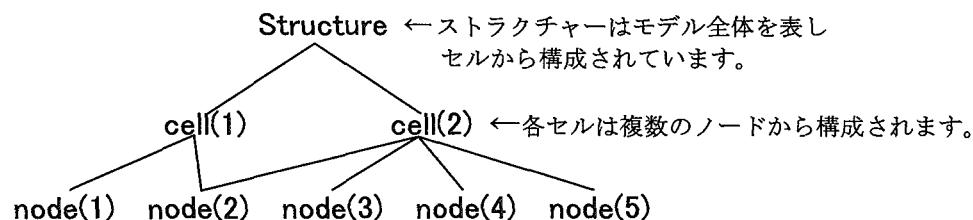


図 4-17 UCD データ構造

ストラクチャを構成するセルには、図 4-18 のように 8 タイプがあります。

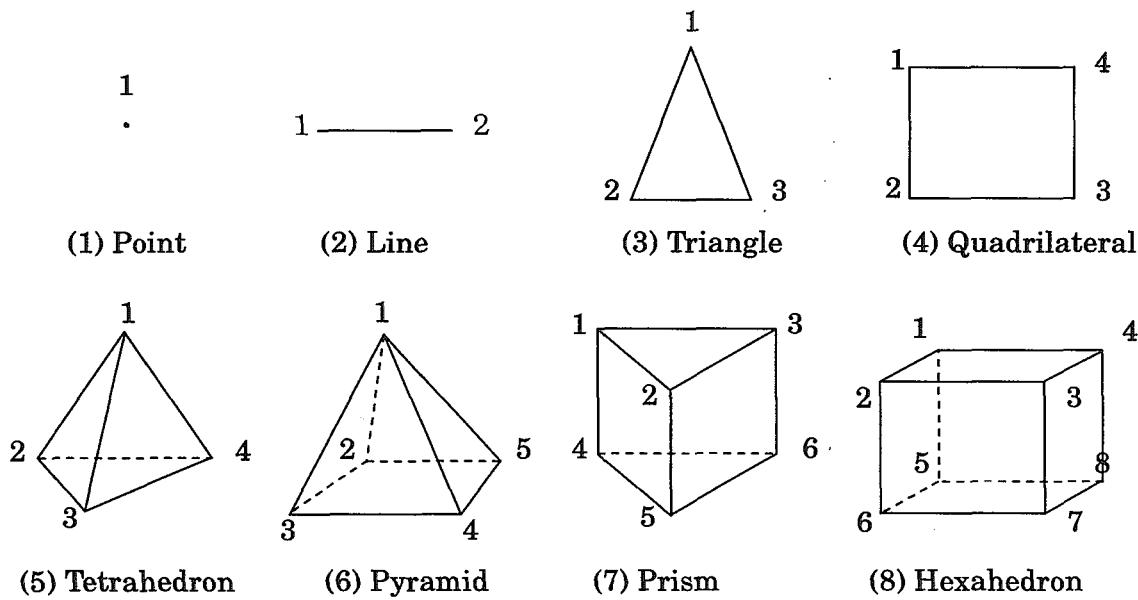


図 4-18 セルタイプ

各セルのノード番号は、データファイルの中でのノードのつながりを定義するときの順番です。番号をつけるときは法線の向きを考慮する必要があります。(図 4-19)

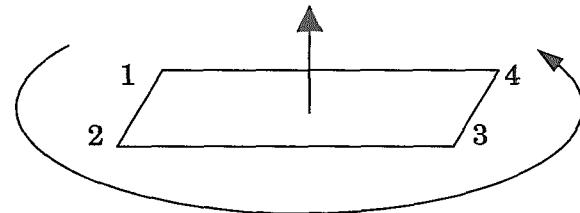


図 4-19 ノード番号の付け方

AVS/Express3.3からは図4-20のように各セルに対して中間節点のデータを持たせることができます。これにより、非構造格子データの2次元要素データを設定できます。

※ [] 内の数字が中間節点のノード番号です。

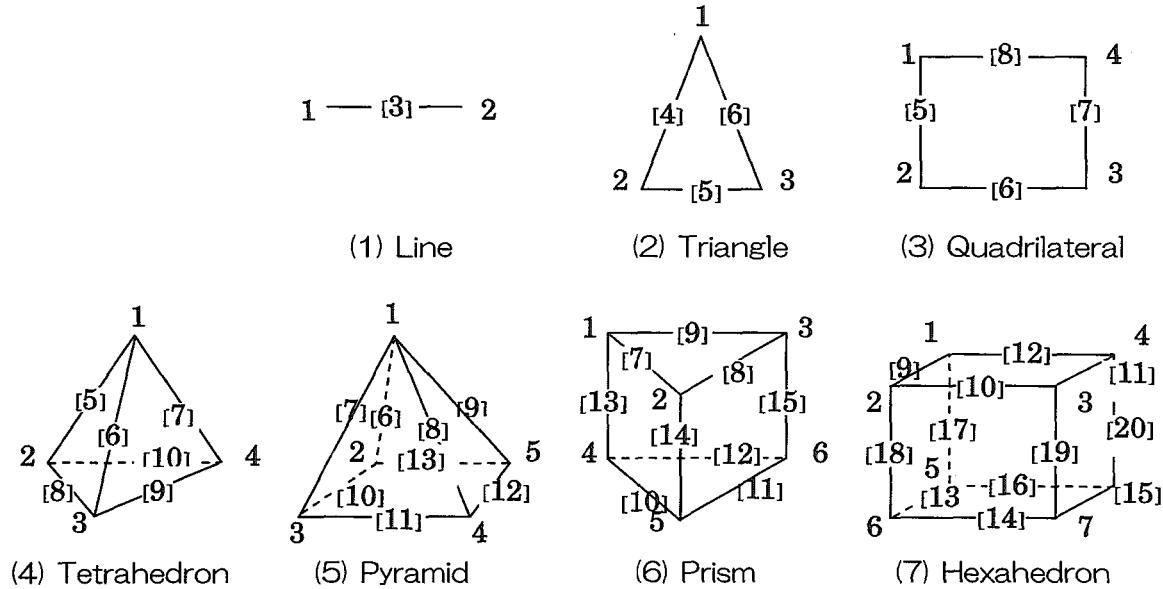


図 4-20 セルの中間節点

4.4.2 ファイルフォーマット

UCD ファイルフォーマットは図 4-21 のようにノードの座標値、つながりなどをアスキイで記述します。UCD ファイルフォーマットは、大きく分けると 6 つの部分から構成されています。(図 4-21 参照)

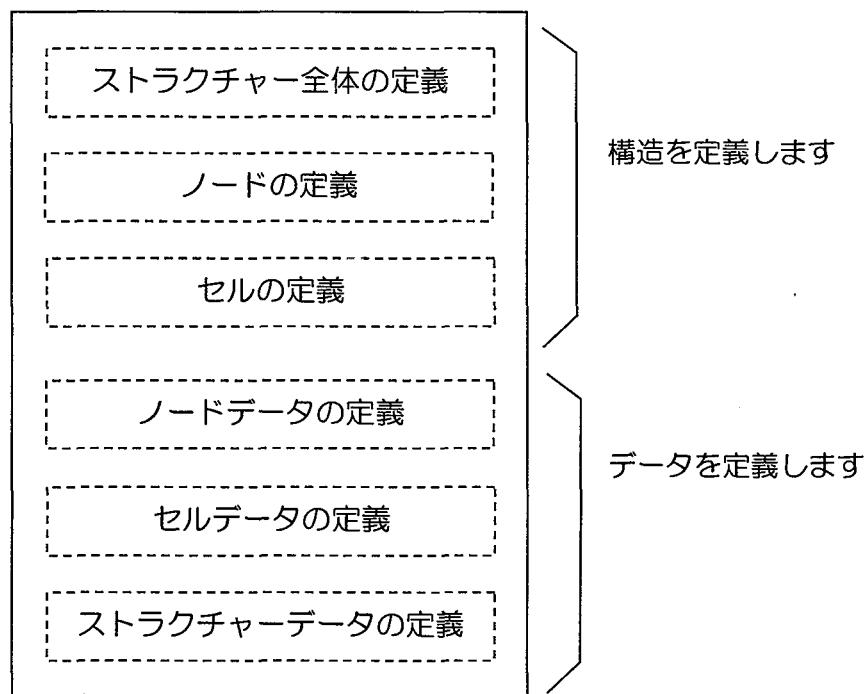


図 4-21 UCD データファイルの構成図

最初の 3 つの部分（構造の定義）については必ず記述します。データの定義部分については必要に応じて記述します。それぞれの定義方法については、図 4-22 を参照してください。

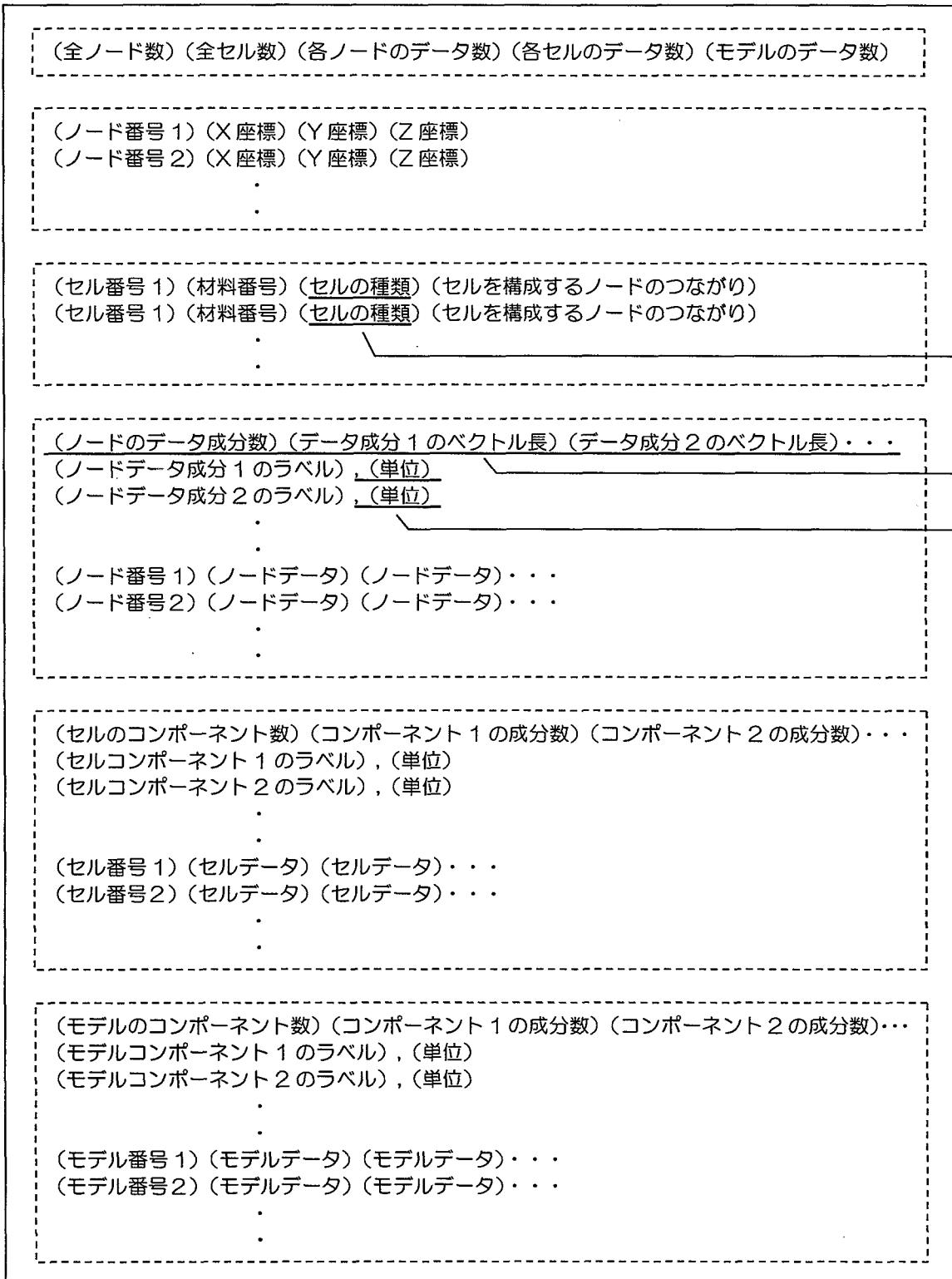


図 4-22 アスキーファイルフォーマット

注1：(セルの種類)には、次のように省略して記述します。

セル形状	1次要素	2次要素
点(Point)	pt	—
線分(Line)	line	line2
三角形(Triangle)	tri	tri2
四角形(Quadrilateral)	quad	quad2
四面体(Tetrahedron)	tet	tet2
角錐(Pyramid)	pyr	pyr2
角柱(Prism)	prism	prism2
六面体(Hexahedron)	hex	hex2

注2：各ノード(セル)が持つデータ成分数とベクトル長の関係についてサンプル例は次の通りです。

	データ成分数	ベクトル長
流速: Vx, Vy, Vz (ベクトルデータ)	1	3
圧力: P (スカラーデータ)	1	1
温度: T (スカラーデータ)	1	1

注3：(単位)はオプションですが、単位がない場合でも(ラベル)の後にカンマ“, ”を入れてください。

4.4.3 複数ステップの UCD ファイルフォーマット

UCD データファイルは、アスキートバイナリの 2 種類のフォーマットがあります。

アスキーフォーマットの場合は、ノード/セルの定義(形状データ)、ノード/セルデータの定義を繰り返し記述することで、複数ステップの UCD ファイルフォーマットを定義することができます。(図 4-23)

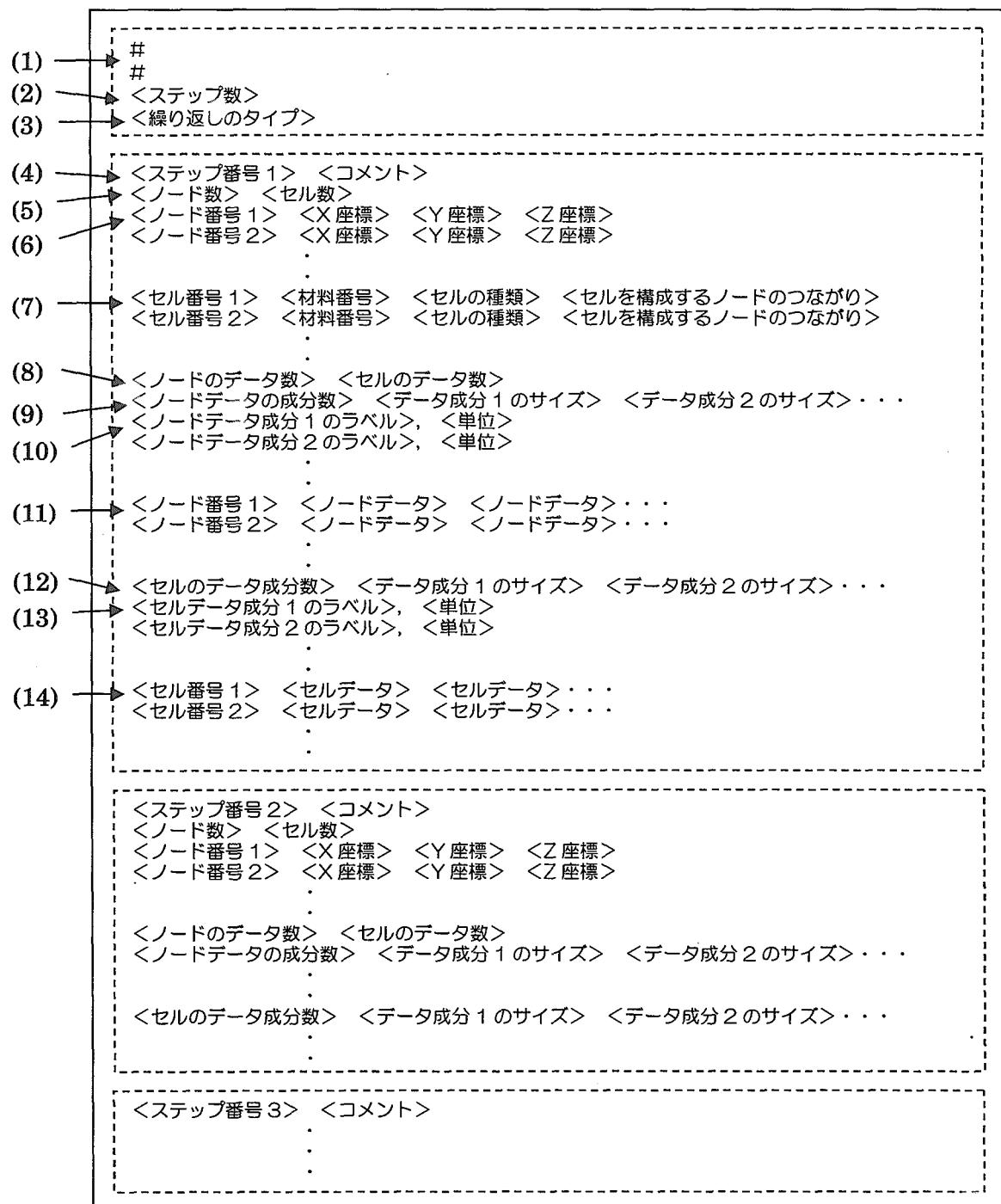


図 4-23 複数ステップのアスキーファイルフォーマット

(1)

コメント行です。コメント行はデータに関する記述の間に入れることはできません。

(2) ステップ数

ステップ数を記述します。この行が記述されていない場合は、従来のフォーマットとして判断されます。

(3) 繰り返しのタイプ

以下の 3 タイプから指定します。

 data

形状データが最初のステップのみに存在し、ノード／セルデータが各ステップに存在するタイプです。

```
step1
<形状データ>
<ノード／セルデータ>
step2
<ノード／セルデータ>
```

 geom

形状データが各ステップに存在し、ノード／セルデータが最初のステップのみに存在するタイプです。

```
step1
<形状データ>
<ノード／セルデータ>
step2
<形状データ>
```

 data_geom

形状データとノード／セルデータが各ステップに存在するタイプです。

```
step1
<形状データ>
<ノード／セルデータ>
step2
<形状データ>
<ノード／セルデータ>
step3
```

(4) <ステップ数> <コメント>

ステップ番号は各ステップの先頭に stepN(Nは整数で 1 から) というように記述します。

ステップ番号の後ろにはコメント(オプション)を記述することができます。

(5) <ノード数> <セル数>

ノードの数とセルの数を記述します。

繰り返しのタイプが geom または data_geom の場合はステップ毎に記述されますが、data の場合は第 1 ステップと同じ内容を第 2 ステップ以降に記述する必要があります。

従来の UCD フォーマットでは、この項目は先頭行にありました。

(6) <ノード番号N> <X座標> <Y座標> <Z座標>

全てのノードに関して記述します。ノード番号は整数値です。重複していなければ連続していなくても構いません。

(7) <セル番号N> <材料番号> <セルの種類> <セルを獲得するノードのつながり>

□ セル番号

全てのセルに関して記述します。セル番号は整数値です。重複していなければ連続していなくても構いません。

□ 材料番号

セルをグループ分けするときに利用する整数値です。

□ セルの種類

セルの種類を指定します。セルには、頂点のみにノードを持つ1次要素と中間接点を持つ2次要素とがあります。セルの種類については、図4-18、図4-20を参照ください。

□ セルを構成するノードのつながり

セルを構成するノード番号を記述します。番号付けを行なう順番は図4-18のセルタイプに付けられた番号で行ないます。

(8) <ノードのデータ数> <セルのデータ数>

各ノードのデータ数と各セルのデータ数を記述します。

ノードのデータ数=0の場合、(9)(10)(11)のノードデータに関する記述は行ないません。

セルのデータ数=0の場合、(12)(13)(14)のノードデータに関する記述は行ないません。

従来のUCDフォーマットでは、これらの項目は先頭行にありました。

(9) <ノードのデータ成分数> <データ成分1のサイズ> <データ成分2のサイズ> . . .

ノードのデータの成分数と各成分のサイズを記述します。

(10) <ノードのデータ成分Nのラベル> , <単位>

ノードのデータ成分のラベルと単位を記述します。ラベルと単位は省略することが可能ですが、カンマ " , " を省略することはできません。

(11) <ノード番号N> <ノードデータ>

全てのノードデータを記述します。ノードデータを記述する順番は(6)に依存します。

(12) <セルのデータ成分数> <データ成分1のサイズ> <データ成分2のサイズ> . . .

セルのデータの成分数と各成分のサイズを記述します。

(13) <セルのデータ成分Nのラベル> , <単位>

ノードのデータ成分のラベルと単位を記述します。ラベルと単位は省略することが可能ですが、カンマ " , " を省略することはできません。

(14) <セル番号N> <セルデータ>

全てのセルデータを記述します。セルデータを記述する順番は(7)に依存します。

図 4-24 に複数ステップの UCD ファイルフォーマットの例を示します。

```

# AVS new UCD format
2
data
step1 no. 1
14 6
    1   3.000000   0.000000   0.000000
    2   2.897780   0.776457   0.000000
    3   2.598080   1.500000   0.000000
    4   2.211320   2.121320   0.000000
    5   1.500000   2.598080   0.000000
    6   0.776457   2.897780   0.000000
    7   0.000000   3.000000   0.000000
    8   3.600000   0.000000   0.000000
    9   3.477330   0.931749   0.000000
   10   3.117690   1.800000   0.000000
   11   2.545580   2.545580   0.000000
   12   1.800000   3.117690   0.000000
   13   0.931749   3.477330   0.000000
   14   0.000000   3.600000   0.000000
    1       1     quad      1       8       9       2
    2       1     quad      2       9       10      3
    3       1     quad      3       10      11      4
    4       1     quad      4       11      12      5
    5       1     quad      5       12      13      6
    6       1     quad      6       13      14      7
4 0
2 3 1
disp, mm
temp, degree
    1   0.433090   0.000000   0.000000   1.0
    2   0.417480   -0.040804   0.000000   1.0
    3   0.372751   -0.080154   0.000000   1.0
    4   0.303528   -0.114977   0.000000   1.0
    5   0.214628   -0.141834   0.000000   1.0
    6   0.111246   -0.158527   0.000000   1.0
    7   0.000000   -0.164147   0.000000   1.0
    8   0.444816   0.000000   0.000000   1.0
    9   0.421018   -0.021786   0.000000   1.5
   10   0.357638   -0.054806   0.000000   1.5
   12   0.180339   -0.146491   0.000000   1.5
   13   0.089219   -0.181521   0.000000   1.0
   14   0.000000   -0.194428   0.000000   1.0

step2
4 0
2 3 1
disp, mm
temp, degree
    1   0.879362   0.000000   0.000000   1.0
    2   0.847907   -0.074771   0.000000   1.5
    3   0.758743   -0.147751   0.000000   1.5
    4   0.620941   -0.211376   0.000000   1.5
    5   0.441610   -0.211376   0.000000   1.5
    6   0.229843   -0.286550   0.000000   1.0
    7   0.000000   -0.295792   0.000000   1.0
    8   0.901270   0.000000   0.000000   1.0
    9   0.853196   -0.038875   0.000000   1.5
   10   0.727523   -0.102175   0.000000   2.0
   11   0.560769   -0.187785   0.000000   2.0
   12   0.348478   -0.272793   0.000000   2.0
   13   0.190488   -0.333520   0.000000   1.5
   14   0.000000   -0.355280   0.000000   1.0

```

図 4-24 複数ステップの UCD ファイルフォーマットの例

5. モジュール開発

5.1 V 言語

5.1.1 V 言語とは

V 言語とは、AVS/Express のカーネルであるオブジェクト・マネージャ(OM)をコントロールすることができる中間言語です。オブジェクトやネットワーク・エディタ、ライブラリなども V 言語によって定義されています。これらの定義ファイルは、以下のディレクトリにあります。

/appli/Express/express/v/

AVS/Express は起動時にこれらのファイルを読み込み、AVS/Express 自身を構築します。

V 言語は、以下の 3 つの要素で構成されます。

- V ステートメント
ネットワーク・エディタ上の基本操作を行うための V コードです。
オブジェクトの作成や変更、接続／切断／消去などを行います。
- V コマンド
アプリケーションに対して実行するコマンドです。
オブジェクトや V ファイル、アプリケーションなどの情報入手、実行制御などを行います。
- 組み込み関数
数値演算や論理演算などを行います。

作成した可視化ネットワーク(アプリケーション)の保存を行う場合も V 言語で記述された V ファイルとして保存されます。

5.1.2 オブジェクトの階層

○ 階層の構造

AVS/Express システムは図 5-1 のような階層構造になっています。

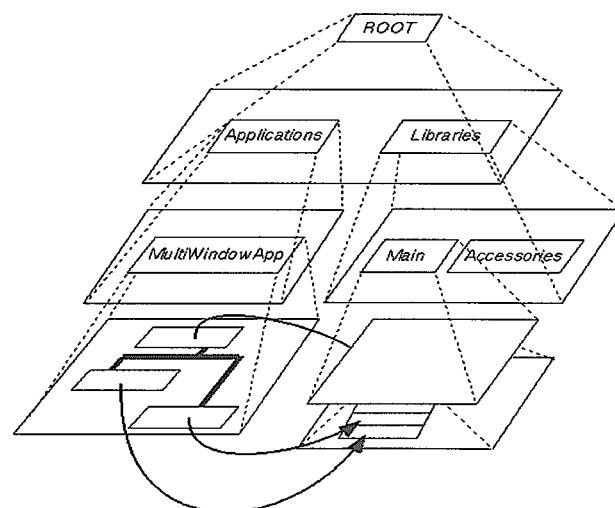


図 5-1 AVS/Express 階層構造

ネットワーク・エディタやライブラリに登録されているモジュールなど、全ては AVS/Express の Root 配下の階層に存在します。

AVS/Express を起動したときにデフォルトで現れる MultiWindowApp は、Root の下にある Applications 配下にあります。MultiWindowApp の存在する階層(パス)を AVS/Express では以下のように表現します。

Root.Applications.MultiWindowApp

○ VCP(V Comand Processor)

AVS/Express の GUI を利用して階層をたどることも可能ですが、ここでは VCP(V Comand Processor)を利用して階層を移動します。

VCP とは、アプリケーションの構築やデバッグなどのために実行する V コマンドを受け付けるための AVS/Express インターフェースです。ネットワーク・エディタとは違い、GUI を持たないテキストベースのインターフェースです。

AVS/Express を起動したウインドウが、VCP ウインドウです。(図 5-2)

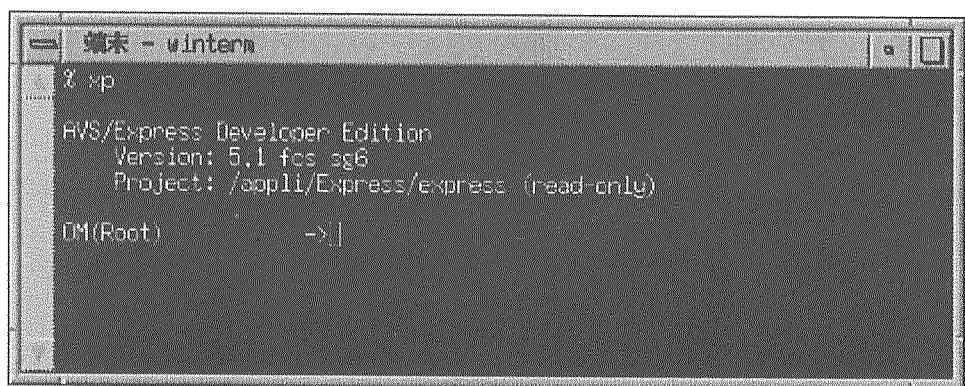


図 5-2 VCP ウィンドウ

VCP ウィンドウには、現在のアプリケーションの最上位階層である Root オブジェクトが表示されています。

OM(Root) ->

"->" の後に V コマンドを入力します。V コマンドの実行はカレントの階層に対して行われるため、V コマンドを実行する前に適切な階層に移動することが必要です。

○ 階層の移動

実際に VCP を利用して MultiWindowApp 階層まで移動する方法を示します。

\$list コマンドを用いてカレント・オブジェクト配下にあるオブジェクトのリストを出力します。

```
OM(Root)          -> $list
scheduler Scheduler
group ProcInstances
group+global Applications
list V
"$XP_PATH<0>/v/proc.v" ProcTemplates
"$XP_PATH<0>/v/templ.v" Templates
link UIdata
library Libraries
NEnetworkEditor NetworkEditor
```

Applications オブジェクト配下に移動します。

移動先のオブジェクトを指定し、"{"を付けて改行します。

```
OM(Root)          -> Applications {
OM(Applications) ->
```

()内に現在の階層が表示されます。

ここで \$list コマンドを実行すると Applications オブジェクト配下に MultiWindowApp があることを確認できます。

同様に、MultiWindowApp に移動してオブジェクト・リストを表示してください。以下のような二つのオブジェクトがあることを確認できます。

```
OM(Applications) -> MultiWindowApp {
OM(MultiWindowApp) -> $list
AU.MultiWindowApp.UI UI
GDM.Uviewer Uviewer
OM(MultiWindowApp) ->
```

Uviewer モジュールは、ネットワーク・エディタに表示されています。

UI モジュールは AVS/Express の GUI を構成しているモジュールで、AVS/Express 起動時には表示されません。

上の階層に戻る場合は、"};" と入力します。

```
OM(MultiWindowApp) -> };
OM(Applications) ->
```

5.1.3 オブジェクトのベースタイプ

AVS/Express にはモジュールを構成するオブジェクトが提供されています。
ここでは、ベースとなる 5 種類のオブジェクトを記述します。

- プリミティブ・オブジェクト (図 5-3)
- グループ・オブジェクト (図 5-4)
- モジュール・オブジェクト (図 5-5)
- マクロ・オブジェクト (図 5-6)
- リンク・オブジェクト (図 5-7)

int オブジェクトをインスタンスします。

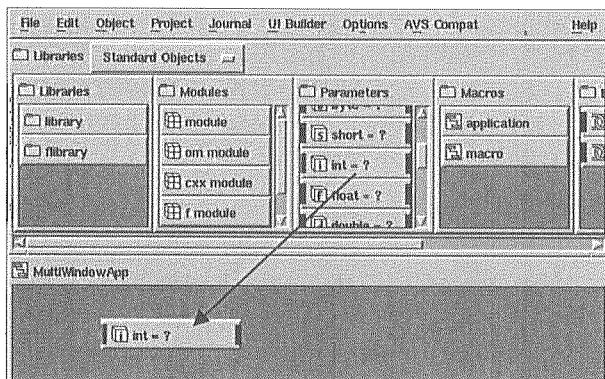


図 5-3 プリミティブ・オブジェクト

OM(MultiWindowApp) → *int A*;
 OM(MultiWindowApp) → *int B = 5*;
 OM(MultiWindowApp) → *A => B*;
 OM(MultiWindowApp) → *A =>*
 OM(MultiWindowApp) → *-A*;

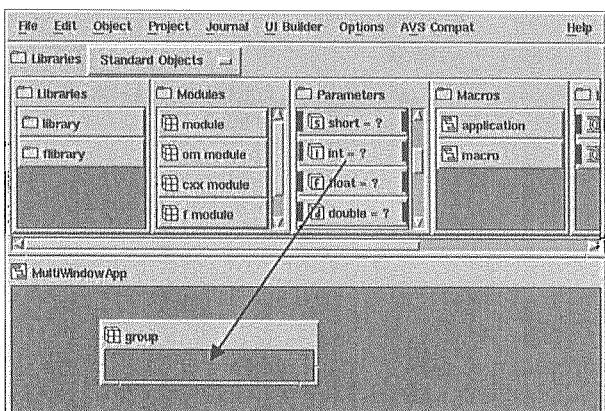
= : 代入

=> : 参照

- : モジュールの削除

○ グループ・オブジェクト

group オブジェクトをインスタンスし、int オブジェクトをその中にインスタンスします。
group オブジェクトがピンク色になつたらマウスボタンを離します。



OM(MultiWindowApp) → group grp1 {
 OM(grp1) → *int C*;
 OM(grp1) → *int D*;
 OM(grp1) → *D => <- B*
 OM(grp1) → *}*;

<- : 一つ上の階層

. : 結合

図 5-4 グループ・オブジェクト

○ モジュール・オブジェクト

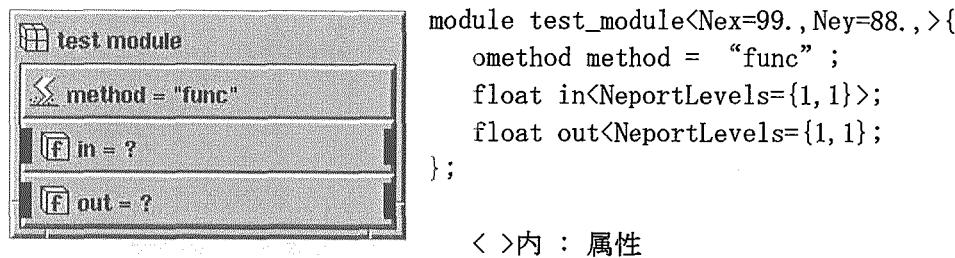


図 5-5 モジュール・オブジェクト

○ マクロ・オブジェクト

macro オブジェクトをインスタンスし、float オブジェクトをその中にインスタンスします。
macro オブジェクトがピンク色になつたらマウスボタンを離します。

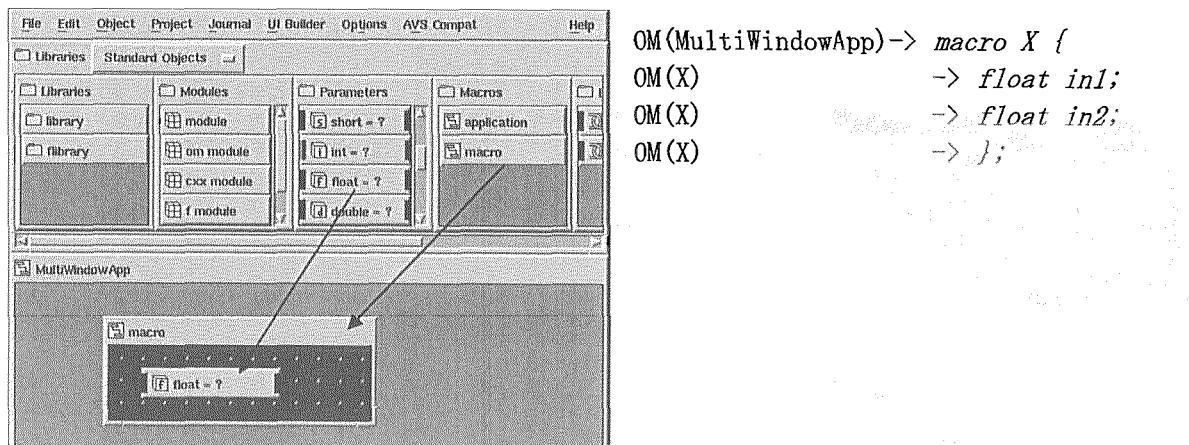
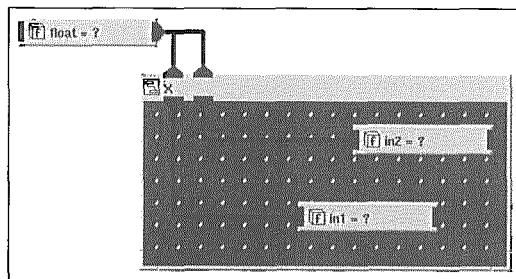


図 5-6 マクロ・オブジェクト

○ リンク・オブジェクト

リンク・オブジェクトは、複数の入出力をまとめ場合やデータの名前を明記する場合に利用します。

リンク・オブジェクトがない場合



リンク・オブジェクトがある場合

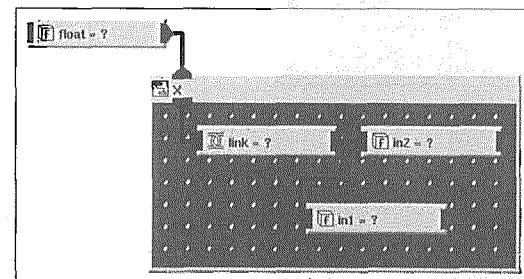


図 5-7 リンク・オブジェクト

5.1.4 計算オブジェクトの作成

プリミティブ・オブジェクトとグループ・オブジェクトを組み合わせて、演算を行うモジュールを作成します。

- (1) グループ・オブジェクトをインスタンスします。
- (2) group オブジェクトの中に float オブジェクトを 3 つインスタンスします。(図 5-8)

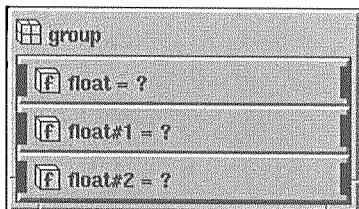


図 5-8 オブジェクトのインスタンス

- (3) 各オブジェクトの名前を下記のように変更します。(図 5-9)



図 5-9 オブジェクト名の変更

- (4) 入出力ポートを作成します。

ポートを作成する場合は、茶色いポート上でマウス右ボタンを押します。(図 5-10)

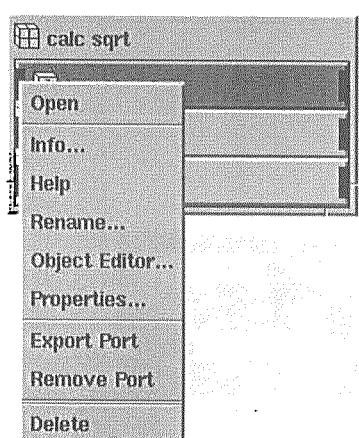


図 5-10 入出力ポートの作成

(5) output オブジェクトに計算式を入力します。(図 5-11)

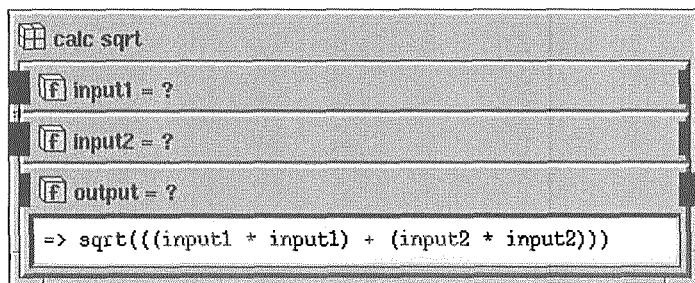


図 5-11 計算式の入力

以上で計算モジュールの作成は終了です。

参考：同様のことを VCP で行うと以下のようになります。

```
OM(MultiWindowApp) ->      group calc_sqrt;
OM(MultiWindowApp) ->      calc_sqrt{
OM(calc_sqrt)    ->      float input1;
OM(calc_sqrt)    ->      float input2;
OM(calc_sqrt)    ->      float output;
OM(calc_sqrt)    ->      float input1<NEportLevels={2, 0}>;
OM(calc_sqrt)    ->      float input2<NEportLevels={2, 0}>;
OM(calc_sqrt)    ->      float output<NEportLevels={0, 2}>;
OM(calc_sqrt)    ->      output=>sqrt((input1*input1)+(input2*input2));
OM(calc_sqrt)    ->      };
OM(MultiWindowApp) ->
```

5.2 ユーザインターフェース (UI)

UI モジュールは、User Interface ライブラリに登録されています。(図 5-12)

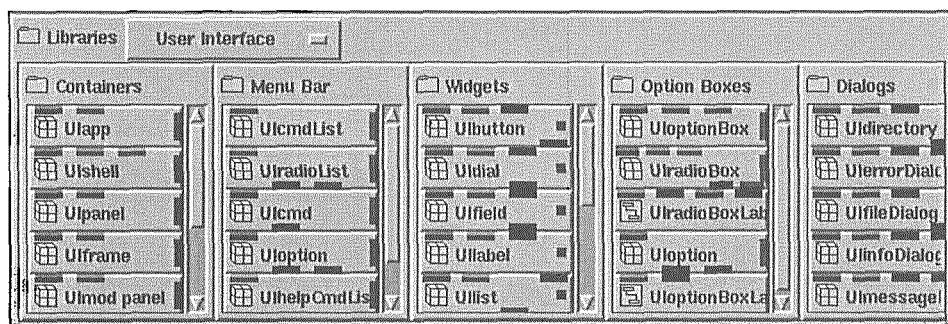


図 5-12 UI モジュール・ライブラリ

5.2.1 基本操作

図 5-13 のネットワークを作成したとします。

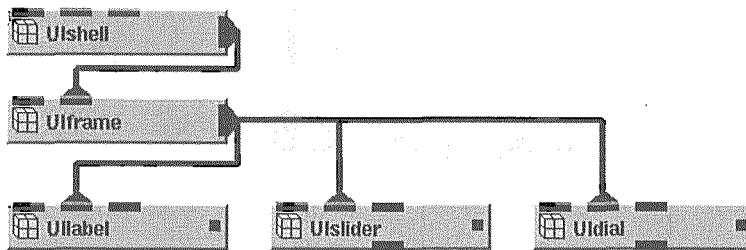


図 5-13 ネットワーク例

5.2.2 位置やサイズの指定

ラベルやダイアルの位置や大きさを指定する場合は、UI モジュールの中にあるサブオブジェクトで設定します。

実際の位置を直接指定する方法（図 5-14）と相対的に指定する方法（図 5-15）の二通りがあります。

○ 直接指定

UI モジュールを開き、位置 (x, y)、幅 (width)、高さ (height) を数値で入力します。

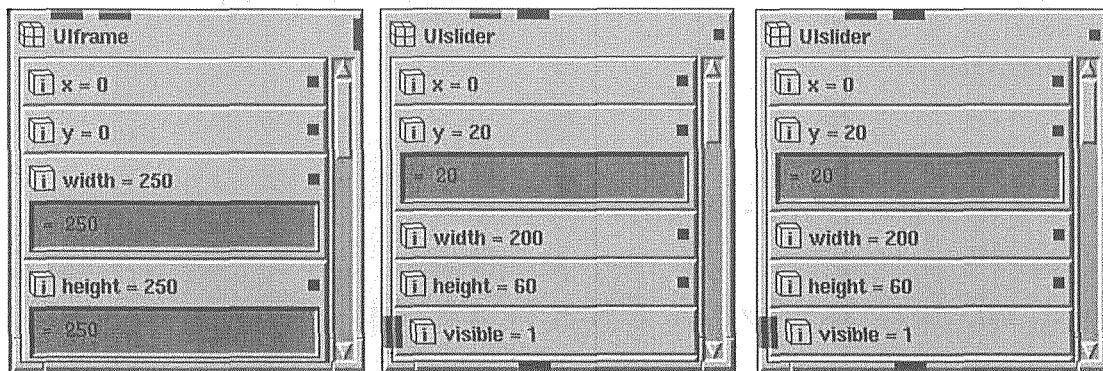


図 5-14 直接指定

○ 相対指定

UI モジュールを開き、以下のように設定します。

UIframe.width	=>	Uishell.clientWidth
UIframe.height	=>	Uishell.clientHeight
UIlabel.x	=>	((UIframe.width - width)/2)
UIslider.x	=>	((UIframe.width - width)/2)
UIslider.y	=>	((UIlabel.y + UIlabel.height) + 10)
UIDial.x	=>	((UIframe.width - width)/2)
UIDial.y	=>	((UIslider.y + UIslider.height) + 10)

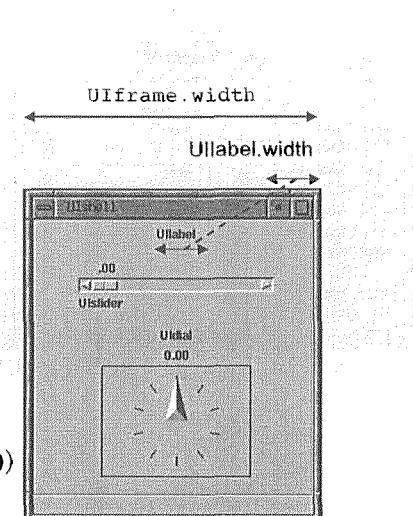


図 5-15 相対指定

5.2.3 自作モジュールの UI

UI モジュールを利用することで、自作モジュールの GUI を作成することができます。

実際に、5.2 で作成したパネルを用いて、5.1.4 で作成した計算オブジェクトをパネルから操作するネットワークを作成する例を示します。

図 5-16 のように calc_sqrt モジュールを接続します。

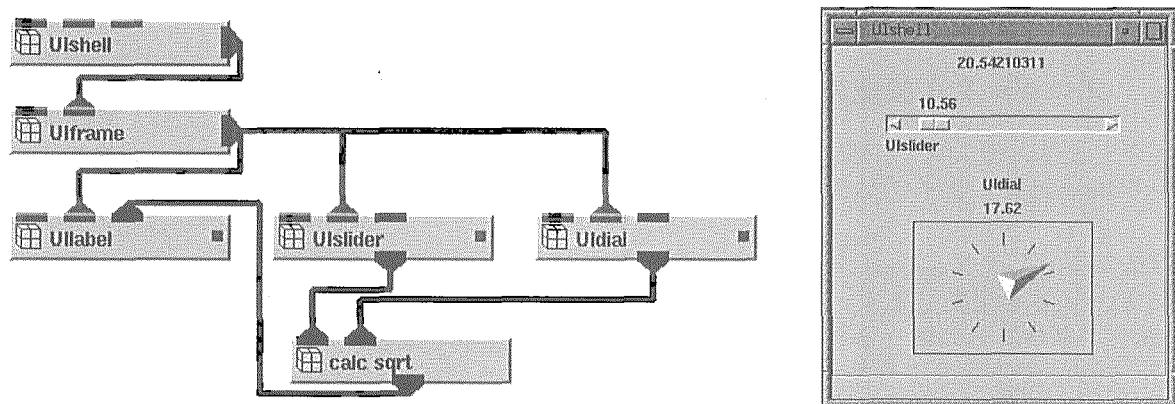


図 5-16 自作モジュールの UI

Ullabel.label と calc_sqrt.output との接続は、一度 Ullabel.label を開き、現在の設定を削除してから接続します。

5.2.4 コントロールパネルへの登録

UImod_panel モジュールを利用すると自作モジュールのパネルをコントロールパネルに表示させることができます。

図 5-17 のように Uishell と UImod_panel を入れ替えます。

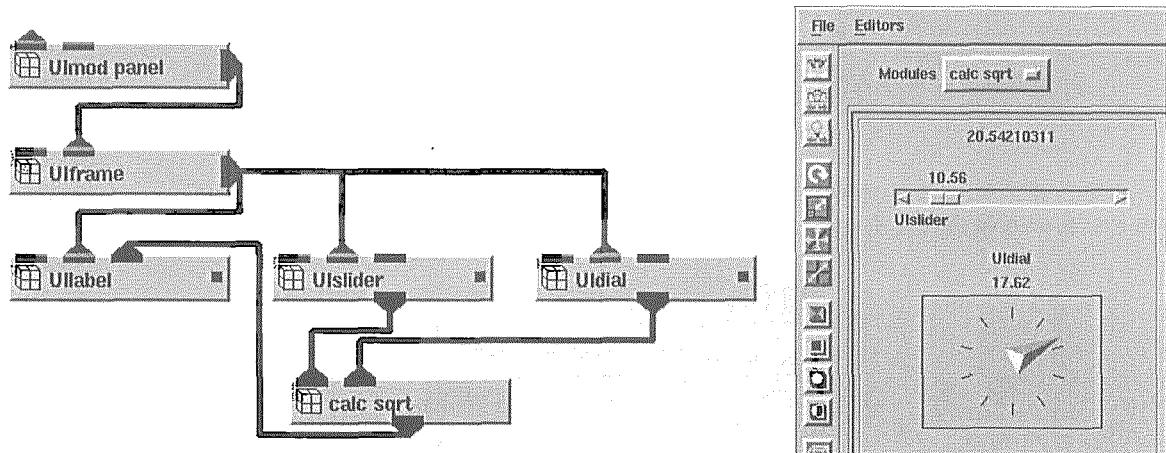


図 5-17 コントロールパネルへの登録

5.3 モジュール作成

モジュール作成の主な手順は以下の通りです。

1. プロジェクト環境の設定
2. モジュールの定義(Vファイルの作成)
3. ソースファイルの自動生成と編集
4. コンパイル・リンク
5. 実行テスト
6. 保存

5.3.1 プロジェクト環境の設定

プロジェクトとは、モジュールを作成するための開発環境(ディレクトリ)のことをいいます。ソースファイルや Makefile、ライブラリなどはプロジェクト環境配下に保存されます。

以下の方法で作成します。

- (1) AVS/Express を起動します。
- (2) 上部プルダウンメニューから [Project]-[Save As] を選択します。
- (3) Save Project ウィンドウでプロジェクト環境を保存するディレクトリ指定します。
新たにディレクトリを作成する場合は、Selection フィールドにディレクトリ名を記入します。

プロジェクト環境を設定すると以下のファイル(ディレクトリ)が生成されます。

avsenv	プロジェクト環境のパスが指定されています。このファイル内に定義されているパスから実行時に必要なファイルが検索されます。
bin/	コンパイルによって生成される実行ファイルが格納されます。
include/	コンパイルに必要な include ファイルが格納されます。
lib/	コンパイル時に生成されるオブジェクト(.o)ファイルが格納されます。
v/	モジュールの V ファイルが格納されます。

次回からこのプロジェクト環境で AVS/Express を利用する場合には、以下のように起動します。

```
% xp -project <プロジェクト環境名>
```

5.3.2 計算モジュールの作成

プリミティブ・オブジェクトを利用して演算を行うモジュールを作成する例を示します。
作成後のモジュール構成は図 5-18 のようになります。

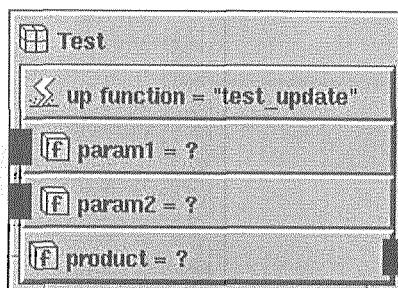


図 5-18 Test モジュール

モジュール作成には、Add Module Tool を利用します。

Libraries プルダウンメニューから Library Workspaces を選択します。

Workspace_1 を選択し、上部プルダウンメニューから [Object]-[Add Module] を選択してください。

その後、図 5-19、図 5-20、図 5-21、図 5-22、図 5-23、図 5-24 の順に操作します。

Add Module Tool(ステップ 1) 計算モジュールの作成

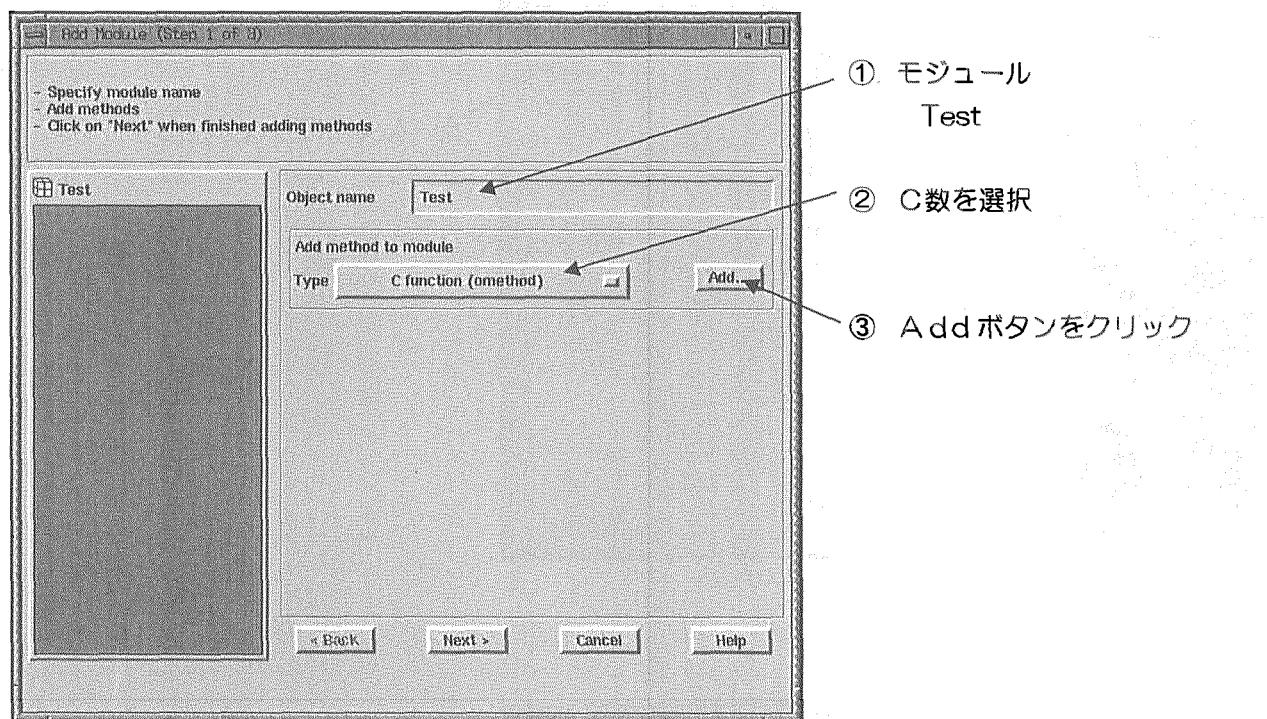


図 5-19 Add Module (Step 1 of 3)

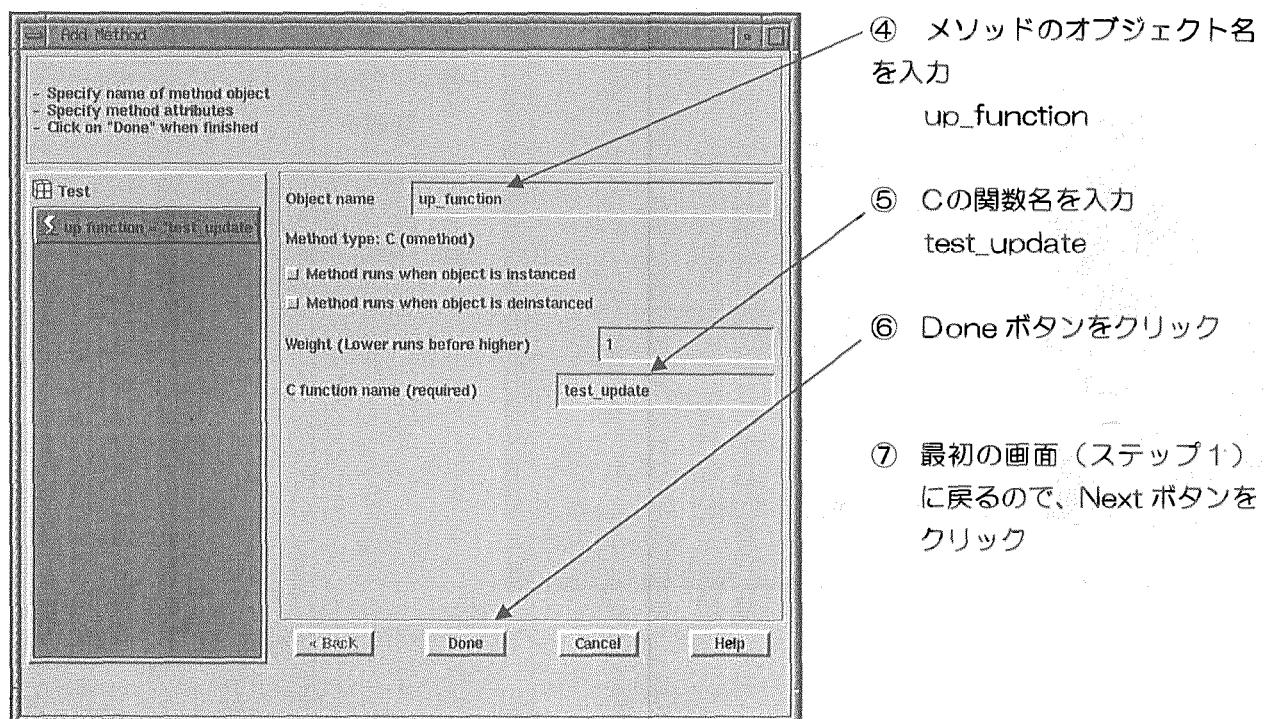


図 5-20 Add Method

Add Module Tool(ステップ 2)

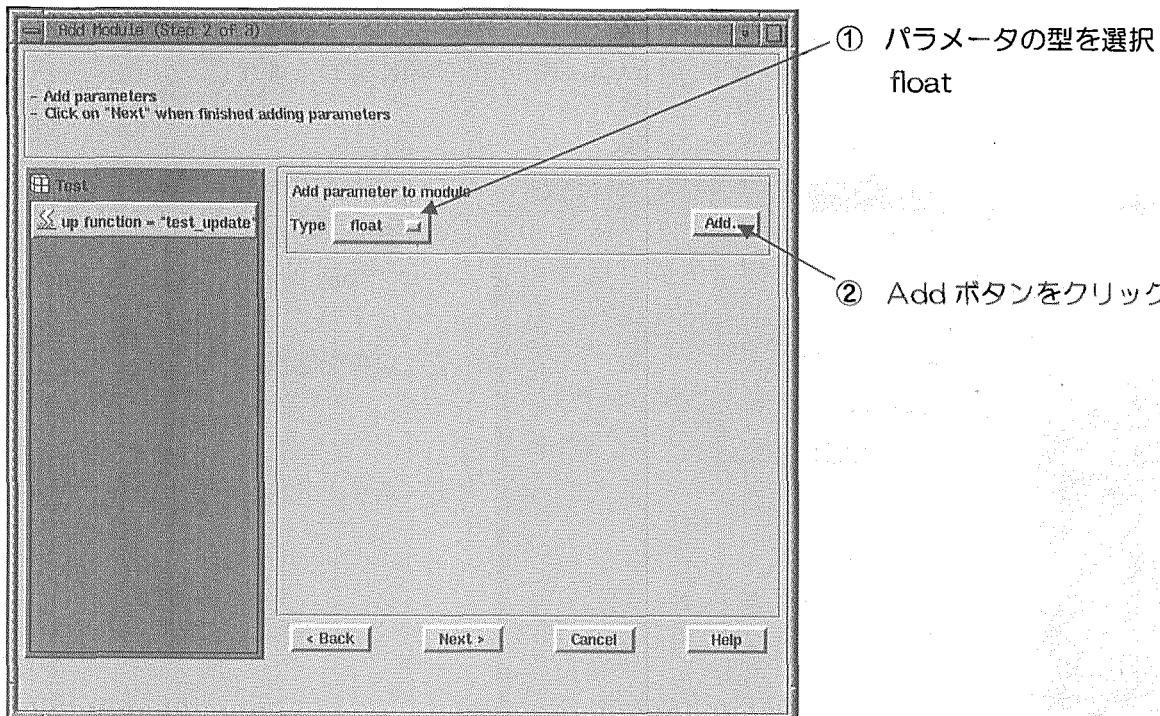


図 5-21 Add Module(Step 2 of 3)

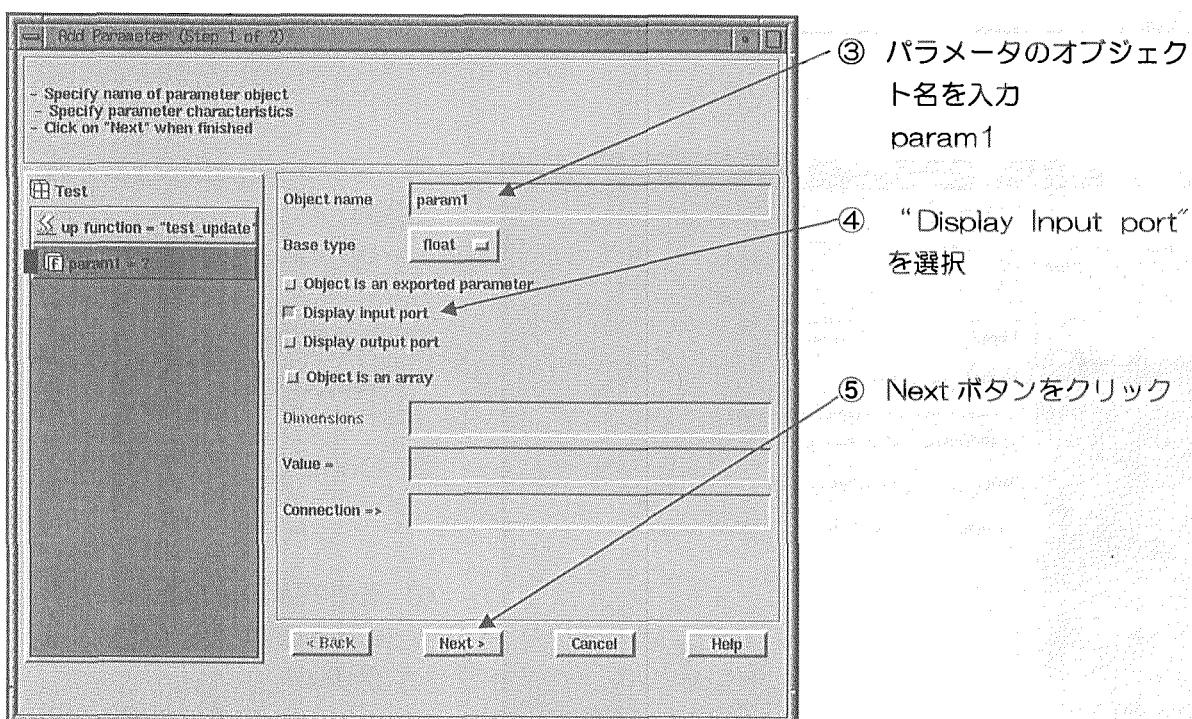


図 5-22 Add Parameter(Step 1 of 2)

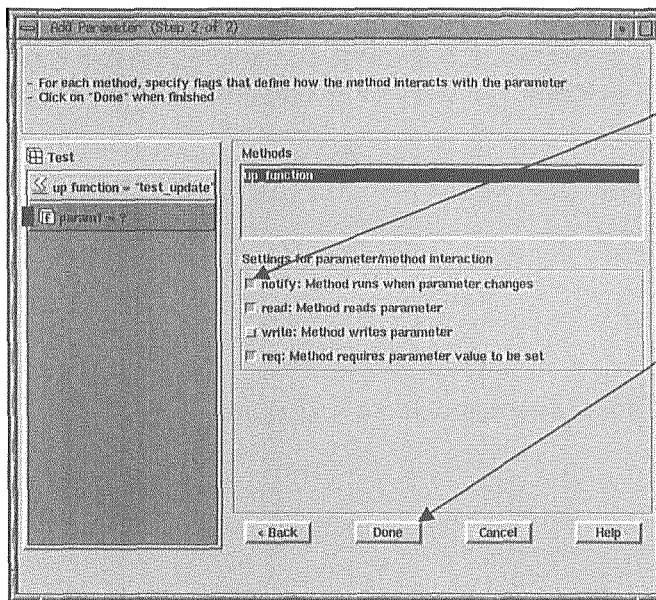


図 5-23 Add Parameter (Step 2 of 2)

⑥ パラメータとメソッドの相互関係を設定
notify, read, req の 3つにチェック

⑦ Done ボタンをクリック

パラメータとメソッドの相互関係について

- notify : このパラメータが変更された時にメソッドが実行される
- read : メソッドは、このパラメータのデータを読み込む
- write : メソッドは、このパラメータにデータを書き込む
- req : メソッドが実行されるためには、このパラメータは必須

上記⑦の Done ボタンをクリックするとステップ 2 の画面に戻ります。

ステップ 2 (①~⑦) を繰り返し、もう一つの入力ポート param2 を作成します。

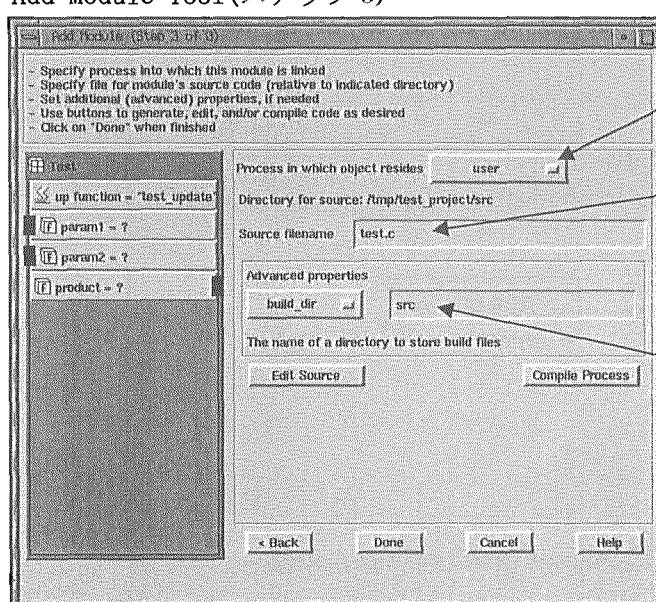
さらに、ステップ 2 (①~⑦) を繰り返し、出力ポート product を作成します。

出力ポートを作成する場合は、④の設定は "Display output port" を選択してください。

また、⑥の相互関係の設定では "write" のみにチェックを付けてください。

3つの入出力ポートに対する設定を終了するとステップ 2 に戻るので、Next ボタンをクリックします。

Add Module Tool (ステップ 3)



① プロセスを指定
user

② ソースファイル名を入力
test.c

③ ソースディレクトリを入力
src

図 5-24 Add Module (Step 3 of 3)

④ Edit Source ボタンをクリック

ソースファイル (test.c) がないため、テンプレートを作成する旨を伝えるメッセージが表示されます。Yes ボタンをクリックするとソースファイルを編集するウインドウが表示されるので、下記の test.c を参考に 1 行追加してください。

ソースファイル: test.c

```
#include "user.h"

int
test_update(OMobj_id Test_id, OMevent_mask event_mask, int seq_num)
{
    /*****
    /* Declare variables */
    *****/
    double param1;
    double param2;
    double product;

    /*****
    /* Get input values */
    *****/
    /* Get param1's value */
    if (OMget_name_real_val(Test_id, OMstr_to_name("param1"), &param1) != 1)
        param1 = 0.0;

    /* Get param2's value */
    if (OMget_name_real_val(Test_id, OMstr_to_name("param2"), &param2) != 1)
        param2 = 0.0;

    /*****
    /* Function's Body */
    *****/
    ERRverror("",ERR_NO_HEADER | ERR_INFO,
    "I'm in function: test_update generated from method: Test.up_function\n");

    product = param1 * param2; ← この行を追加

    /*****
    /* Set output values */
    *****/
    /* Set product's value */
    OMset_name_real_val(Test_id, OMstr_to_name("product"), product);
    return(1);
}
```

⑤ Compile Process ボタンをクリック

コンパイルメッセージが表示されます。エラーが表示されなければコンパイルは終了です。
リターンキーを押してウィンドウを閉じてください。

5.3.3 実行テスト

作成した Test モジュールを Workspace_1 からインスタンスします。
Test モジュールを開き、param1 と param2 に値を入力します。

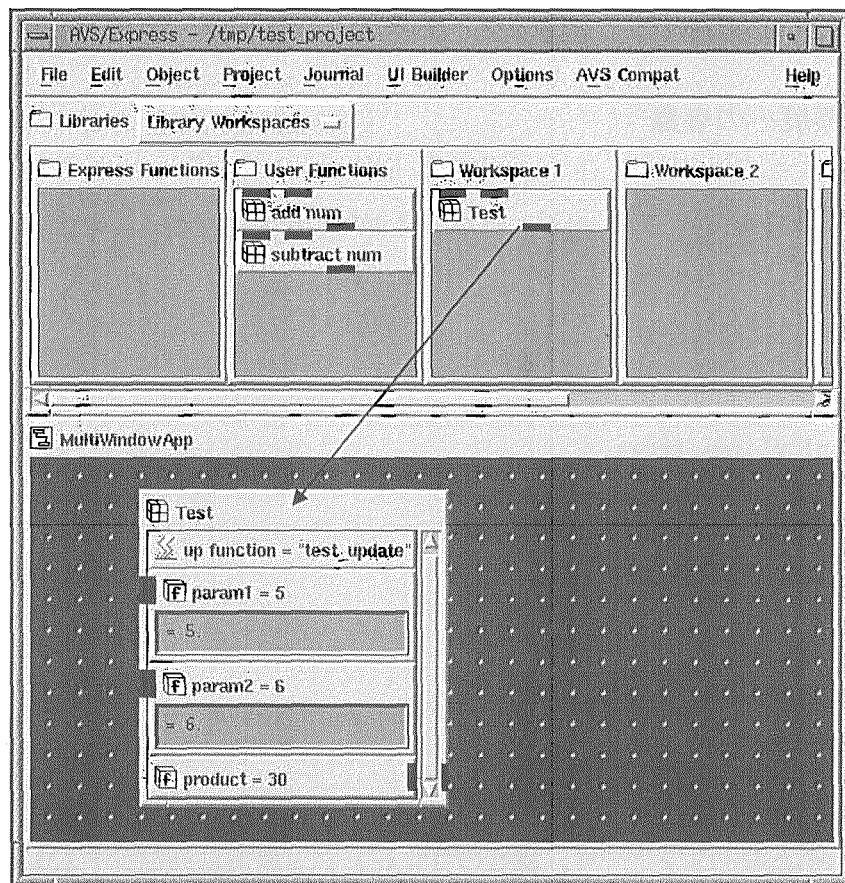


図 5-25 Test モジュールの実行

数値を入力するとモジュールが起動し、計算結果が product に格納されます。(図 5-25)

このとき、Error Dialog が表示されます。これは自動生成されるソースの中にモジュールが実行されたことをメッセージとして出力する行が含まれているためです。

ソースファイルの中にある ERRerror で始まる行を削除し、再度コンパイルを行うと表示されなくなります。

モジュールが正常に動作することを確認した場合は、Add Module Tool の Done ボタンをクリックし、ウィンドウを閉じます。

5.3.4 モジュールの保存

この作業を行わずに AVS/Express を終了させると、Add Module Tool で行った設定が保存されません。モジュールを作成した場合は、必ずモジュールの保存を行ってください。

Workspace_1 にある Test モジュールを選択します。

上部メニューから [Object]-[Save Object] を選択し、ファイル名を指定します。

モジュールの保存先に制限はありませんが、ここではプロジェクト環境配下にある v ディレクトリの下に保存します。

<プロジェクト環境>/v/test.v

5.3.5 プロジェクト環境への登録

AVS/Express を起動したときに、作成した Test モジュールが含まれた状態で起動されるように設定します。

v/temp1.v ファイルを以下のように編集します。

```
"$XP_PATH<0>/v/temp1.v" Templates {
    WORKSPACE_1 {
        $include test.v
    };
};
```

Text モジュールを保存する際にモジュール単体の保存 ([Object]-[Save Object]) ではなく、[Project]-[Save] を選択してプロジェクト全体を保存すると、temp1.v ファイルに今の状態が保存されます。

\$include は、V ファイルを読み込むコマンドです。複数のプロジェクト環境をマージする場合や自作モジュールの数が多い場合に利用するとモジュール修正などの作業が行いやすくなります。

次回から Test モジュールを含んだ AVS/Express を起動する場合には、以下のように行います。

```
% xp -project <プロジェクト環境>
```

AVS/Express Developer Edition

Version: 5.1 fcs sg6

Project: /tmp/test_project ← プロジェクト環境が表示されていることを確認してください。

5.4 モジュール作成

5.4.1 ソースファイルと再コンパイル

ソースファイルは、プロジェクト環境配下で指定したディレクトリに作成されます。
Makefile はプロジェクト環境配下に作成されます。

```
<プロジェクト環境>/src/test.c
          /user.mk
```

モジュール内にあるオブジェクトとデータのやり取りを行う場合は、オブジェクト毎に付けられる「オブジェクト ID」を利用して行います。

ソースファイル(test.c)にある OMget_name_real_val 関数は、オブジェクトの名前から ID を検索し、そのオブジェクトの値を入手する関数です。(図 5-26)

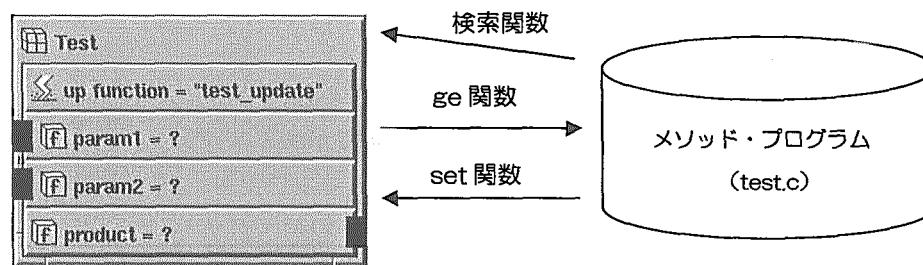


図 5-26 モジュール・オブジェクトと関数

OMget_name_real_val 関数は、以下のように書き換えることもできます。

```
param1_id=OMfind_subobj(Test_id, OMstr_to_name("param1"), OM_OBJ_RD);
OMget_real_val(param1_id, &param1);
```

AVS/Express を終了した後でソースファイルを修正し、再コンパイルする場合には以下のように行います。

```
make -f user.mk
```

5.4.2 モジュール定義

モジュールを保存した V ファイルには以下の情報が記述されています。

例) test.v

```
module Test<process="user", src_file="test.c", build_dir="src"> {
    method+req up_function(
        . param1+read+notify+req,
        . param2+read+notify+req,
        . product+write) = "test_update";
    float param1<NEportLevels={2, 0}>;
    float param2<NEportLevels={2, 0}>;
    float product<NEportLevels={0, 2}>;
};


```

The diagram shows annotations for the V module definition:

- モジュールの定義**: Points to the `module Test` declaration.
- プロセス・ソースファイルの指定**: Points to the attributes `process="user"`, `src_file="test.c"`, and `build_dir="src"`.
- メソッドの宣言**: Points to the `method+req up_function` declaration.
- 入出力パラメータの宣言**: Points to the parameters `param1`, `param2`, and `product`.
- ポートの宣言:{入力ポート階層数、出力ポート階層数}**: Points to the port declarations with their respective layer ranges.

5.4.3 プロセス管理

モジュール作成時に設定するプロセスタイプには、以下の二種類があります。

■ user プロセス

作成したモジュールはインストールされている AVS/Express システムの管理下で、別プロセスとして動作します。モジュールを使用する場合は、下記のようにインストールされている AVS/Express を起動します。個々のモジュールテストを行う際に使用します。

```
% /appli/Express/express/bin/sg6/xp -project <プロジェクト環境>
```

長所：AVS/Express を終了することなくコンパイルが可能

短所：単一プロセスではないため、処理が遅い

■ express プロセス

モジュールをコンパイルする際にインストールされている AVS/Express システムの実行バイナリを再リンクします。そのため、モジュールを使用する場合は、プロジェクト環境配下の `express` バイナリを実行します。最終的にアプリケーションを作成する場合にはこの方法を使用します。

```
% <プロジェクト環境>/bin/sg6/express -project <プロジェクト環境>
```

長所：作成したモジュールは、AVS/Express システム・プロセス中の 1 関数になるため、処理速度が遅くならない

短所：コンパイル時に、一旦 AVS/Express を終了しなければならない

5.4.4 プロジェクト環境の再構築

以下の場合には、プロジェクト環境の再構築を行う必要があります。

- 既存のモジュールVファイルを編集する場合
- vi やメモ帳などのエディタを利用してモジュールVファイルを作成した場合
- プロジェクト環境やモジュールVファイル、ソースコードを第三者から入手した場合
- プロジェクト環境を別ディレクトリにコピーした場合

プロジェクト環境の再構築を行う場合は、base コマンドを利用します。base コマンドは以下のディレクトリにあります。

```
/appli/Express/express/bin/sg6/
```

base コマンドを実行する場合はプロジェクト環境に移動します。
実行方法には、以下の二種類があります。

プロジェクト環境の設定を整えた後、コンパイルを行う方法

```
% cd <プロジェクト環境>
% base -project . -gen_proc user
    ("OM(Root)->"と表示された場合は "$quit"と入力し、終了します)
% make -f user.mk
```

プロジェクト環境の設定からコンパイルまでを一度に行う方法

```
% cd <プロジェクト環境>
% base -project . -comp_proc user -exit
```

express プロセスの場合は、上記 user 部分を express に置き換えます。
プロジェクト環境を第三者から入手した場合やプロジェクト環境を別ディレクトリにコピーした場合は、avsenv ファイルの書き換えを行った後に base コマンドを実行します。

5.4.5 user プロセスから express プロセスへの変更

5.3 で作成した計算モジュール (user プロセス) を express プロセスに変更します。

(1) モジュール定義Vファイルの変更

```
% cd <プロジェクト環境>/v
% vi test.v


module Test<process="express",src_file="test.c",build_dir="src">


```

(2) ソースファイルにある include 分の変更

```
% cd <プロジェクト環境>/src
% vi test.c


# include <express.h>


```

(3) base コマンドの実行

```
% base -project . -comp_proc express -exit
```

(4) express バイナリの実行

```
% cd <プロジェクト環境>
% ./bin/sg6/express -project <プロジェクト環境>
```

5.5 Field データを扱うモジュールの開発

5.5.1 Field データの構造

AVS/Express のデータタイプは、フィールド(Field)と呼ばれる構造体を利用しています。可視化を行いたいデータを読み込むとフィールド構造体にデータがセットされ、各モジュール間の受け渡しが行われます。

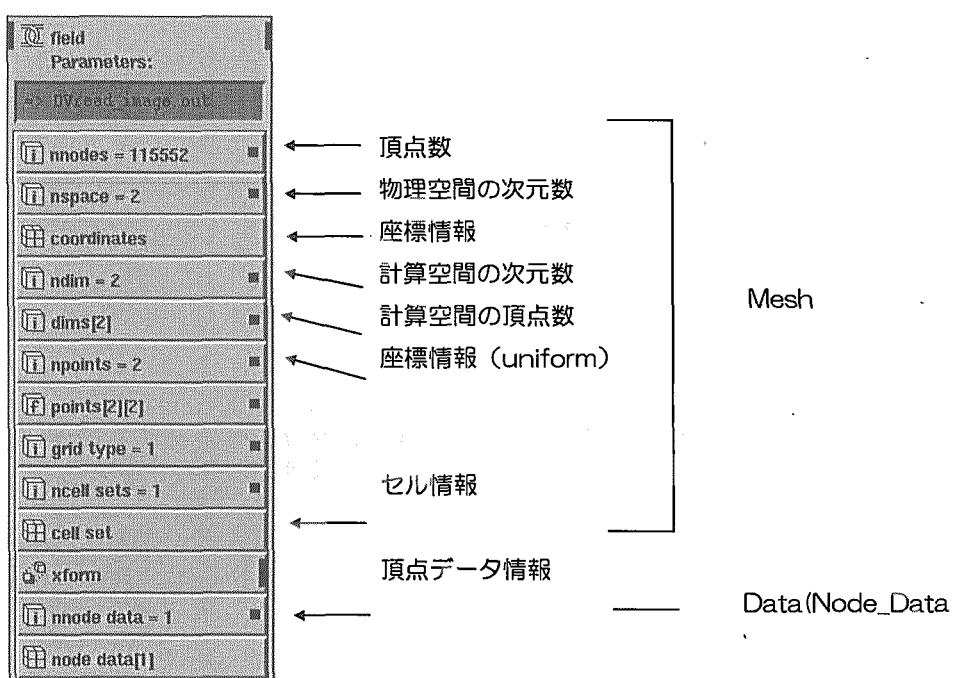


図 5-27 Read_Image.field オブジェクト

図 5-27 は Read_Image モジュールに含まれる出力データ field(リンク・オブジェクト)を開いた例です。

このフィールドデータのオブジェクトは、座標値や物理量を保持する複数のオブジェクトから構成されています。これらのオブジェクト階層は以下のようになっています。(図 5-28)

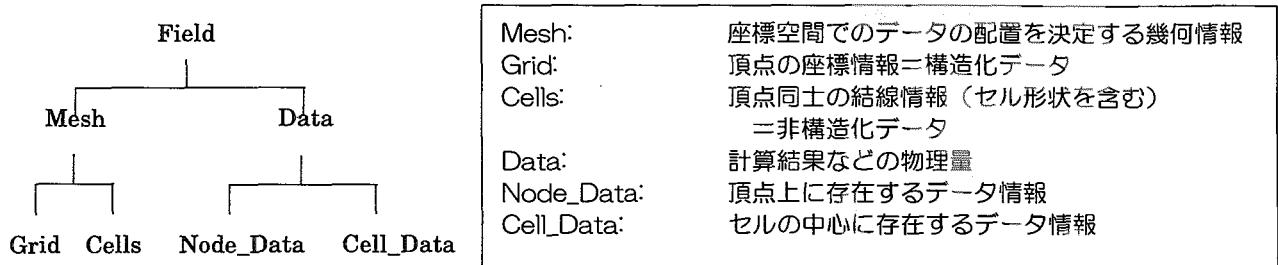


図 5-28 フィールド構造体

フィールドデータは非常に多くのオブジェクトによって構成されていますが、可視化するデータタイプによってアクセスされるオブジェクト(必要なオブジェクト)が決定されます。

5.5.2 サンプル(メッシュ形状)

3 次元空間に存在する 3×2 のメッシュデータを想定します。(図 5-29)

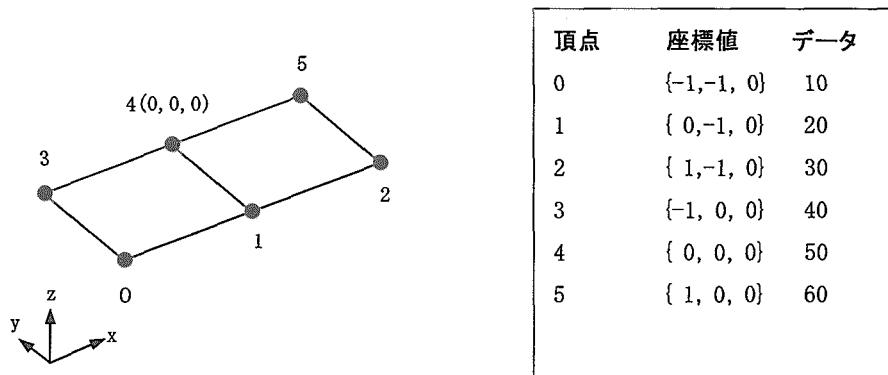


図 5-29 メッシュデータ

このデータは Field_Unif と呼ばれる直行等間隔のオブジェクトを用いて設定することができますが、ここでは Field_Struct オブジェクトを用いて irregular 型のデータとして設定します。

(1) Field_Struct オブジェクトのインスタンス

Accessories.Field_Schema から Field_Struct モジュールをインスタンスします。

(2) 数値の設定

Field_Struct モジュールのサブオブジェクトに下記の値を設定します。(図 5-30)

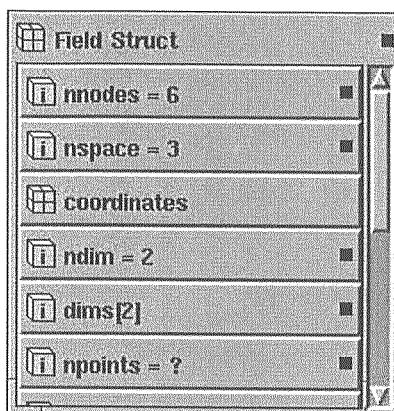


図 5-30 Field_Struct モジュール

```

nspace = 3
coordinates.values
= {{-1, -1, 0}, {0, -1, 0},
  {1, -1, 0}, {-1, 0, 0},
  {0, 0, 0}, {1, 0, 0}}
ndim   = 2
dims   = {3, 2}
nnode_data      = 1
node_data.veclen = 1
node_data.values
= {10, 20, 30, 40, 50, 60}

```

配列データを入力した場合は、Ctrl キーを押しながらリターンキーで確定します。

Field_Struct オブジェクトの出力ポートを追加し、図 5-31 のネットワークを作成します。

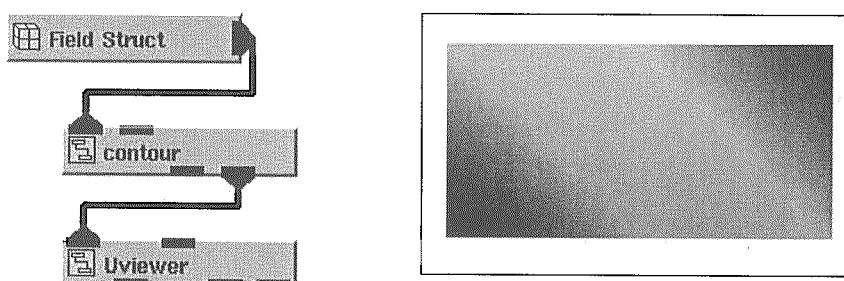


図 5-31 メッシュデータの確認

5.5.3 プロジェクト環境の設定

5.3.1 章で作成したプロジェクト環境を利用します。

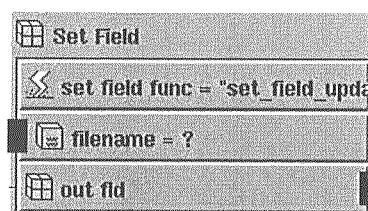
新たにプロジェクト環境を設定される場合は、以下の方法で作成します。

- (1) AVS/Express を起動します。
- (2) 上部プルダウンメニューから [Project]-[Save As] を選択します。
- (3) Save Project ウィンドウでプロジェクト環境を保存するディレクトリ指定します。
新たにディレクトリを作成する場合は、Selection フィールドにディレクトリ名を記入します。

5.5.4 フィールドデータを扱うモジュールの作成

フィールドデータを出力(セット)するモジュールを作成します。

作成後のモジュール構成は図 5-32 のようになります。

図 5-32 Set_Field モジュール
- 78 -

モジュール作成には、Add Module Tool を利用します。

Libraries プルダウンメニューから Library Workspaces を選択します。

Workspace_1 を選択し、上部プルダウンメニューから [Object]-[Add Module] を選択してください。

図 5-33、図 5-34、図 5-35、図 5-36、図 5-37、図 5-38、図 5-39、図 5-40、図 5-41 の説明にしたがつて作成していきます。

Add Module Tool(ステップ 1)

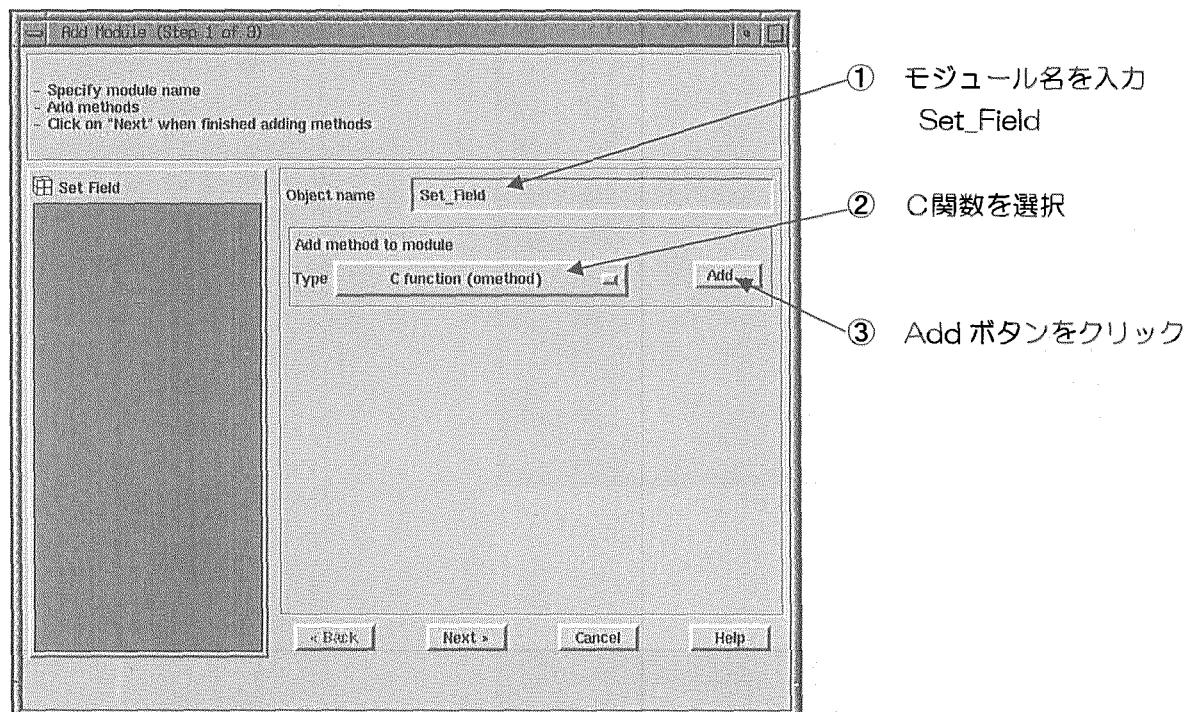


図 5-33 Add Module(Step 1 of 3)

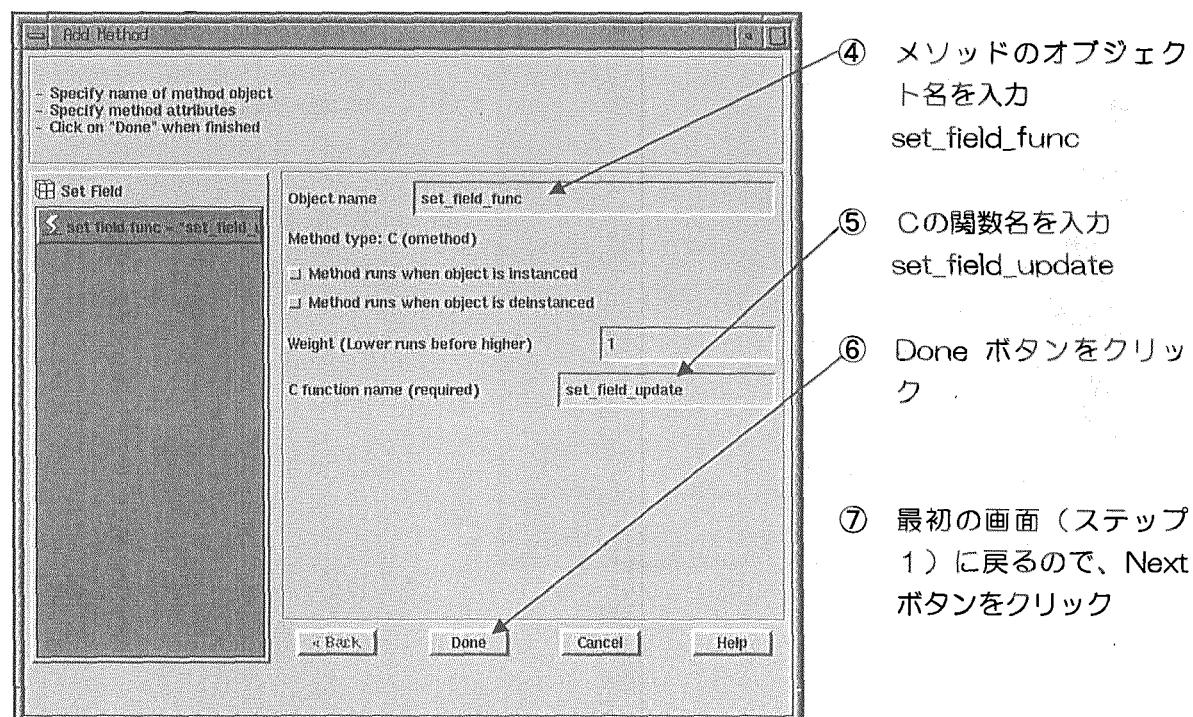


図 5-34 Add Method

Add Module Tool(ステップ 2 : filename)

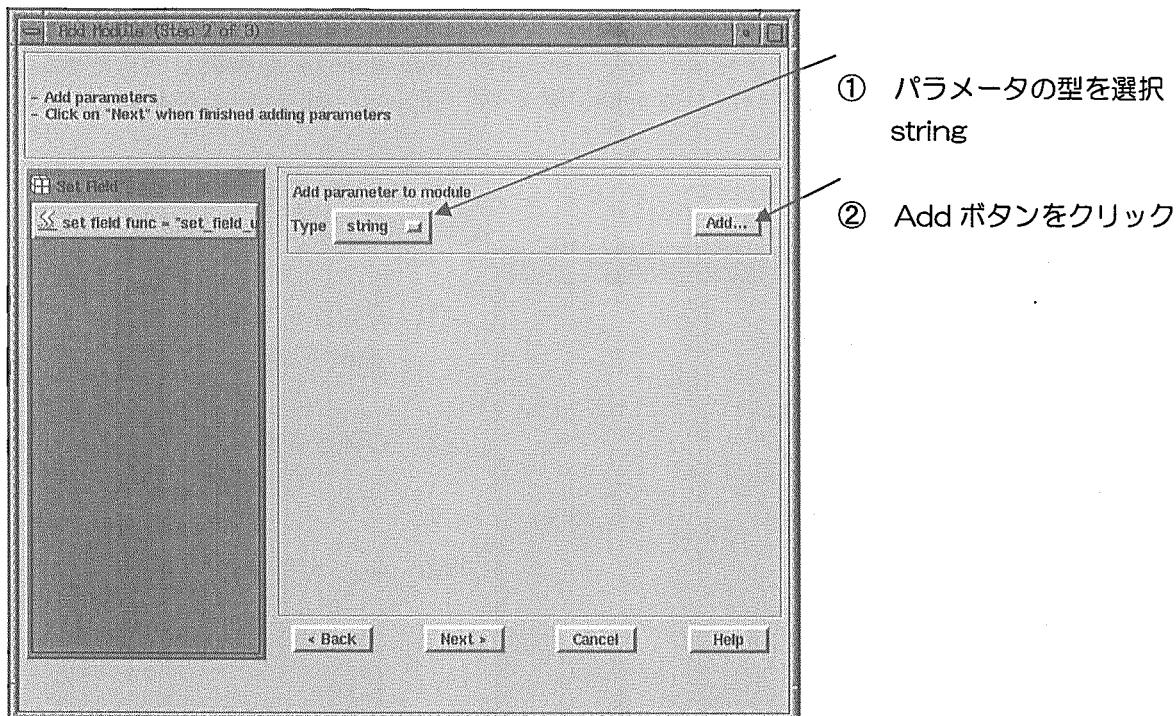


図 5-35 Add Module(Step 2 of 3)

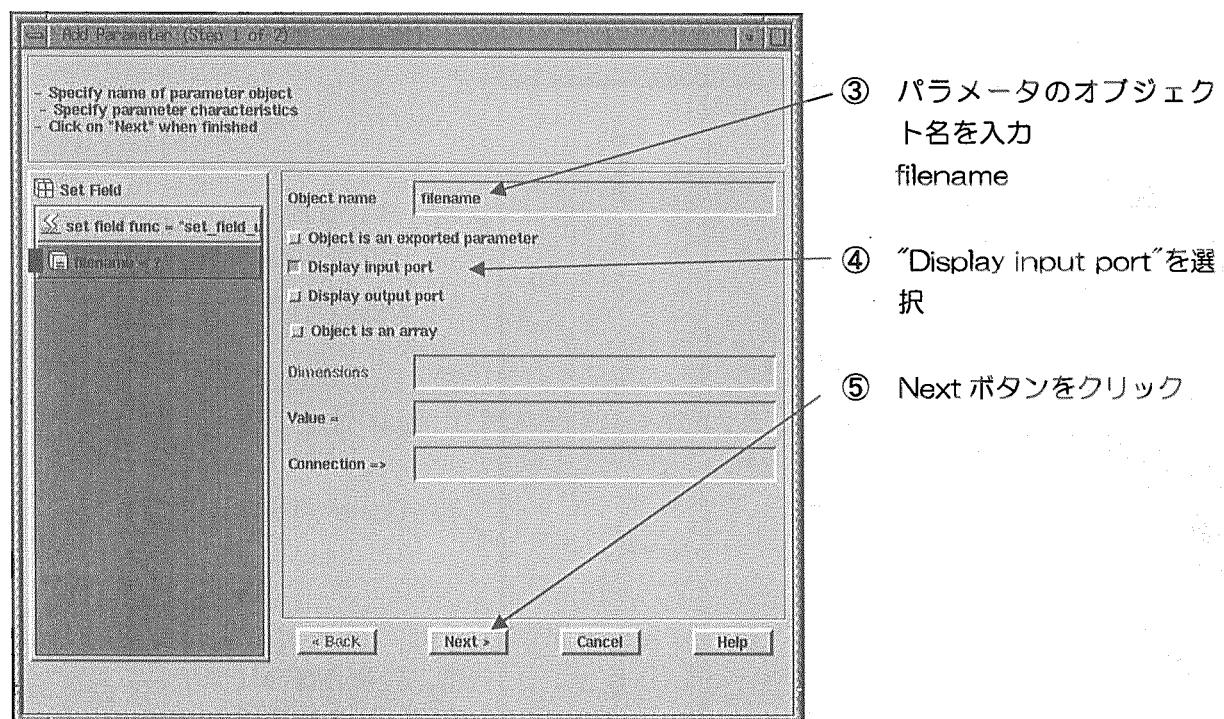


図 5-36 Add Parameter(Step 1 of 2)

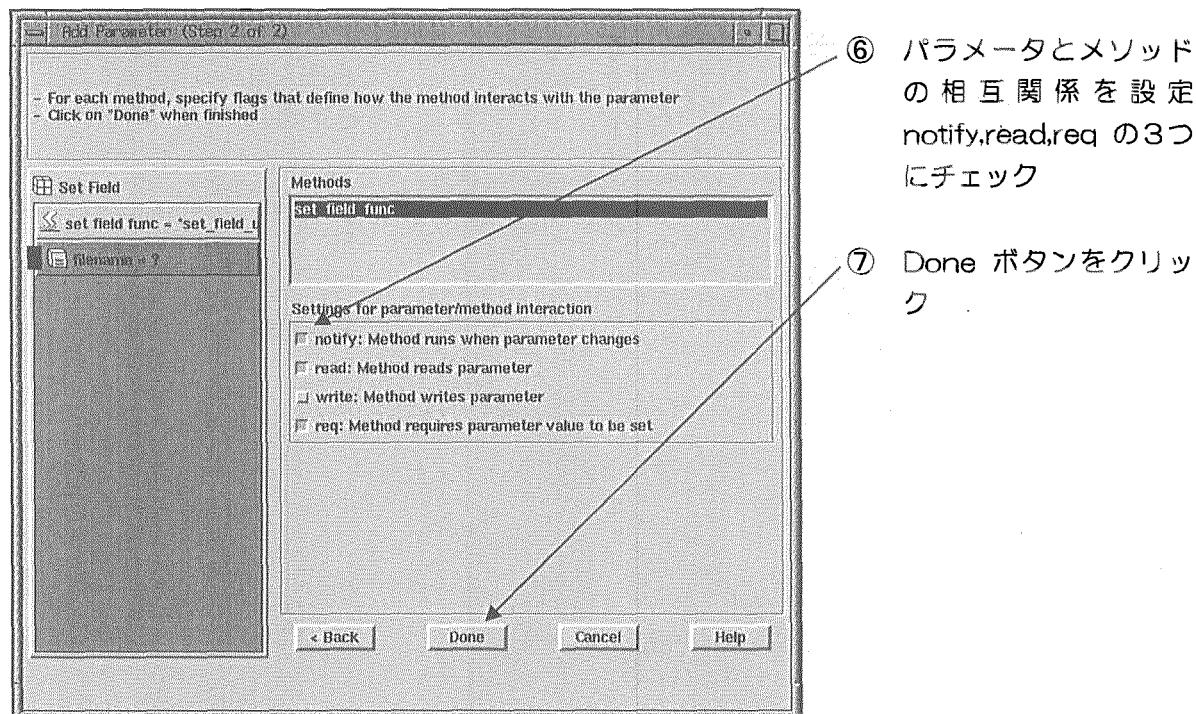


図 5-37 Add Parameter (Step 2 of 2)

Add Module Tool(ステップ 2:out_fld)

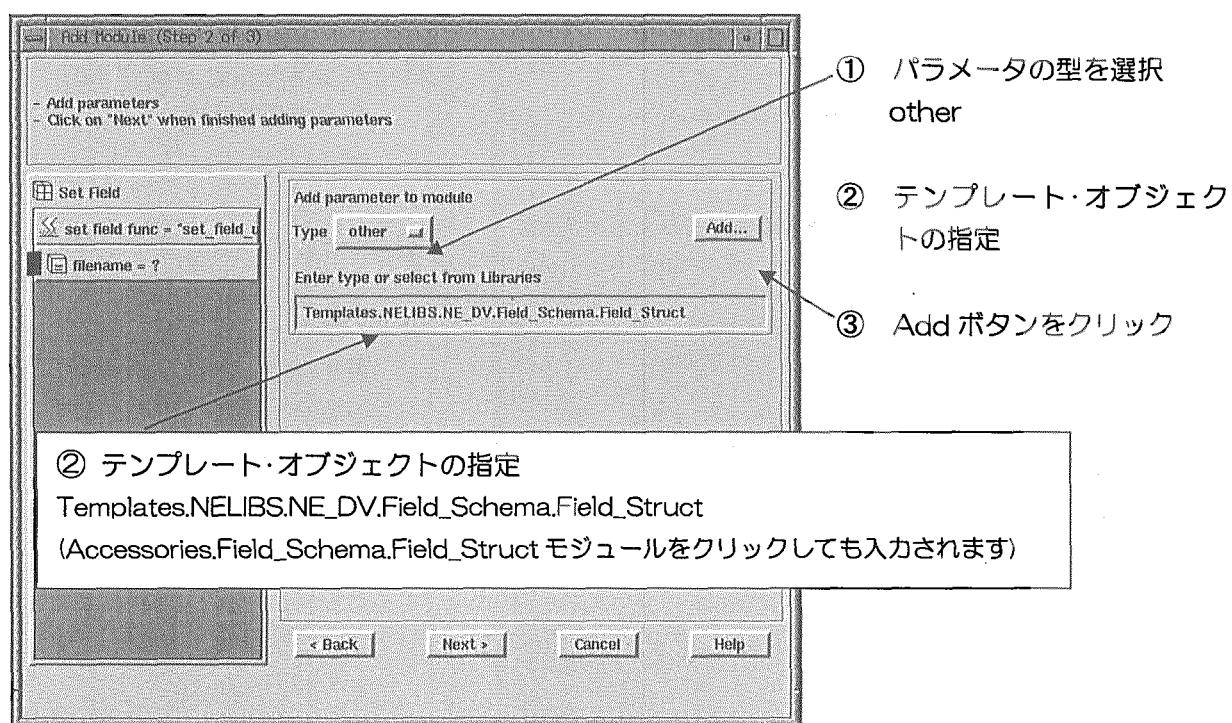


図 5-38 Add Module (Step 2 of 3)

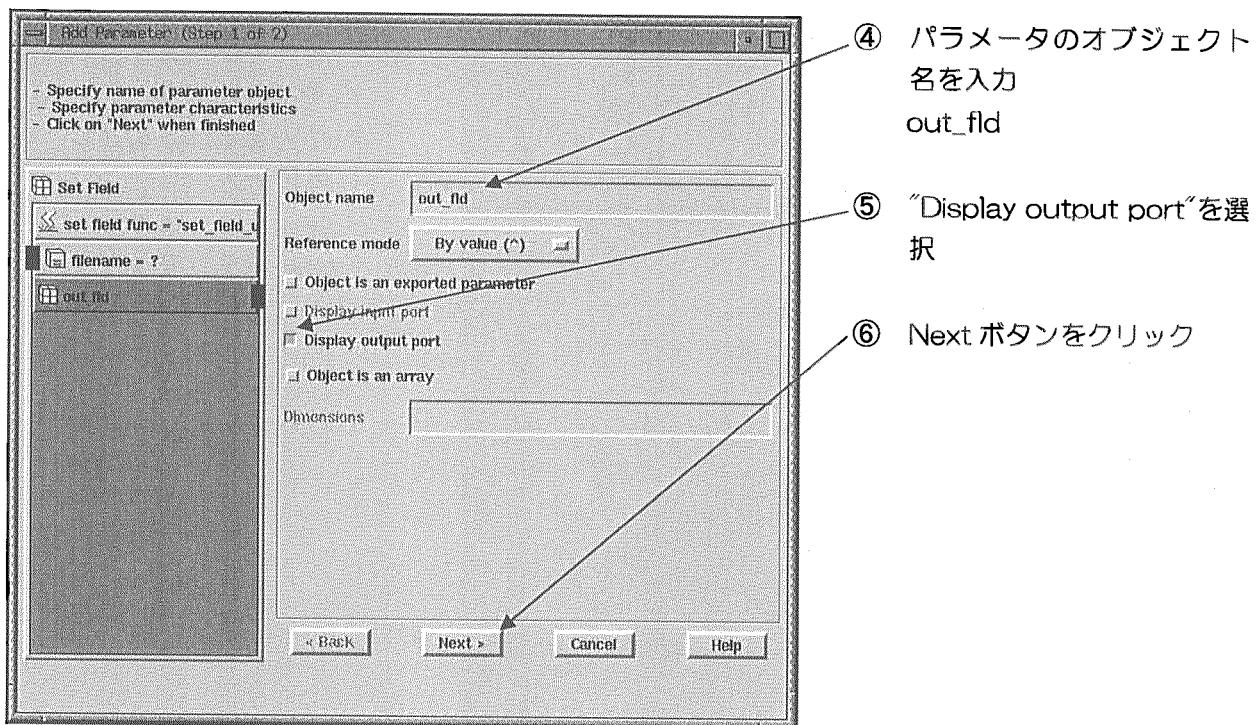


図 5-39 Add Parameter (Step 1 of 2)

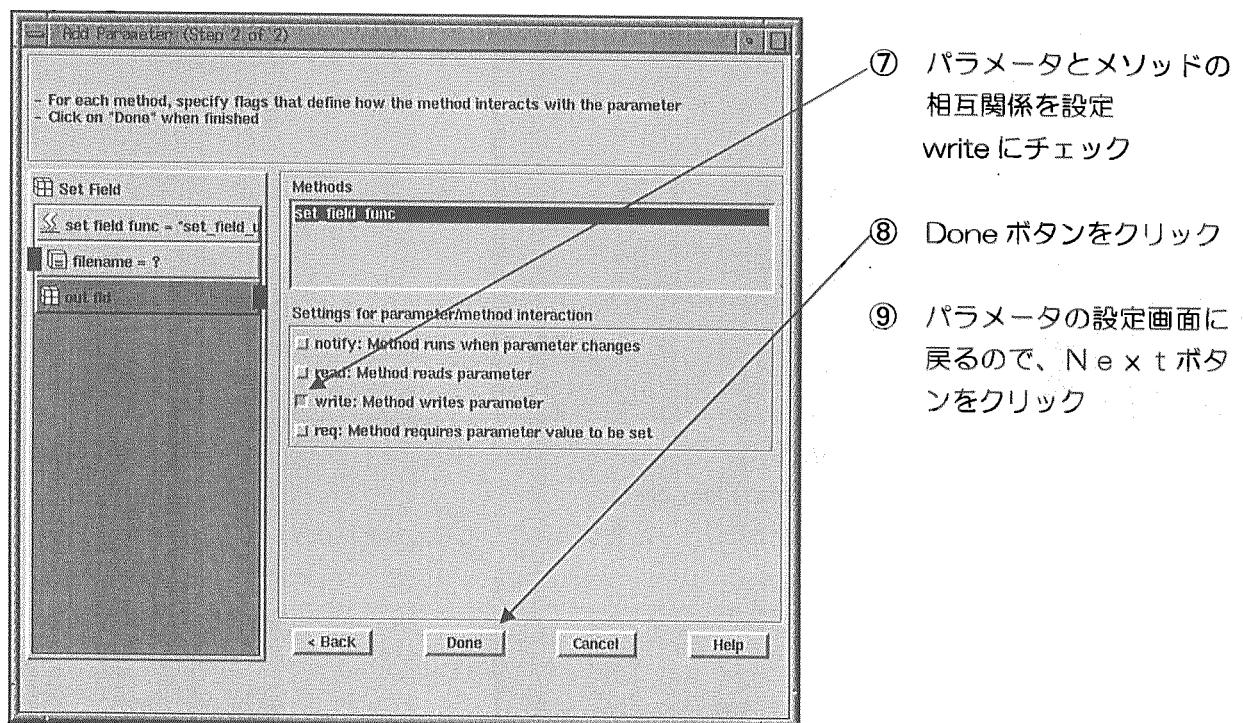


図 5-40 Add Parameter (Step 2 of 2)

Add Module Tool(ステップ 3)

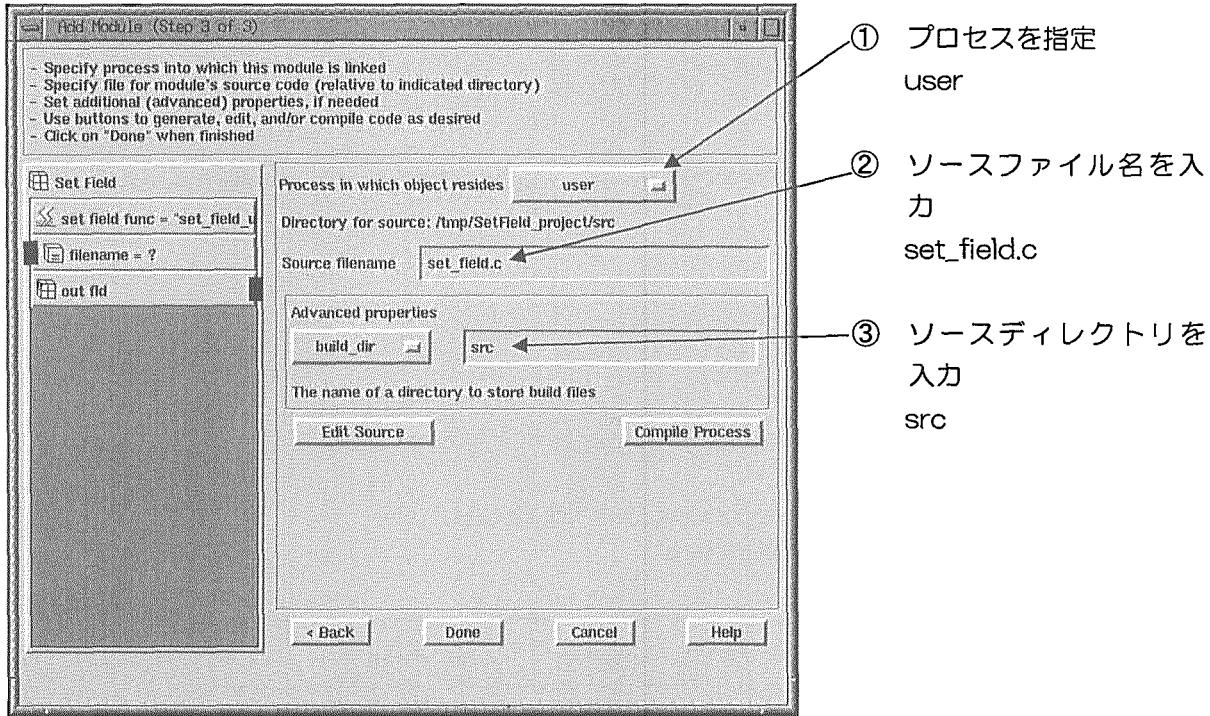


図 5-41 Add Module (Step 3 of 3)

④ Edit Source ボタンをクリック

ソースファイル(set_field.c)がないため、テンプレートを作成する旨を伝えるメッセージが表示されます。Yes ボタンをクリックすると、ソースファイルを編集するウインドウが表示されるので、set_field.c (図 5-42) を参考に編集してください。

⑤ Compile Process ボタンをクリック

コンパイルメッセージが表示されます。エラーが表示されなければコンパイルは終了です。リターンキーを押してウインドウを閉じてください。

ソースファイル：set_field.c

```
#include "user.h"

int
set_field_update(OMobj_id Set_Field_id, OMevent_mask event_mask, int seq_num)
{
    /*****
    /* Declare variables */
    *****/
    char *filename = NULL;
    OMobj_id out_fld_id;
    int out_fld_ndim, *out_fld_dims, out_fld_nspace, out_fld_nnodes;
    float *out_fld_coord;
    int out_fld_ncomp, out_fld_comp_count, out_fld_veclen;
    int out_fld_data_type, out_fld_ntdata;

    /* (1) char *out_fld_data */
    float *out_fld_data; ←
    /* (2) */
    int i;
    static float out_data[6]={10,20,30,40,50,60};
    static float coords[18]={-1,-1,0,0,-1,0,1,-1,0,-1,0,0,0,0,0,1,0,0};

    /*****
    /* Get input values */
    *****/
    /* Get filename's value */
    if (OMget_name_str_val(Set_Field_id,
                           OMstr_to_name("filename"), &filename, 0) != 1)
        filename = NULL;

    /*****
    /* Function's Body */
    *****/
    ERRerror("",ERR_NO_HEADER | ERR_INFO,"filename = %s\n",filename);

    /*****
    /* Set output values */
    *****/
    /* Set out_fld structured mesh */

    /* Get mesh id */
    out_fld_id = OMfind_subobj(Set_Field_id, OMstr_to_name("out_fld"), OM_OBJ_RW);
```

①float のポインター宣言に変更

②頂点データ、座標データの設定

入手したファイル名をメッセージとして出力

図 5-42 set_field.c

5.5.5 モジュールの保存

Workspace_1 にある Set_Field モジュールを選択します。

上部メニューから [Object]-[Save Objects] を選択し、ファイル名を指定します。

モジュールの保存先に制限はありませんが、ここではプロジェクト環境配下にある v ディレクトリの下に保存します。

<プロジェクト・ディレクトリ>/v/set_field.v

5.5.6 プロジェクト環境への登録

次回、AVS/Express を起動したときに、作成した Set_Field モジュールが含まれた状態で起動されるように設定します。

v/templ.v ファイルを以下のように編集します。

```
"$XP_PATH<0>/v/templ.v" Templates {
    WORKSPACE_1 {
        $include set_field.v
    };
};
```

Set_Field モジュールを保存する際にモジュール単体の保存 ([Object]-[Save Object]) ではなく、[Project]-[Save] を選択してプロジェクト全体を保存すると、templ.v ファイルに今の状態が保存されます。

\$include は、V ファイルを読み込むコマンドです。複数のプロジェクト環境をマージする場合や自作モジュールの数が多い場合に利用するとモジュール修正などの作業が行いやすくなります。

次回から Set_Field モジュールを含んだ AVS/Express を起動する場合には、以下のように行います。

```
% xp -project <プロジェクト環境>
```

AVS/Express Developer Edition

Version: 5.1 fcs sg6

Project: /tmp/SetField_project ← プロジェクト環境が表示されていることを確認してください。

5.5.7 実行テスト

Set_Field モジュールをインスタンスし、図 5-43 のネットワークを作成します。

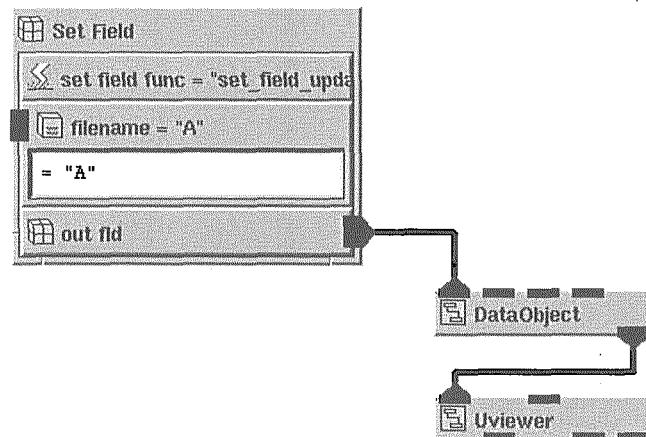


図 5-43 Set_Field モジュールの実行例

DataObject モジュールは、Accessories.Graphics.Modules 配下にあります。

レンダリングを行うためのモジュールで、フィールドデータからレンダリング用のデータを作成します。

Set_Field モジュールは filename の属性に req を付けているため、filename に値が入力されないと実行されません。Set_Field を開き、filename に "A" と入力してください。

5.5.8 UI モジュールとの組み合わせ

Set_Field モジュールはソースファイル内に実データを記述していますが、filename の受け渡しによってデータを外部から読み込むことが可能になります。

その場合は、図 5-44 のように UI モジュールを組み合わせます。

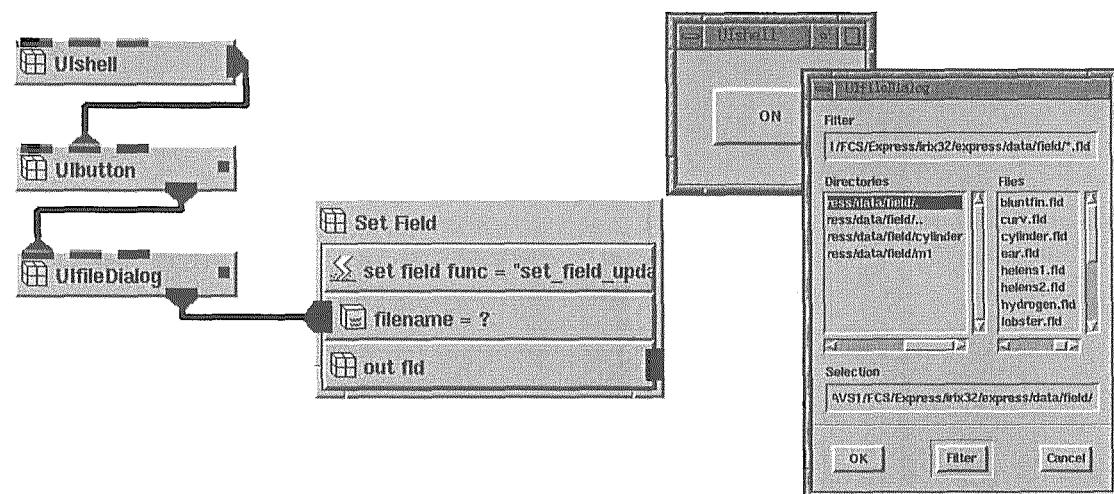


図 5-44 外部ファイルの読み込み

UIFileDialog モジュールは、User_Interface.Dialogs 配下にあります。
UIFileDialog.visible に 1 を入力するとファイルブラウザが表示されます。
UIFileDialog.searchPattern に “*.fld” と入力すると、拡張子が fld のファイルのみを表示させることができます。また、ファイルブラウザが表示されたときに現れるデフォルト・ディレクトリを指定することも可能です。

5.5.9 マクロ化

AVS/Express で提供されているモジュールは、図 5-45 のようにフィールドデータとレンダリングデータの二種類のポートを持ったマクロモジュールです。

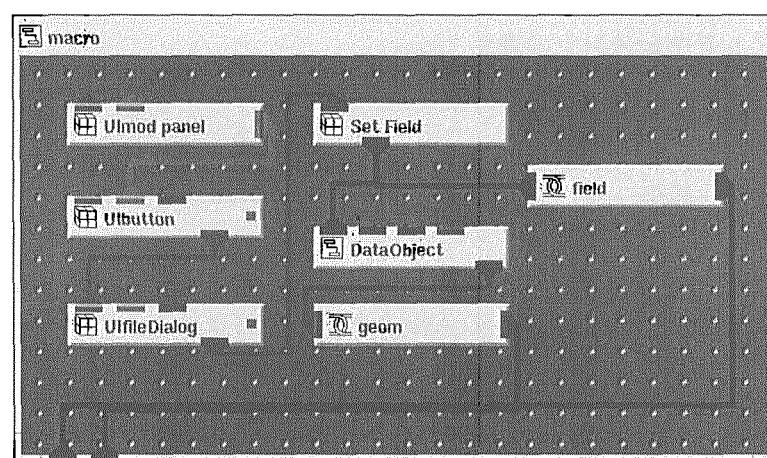


図 5-45 モジュールのマクロ化

6. AVS/Express のヘルプ機能

この章では、AVS/Express にあるオンライン・ドキュメントの表示方法とデモ・アプリケーションの起動方法について説明します。

6.1 オンライン・ドキュメントの表示方法

AVS/Express にはオンライン・ドキュメントを表示する方法として、次に挙げる 3 通りがあります。

1. モジュール情報の表示
2. モジュール毎のオンライン・ドキュメントの表示
3. 全オンライン・ドキュメントの表示

6.1.1 モジュール情報の表示

表示したいモジュール上でマウス右ボタンを押したままにします。プルダウンメニューが現れるので [Info…] を選択すると、モジュールの入出力ポートやパラメータのデータタイプなどの情報が表示されます。(図 6-1)

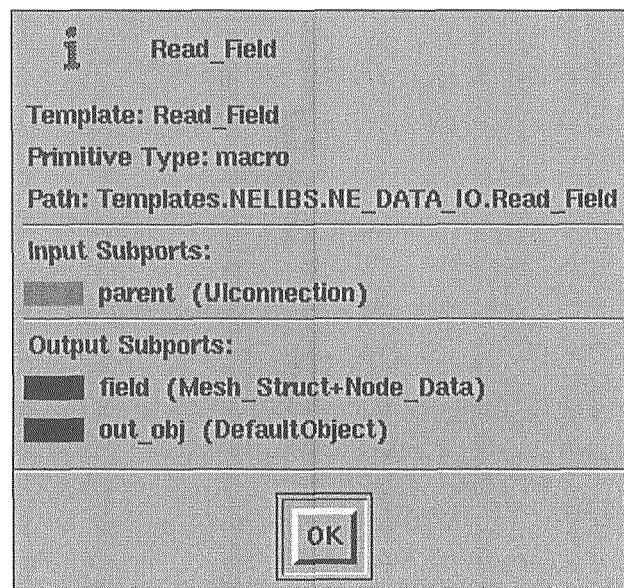


図 6-1 Object Info ウィンドウ

6.1.2 モジュール毎のオンライン・ドキュメントの表示

表示したいモジュール上でマウス右ボタンを押したままにします。プルダウンメニューが現れるので [Help] を選択します。

モジュールの機能や使い方などの詳細な情報が表示されます。(図 6-2)

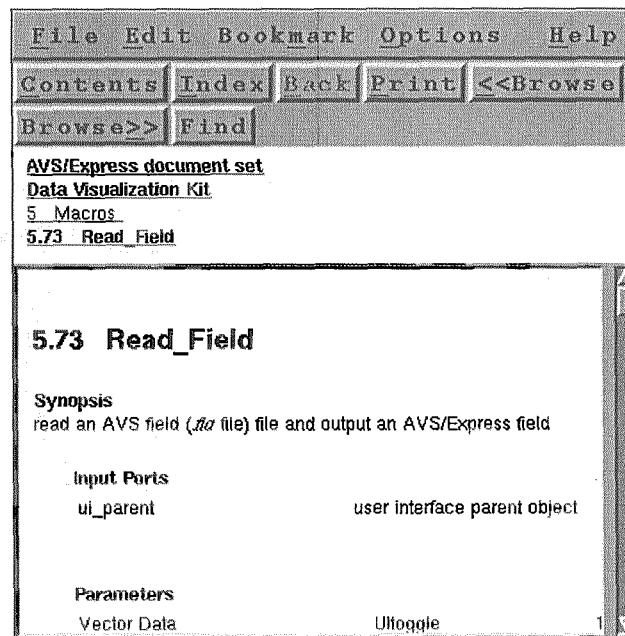


図 6-2 モジュール毎のオンライン・ドキュメント

6.1.3 全オンライン・ドキュメントの表示

ネットワーク・エディタのプルダウンメニューから [Help]-[Contents] を選択します。図 6-3 のようなマニュアル一覧が表示されるので、読みたいドキュメントを選択してください。

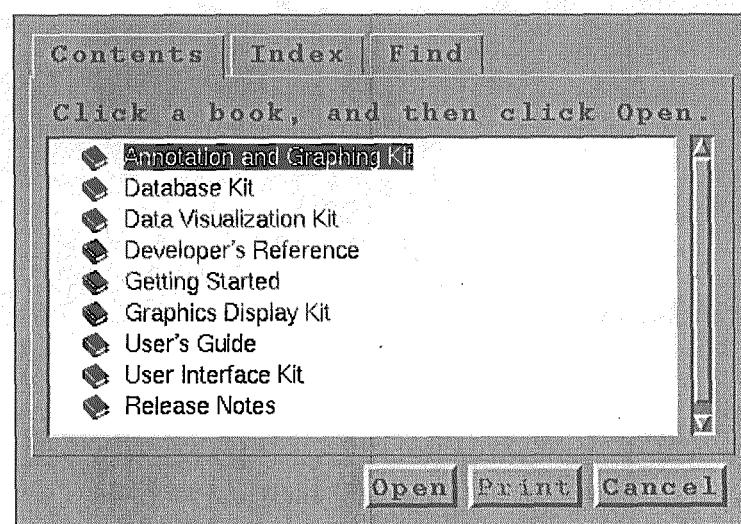


図 6-3 Contents 一覧

6.2 デモ・アプリケーションの起動方法

すでにアプリケーションが読み込まれている場合は、アプリケーションを消去します。
ネットワーク・エディタのプルダウンメニューから [File]-[Delete Application] を選択します。

[Libraries]選択メニューから [Examples]を選択します。
Visualization カテゴリから Advect を選択し、ワークスペースに移動します。(図 6-4) advector モジュールを用いた可視化アプリケーションが起動されます。(図 6-5)

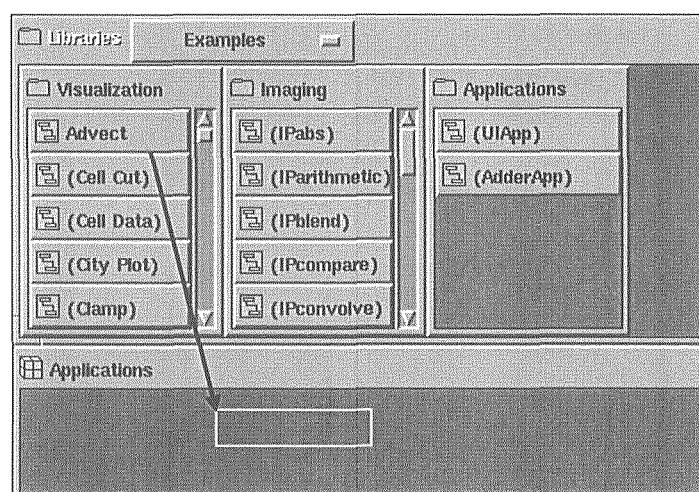


図 6-4 Advect アプリケーションのインスタンス

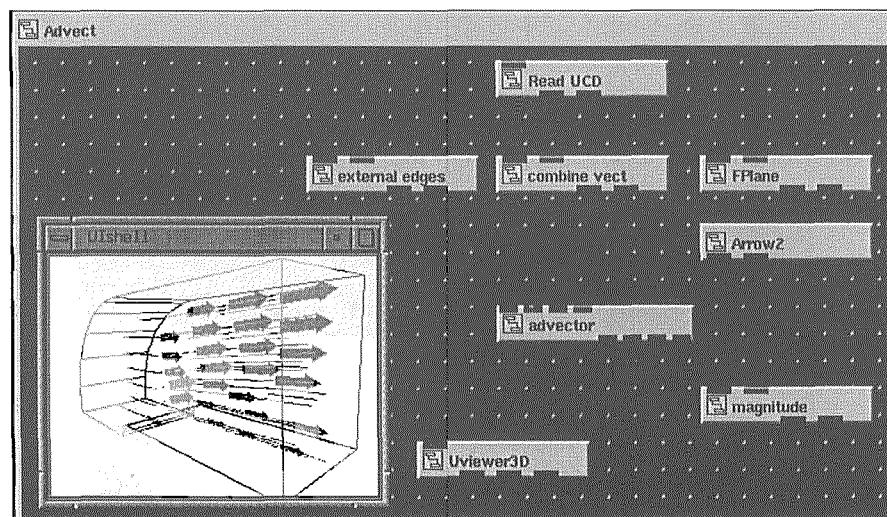
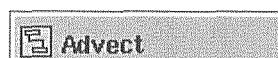


図 6-5 Advect アプリケーションのネットワークと可視化例

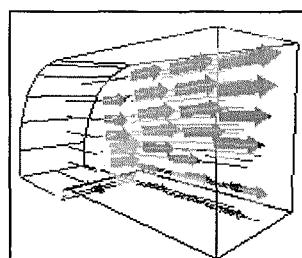
Examples ライブラリに提供されているデモはアプリケーションとして保存されています。そのため、デモ・アプリケーションを起動する場合は、SingleWindowApp ワークスペースや MultiWindowApp ワークスペースは削除しておきます。

デモ・アプリケーション一覧

■ Visualization カテゴリ



Advect
流速ベクトルデータを基に計算を行ない、フィールドデータ空間にパーティクルを飛ばす。

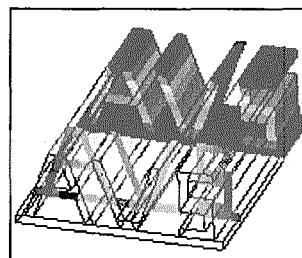


[advector]

Run: オンにするとパーティクル位置が時間発展する
Step: 発展時間の刻み幅



Cell Cut
セルデータを任意断面で切断する。標準で提供されている 2 種類のカット方法を重ねて表示している。

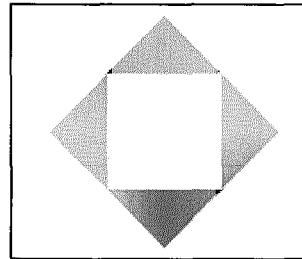


[cut_plane]

plane distance: スライス面の位置
Above: カット面に対する消去方向
[slice_plane]
plane distance: スライス面の位置



Cell Data
セルごとに複数のデータをトグルスイッチによって表示非/表示にする。



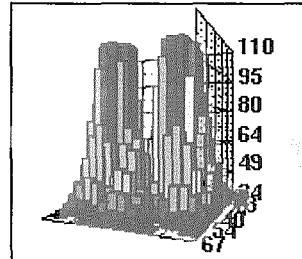
[select_cells]

cell sets: 0~3 Tri, 4 Quad. (それぞれをオン/オフするとセルが表示/非表示する)

[extract_cell_component]
cell data: セルデータ



City Plot
2 次元フィールドデータから四角柱の棒グラフを作成する。

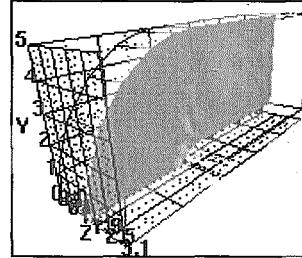


[city_plot]

height scale: 棒グラフの高さ方向のスケール
x-scale, y-scale: 四角柱の x, y 方向のスケール
(0.8 に設定すると他の棒グラフと接しない)

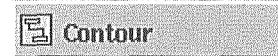


Clamp
データの値域を指定範囲に収める。

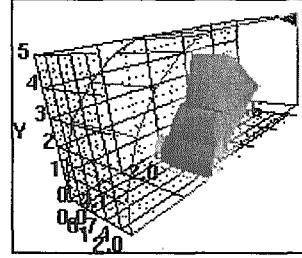


[clamp]

data component: clamp 处理の対象データ
Clamp below min value: オンの場合、min value 以下のデータを全て min value に設定
Clamp below max value: オンの場合、max value 以上のデータを全て max value に設定



Contour
等価ボリュームとカラー コンター表示を行なう。

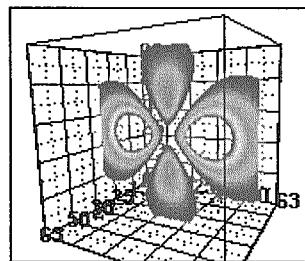


[contour]

min level: ボリュームの最小値
max level: ボリュームの最大値

Contour 2D Unif

2次元ユニフォームデータに対して指定値域でカラー・コンター表示を行なう。

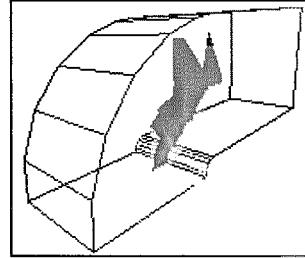


[contour]

min level: コンターの最小値
max level: コンターの最大値

Contour 2D

2次元メッシュデータに対して指定値域でカラー・コンター表示を行なう。

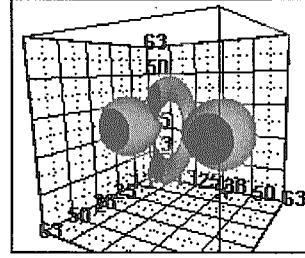


[contour]

min level: コンターの最小値
max level: コンターの最大値

Crop

データ領域の一部を切り取り、等値面表示を行なう。



[crop]

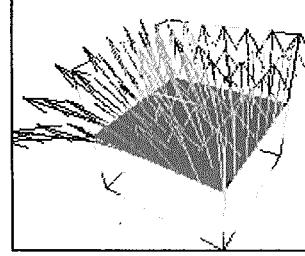
I~K min: 抽出する値域の最小値
I~K max: 抽出する値域の最大値

[isosurface]

iso level: 等値面を張る値

Curl

ベクトルフィールドの回転を計算して、矢印表示を行なう。

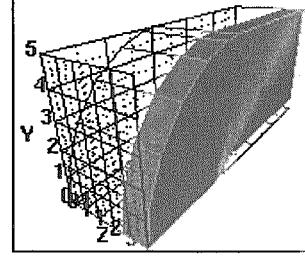


[Plane]

x-dimension: x 方向のメッシュの数
y-dimension: y 方向のメッシュの数
Transformation Editor: スライス面の位置を指定するためのエディタの表示

Cut

3次元ユニフォーム型のデータをソリッド状態でスライスし、テクスチャーマッピングを行なう。

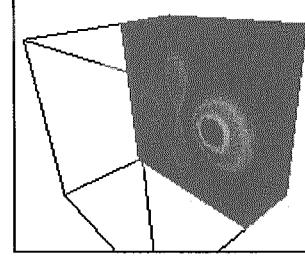


[cut]

Above: カットする向き
plane distance: カットする位置

Cut Texture

3次元ユニフォーム型のデータをソリッド状態でスライスし、テクスチャーマッピングを行なう。

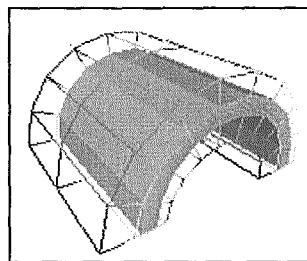


[cut_texture3D]

Above: カットする向き
plane distance: x 方向に対するスライス位置

 **Cylinder**

円筒座標系の表示を行なう。



[contour]

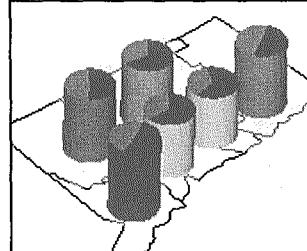
min level: コンターの最小値
max level: コンターの最大値

※座標の指定は cyl_field オブジェクト内の points で行なう。

 **Cylinder Plot**

円柱の立体棒グラフを作成する。

※AVS/Express viz で実行するとエラーが発生する。棒グラフは表示されるが、地図データが表示されない。

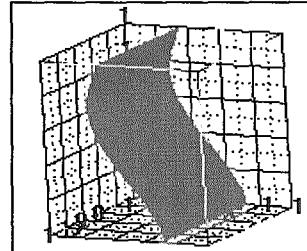


[cylinder_plot_unif]

vertical segmentation: グラフの描画方法
scale height by total: オンの場合、円柱の高さを合計値にする。
scale radius by total: オンの場合、円柱の半径を合計値にする
color sides by total: オンの場合、円柱の側面の色を合計値に対応させる

 **Div**

ベクトルフィールドの発散を計算し、等値面表示を行なう。

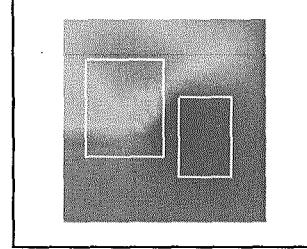


[isosurface]

iso level: 発散値

 **Display Vol ROI**

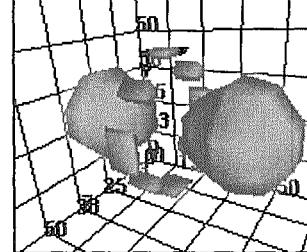
3次元ユニフォーム型のデータをスライスし、指定した矩形領域とカラー コンター表示を行なう。



※矩形領域指定はマウス右ボタンで行なう。

 **Downsize**

フィールドデータに対してデータの間引きを行なう。

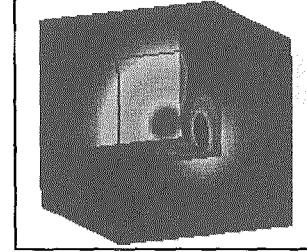


[downsize]

I~K downsize factor: データを間引く割合

 **Excavate Brick**

3次元ユニフォームデータを指定された領域で切り取り、断面をカラーコンターで表示する。

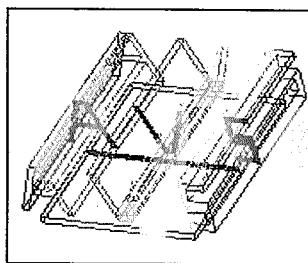


[excavate_brick3D]

X~Z: 切り取る領域
Draw Sides: オンの場合は、側面もカラーコンター表示する

Explode Fields

データの存在範囲の中心から各軸方向に爆発させるかのように幾何変換させる。

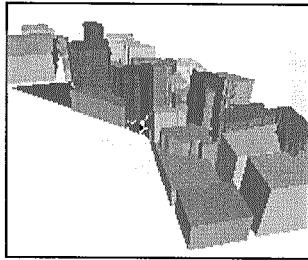


[**explode_fields**]
x~z-scale: 各軸方向の移動倍率

Extrude Cells

セルデータに基づいてセルオブジェクトを高さ方向に拡大表示する。

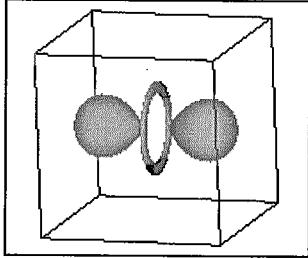
※AVS/Express viz では実行できません。



[**extrude_cells**]
height scale: 高さ方向のスケール
shrink factor: セルオブジェクトのスケール
draw skirts: オンの場合、セルオブジェクトの側面を表示する
color skirts: オンの場合、セルオブジェクトの側面をカラー表示する

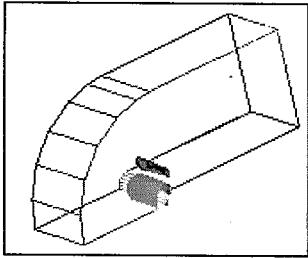
File Field Unif

ファイルアクセスモジュールを使用してフィールドデータを読み込み、ユニフォーム型メッシュのフィールドデータを作成する。



File Field UCD

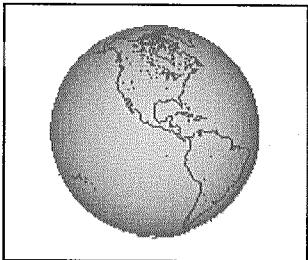
ファイルアクセスモジュールを使用してUCDデータを読み込み、メッシュデータを作成する。



[**GISWorldDataBank**]
Western Long, Eastern Long,
Southern Lat, Northern Lat: 地図を描く範囲

GISMapDisplay

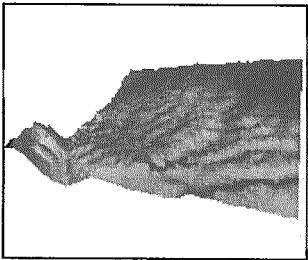
地球儀を表示する。



GISDEMQuad

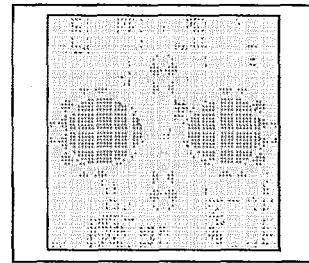
US Geologic Survey (USGS)
Digital elevation Model (DEM) ファイルを読み込み、表示する。

※XP_ROOT の設定が必要。



 **Geo Glyph**

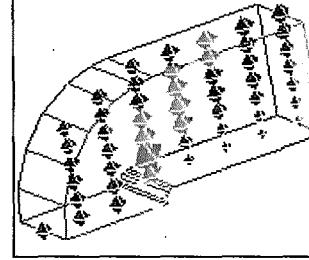
ノードごとのデータ値に応じてオブジェクトの大きさを変え、グリフを表示する。



[orthoslice]
axis: 断面方向
plane: 断面位置

 **Glyph Interp**

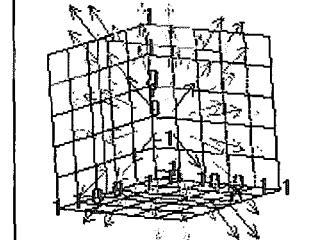
任意断面でのデータ値に応じてオブジェクトの大きさを変え、グリフを表示する。



[FPlane]
x, y-dimension: 作成する断面の格子数
Transformation Editor: 断面位置を指定するためのエディタの表示

 **Grad**

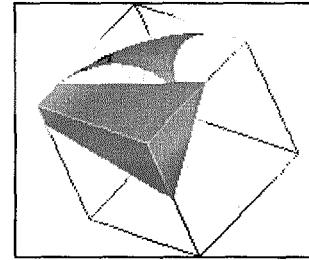
ベクトル勾配を計算し、勾配量を矢印表示する。



[FPlane]
x, y-dimension: 作成する断面の格子数
Transformation Editor: 断面位置を指定するためのエディタの表示

 **Hex2**

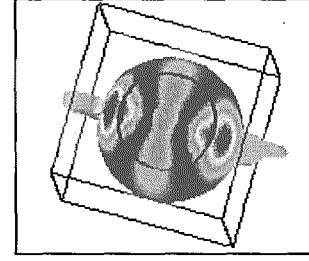
中間節点を持つ Hexahedron タイプのセルデータをファイルから読み込み、表示する。



[isosurface]
iso level: 等値面表示を行なう値
[cut_plane]
Above: 切り取る方向
plane distance: 断面位置を指定するためのエディタの表示

 **Interp Texture3D**

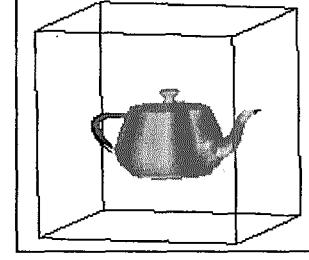
計算空間に任意のメッシュ(デモではティーポット)オブジェクトを配置し、メッシュにカラーコンターイメージをテクスチャーマッピングする。



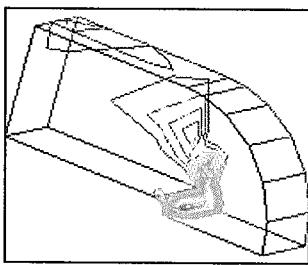
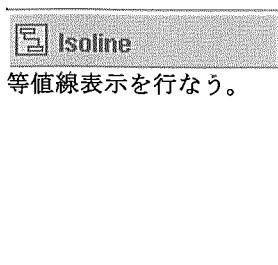
ティーポットオブジェクトを移動させるとテクスチャーアイメージが変化する。

 **Interp Unif**

計算空間に任意のメッシュオブジェクトを配置し、データを取り出す。取り出したデータに対してカラーコンター表示を行なう。

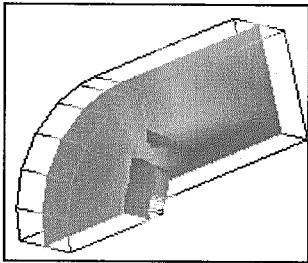
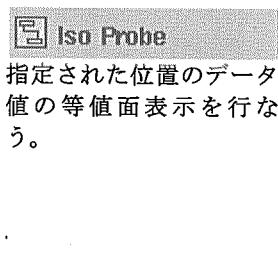


ティーポットオブジェクトを移動させるとテクスチャーアイメージが変化する。



[isoline]

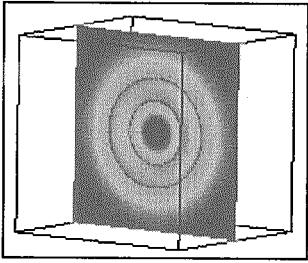
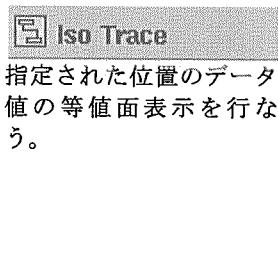
iso component: 等值線表示を行なうデータ
number of contours: 指定された値域の等值線の本数
min level: 値域の最小値
max level: 値域の最大値



[iso_probe]

iso component: 等值面表示を行なうデータ
map component: 等值面のカラーコンター表示を行なうデータ
value: 指定された位置の値

※位置の指定は Ctrl+マウス左ボタンクリックで行なう。

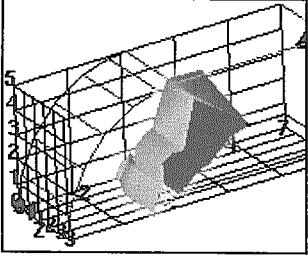
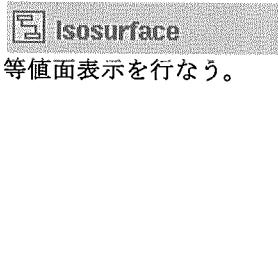


[orthoslice]

axis: スライス方向
plane: スライス位置
[isosurface_trace]

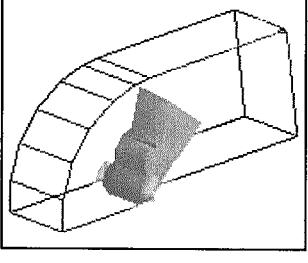
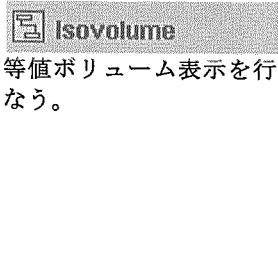
iso level: 指定された位置の値

※位置の指定は断面のみが表示された画面上で Ctrl+マウス左ボタンクリックにより行なう。



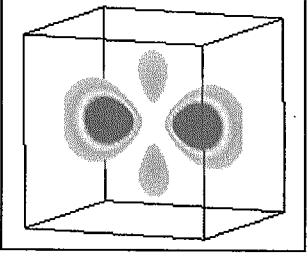
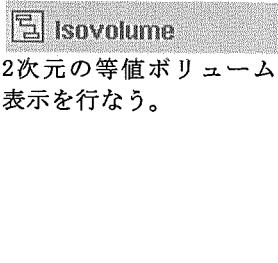
[isosurface]

iso component: 等值面表示を行なうデータ
iso level: 等值面表示を行なう値
map component: 等值面のカラーコンター表示を行なうデータ



[isovolumne]

Above: オンの場合、iso level より大きい値に対して等值ボリューム表示を行なう
iso component: 等值ボリューム表示を行なうデータ
iso level: 等值ボリューム表示を行なう値
map component: 等值ボリュームのカラーコンター表示を行なうデータ

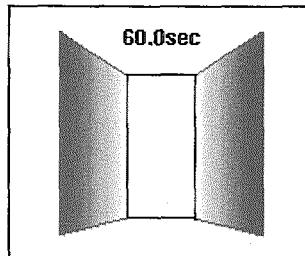


[isovolumne]

Above: オンの場合、iso level より大きい値に対して等值ボリューム表示を行なう
iso level: 等值ボリューム表示を行なう値

[Multistep Field]

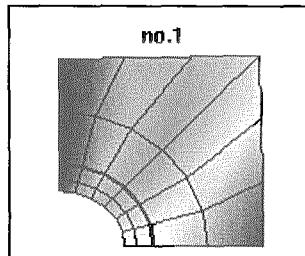
時系列フィールドデータを読み込み、表示する。

**[Read_Field]**

Step Forward: 1ステップ進めたデータを出力
Step Backward: 1ステップ戻したデータを出力
One-time: 現在のステップから最終ステップまで連続してデータを出力

[Multistep UCD]

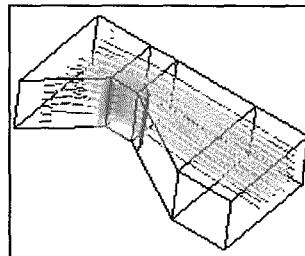
時系列UCDデータを読み込み、表示する。

**[Read_Field]**

Step Forward: 1ステップ進めたデータを出力
Step Backward: 1ステップ戻したデータを出力
One-time: 現在のステップから最終ステップまで連続してデータを出力

[Multi Block]

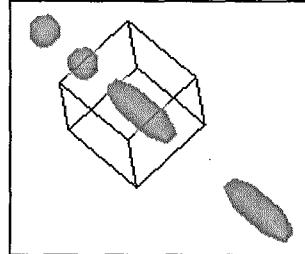
計算領域を複数のブロックに分割したマルチブロックデータを読み込み、表示する。

**[Fplane]**

x, y-dimension: 作成する断面の格子数
[isosurface_ARR]
iso component: 等値ボリューム表示を行なうデータ
iso level: 等値ボリューム表示を行なう値
map component: 等値ボリュームのカラーコンター表示を行なうデータ

[Mirror Scale]

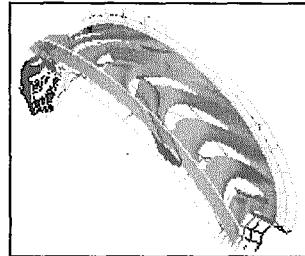
データ(オブジェクト)を軸対称に反転させ、更に拡大表示する。

**[mirror]**

mirror X~Z axis: 反転対称の軸
[scale]
X~Z-scale: 軸ごとのスケール

[Null Data]

障害物部分にヌルフラグを立て、可視化時に非表示にする。

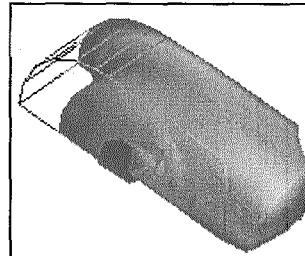
**[set_null]**

Set Null: ヌルデータの表示/非表示の切り替え
Null Value: ヌルフラグを立てる値

※AVS/Express Viz では利用できないモジュールが一部含まれているため実行時にエラーが表示されます。しかし、ヌルデータによる非表示機能は確認することができます。

[Offset]

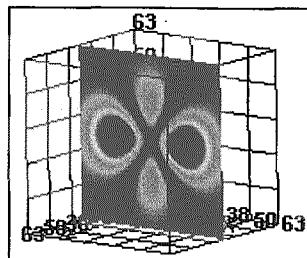
各点が持つベクトルデータを基にメッシュを変形させる。カラーコンターはベクトルの大きさを表している。

**[offset]**

scale: 変形のスケール

[Orthoslice]

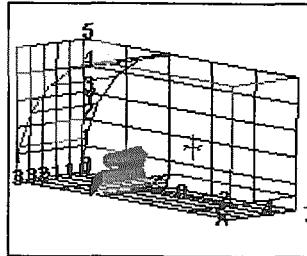
3次元フィールドデータを軸方向にスライスし、その断面をコンター表示する。

**[orthoslice]**

axis: スライスする軸方向
plane: スライスする位置

[Probe]

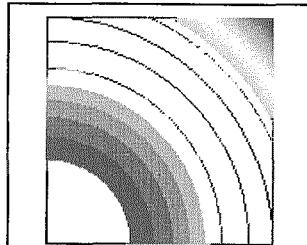
probe オブジェクトを移動させることにより、任意の場所のデータを得る。

**[probe]**

data component: probe により情報を得たいデータの選択
probe scale: probe のスケール

[Quad2]

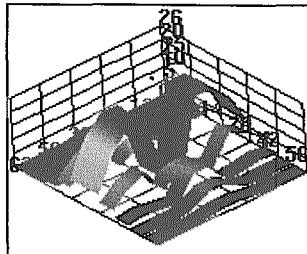
中間節点を持つ quad タイプのデータをファイルから読み込み、等値線表示、カラーコンター表示、領域分けされたカラーコンター表示を行なう。

**[isoline]**

number of contours: ラインの数
[solid_contour]
number of contours: コンター領域の分割数
[isoline][contour][solid_contour]共通
min level: コンターの最小値
max level: コンターの最大値

[Ribbon Plot]

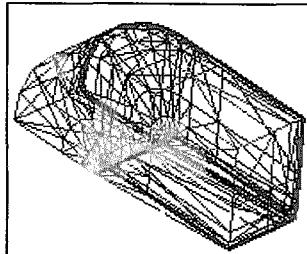
リボン形状による立体グラフを表示する。

**[downsize]**

I~K downsize factor: データを間引く割合
[ribbon_plot]
ribbon direction: リボンを表示する方向
scale: リボンの高さ方向のスケール

[Scat to Tet]

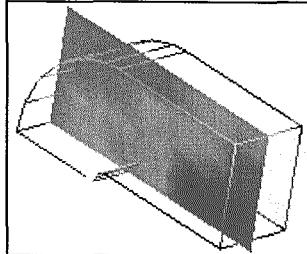
離散データから三角形または四面体の非構造格子データを生成する。

**[scat_to_tri]**

2D(triangles): オンの場合、三角メッシュを生成する。

[Scat to Unif]

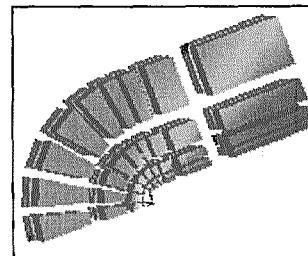
離散データからユニフォーム型メッシュデータを生成する。

**[scat_to_unif]**

I~K-dimension: 作成するメッシュの大きさ
interpolate range I~K: 植間計算するために得るメッシュデータの大きさ
order: 植間の精度

Shrink Cells

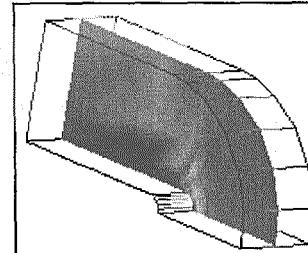
セルの各頂点をセルの重心方向に縮小させ、オブジェクトを構成する各セルを理解しやすく表示する。



[shrink_cells]
scale factor: 縮小率

Slice

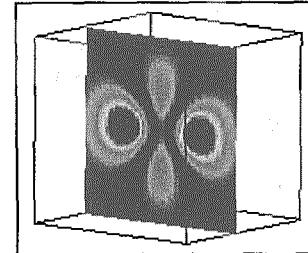
任意断面でスライスし、コンター表示する。



[slice]
plane distance: スライス断面の位置

Slice Texture3D

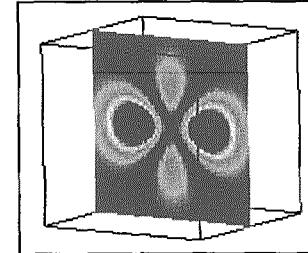
任意断面でスライスし、コンター表示する。



slice_texture3D オブジェクトを選択し、マウスで移動すると、任意断面のコンター図をリアルタイムに見ることができる。

Slice Texture2D

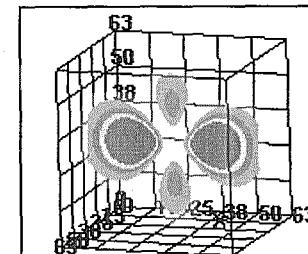
任意断面でスライスし、コンター表示する。



slice_texture2D オブジェクトを選択し、マウスで移動すると、任意断面のコンター図をリアルタイムに見ることができる。

Solid Contour

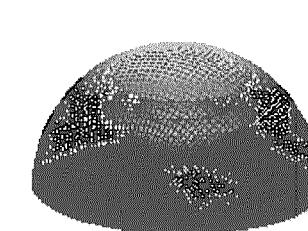
領域分けされたカラーコンター表示を行なう。



[orthoslice]
axis: 軸方向
plane: スライス断面の位置
[solid_contour]
number of contours: 領域の分割数
min level: コンターの最小値
max level: コンターの最大値

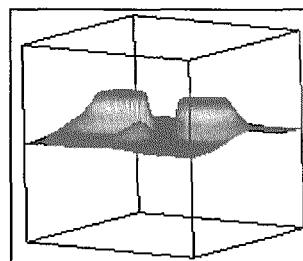
Sphere

入力フィールドを球座標へ変換する。



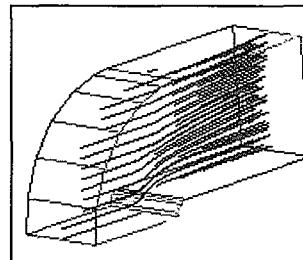
[isosurface]
iso level: 等値面表示を行なうための値

 **Surf plot**
データに基づき、高さ方向にメッシュを変形させる。



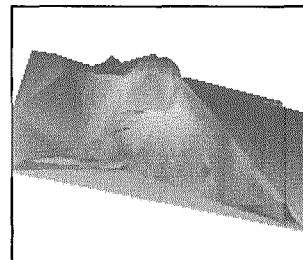
[surf_plot]
scale: 高さ方向のスケール

 **Stream**
ベクトルデータを流線表示する。



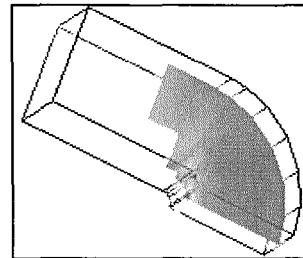
[FPlane]
x-dimension: 流線の始点面の x 方向の格子数
y-dimension: 流線の始点面の y 方向の格子数
Transformation Editor: 流線の始点位置

 **Surface Optimize**
非常に多いポリゴンを少ないトライアングルメッシュで最適化する。



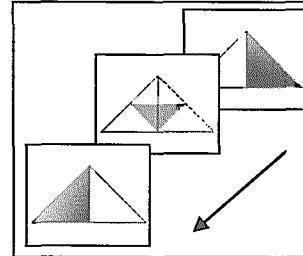
[downsize]
I downsize factor: I 方向のデータを間引く割合
J downsize factor: J 方向のデータを間引く割合
[surf_optimize]
tolerance: トライアングルメッシュの大きさ

 **Threshold**
指定された範囲より外れたデータを NULL データに置き換えて可視化を行なう。



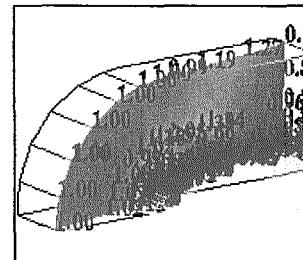
[threshold]
Below min value: オンの場合、min value 未満の値が NULL
min value: 指定値域の最小値
Above max value: オンの場合、max value を超える値が NULL
max value: 指定値域の最大値

 **Time Data**
100ステップ目と200ステップ目のデータ間を補間し、100～200ステップ間で任意ステップのデータを可視化する。



[Loop]
Run: アニメーション開始
Step: Increment で指定された分カウントを加算する
Cycle Options: アニメーションの繰り返し設定

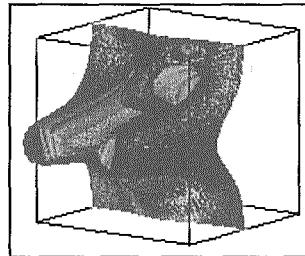
 **TextGlyph**
ノードごとにデータをラベル表示する。



[slice_plane]
plane distance: スライス面の位置

Texture Mesh

メッシュデータにイメージを貼り付ける(テクスチャーマッピング)。

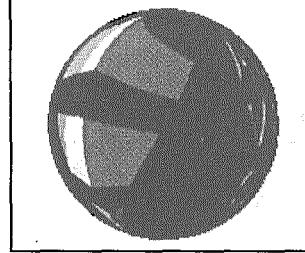


[texture_mesh]

switch_uv: イメージの縦横反転
flip_u: u 方向に対してイメージを反転
flip_v: v 方向に対してイメージを反転
u, v scale: イメージの縦横のスケール
u, v shift: イメージの平行移動

Texture Sphere

球状のメッシュにテクスチャーマッピングを行なう。

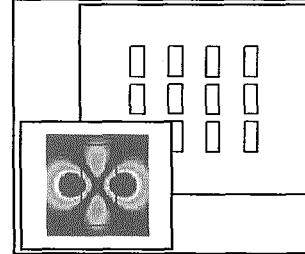


[texture_sphere]

Eastern, Western Long: テクスチャーマッピングの開始位置(東西位置)
Southern, Northern Lat: テクスチャーマッピングの終了位置(南北位置)

Tile ROI

3次元ユニフォーム型のデータをスライスし、切り出した矩形領域をタイルのように敷き詰める。



[copy_ROI]

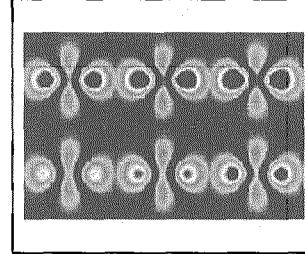
Start: コピーするスライス面の開始位置
Stop: コピーするスライス面の終了位置
Run: コピー開始

[Tile_ROIs]

Start: スライス開始位置
Width, Height: タイルの幅と高さ
Border Width: タイル間の隙間
Orientation: スライス面の整列順番の指定

TileVolume

3次元ユニフォーム型のデータをスライスし、タイルのように敷き詰めてカラーコンター表示を行なう。

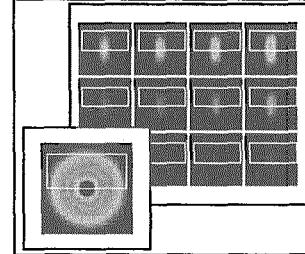


[Tile_Volume]

Start: スライス開始位置
Width, Height: タイルの幅と高さ
Border Width: タイル間の隙間
Orientation: スライス面の整列順番の指定

TileVolumeROI

3次元ユニフォーム型のデータをスライスし、切り出した矩形領域と全体のカラーコンター図をタイルのように敷き詰める。



[copy_ROI]

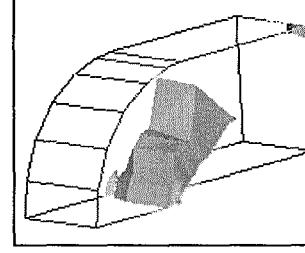
Start: コピーするスライス面の開始位置
Stop: コピーするスライス面の終了位置
Run: コピー開始

[Tile_Volume_ROIs]

Start: スライス開始位置
Width, Height: タイルの幅と高さ
Border Width: タイル間の隙間
Orientation: スライス面の整列順番の指定

Volume Integral

ボリュームデータの体積を計算する。



[isovolume]

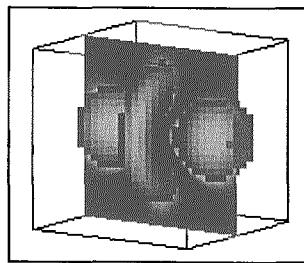
iso level: 等値面を張る値

[volume_integr]

volume: 等値面内の体積量

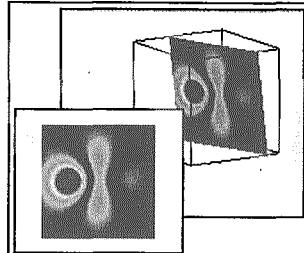
volume integral: 体積積分

 **Volume Render**
ボリュームレンダリングを行なう。



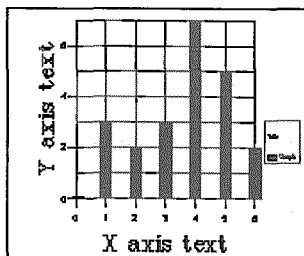
[volume_render]
Minimum Alpha: 最小値に対する透明度(0.03程度)
Maximum Alpha: Range Position の値に対する透明度(0.18程度)

 **Xform Slice**
スライス断面の幾何変換情報を入れ替える。



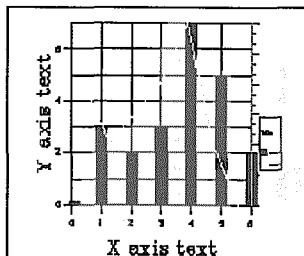
■Graph Macros カテゴリ

 **XYGraph**
XYデータに対するグラフを作成する。データはfloatオブジェクトで与えている。

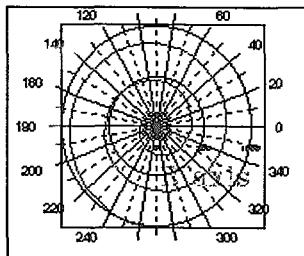


[Graph]
graphtype: グラフのタイプ
line color: グラフの色

 **Two XYGraphs**
同一ビューワ内に2つの座標系を持ったグラフを作成する。



 **PolarGraph**
極座標グラフを作成する。

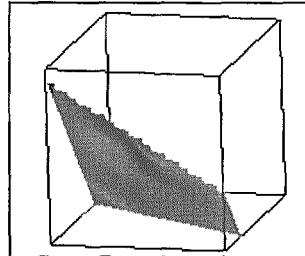


[PolarGraph]
graphtype: 極座標グラフのタイプ
line style: ラインのスタイル
line color: ラインの色

■Viz Macros カテゴリ

 Arbitrary Slicer

3次元データの任意断面の表示を行なう。

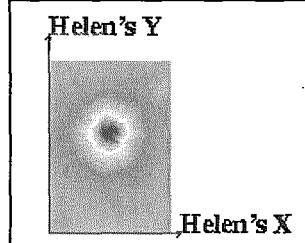


[Arbitrary_Slicer]

Plane Transform Editor: スライス面の位置を指定するためのエディタを表示
plane distance: スライス面の位置

 Axis2D

2次元の座標軸を作成する。

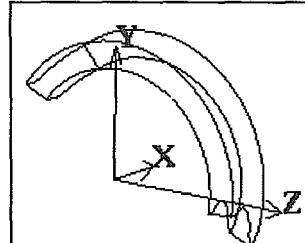


[Basic_Axis2D]

Axis title: 各軸のタイトル
Axis scale: 各軸のスケール

 Axis3D

3次元の座標軸を作成する。

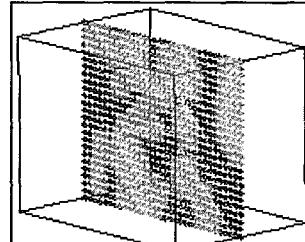


[Basic_Axis3D]

Axis title: 各軸のタイトル
Axis scale: 各軸のスケール

 Bubbleviz

ノード点に球を表示する。サンプルでは3次元データをスライスし、断面の各ノードに球を作成している。

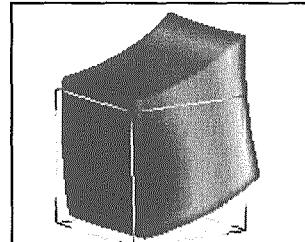


[Bubbleviz]

scale: 球のスケール
Subdivision: 球のポリゴン分割数(大きいほど滑らかな球)

 Offset

ベクトルデータに対して変形表示を行なう。

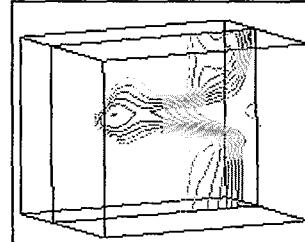


[Offset]

Factor: 変形対象の要素
Offset Scale: 変形のスケール

 Orthoslice

2次元または3次元データに対してスライス断面を抽出する。サンプルでは切り出された断面データを等値線表示している。



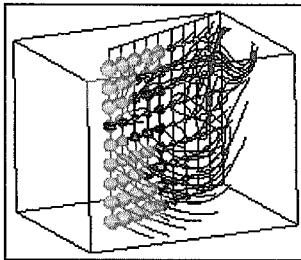
[Orthogonal_Slicer]

axis: スライス断面の軸
plane: スライス断面の位置

 **OutputImage**
ビューワ内のイメージデータをファイルに保存する。

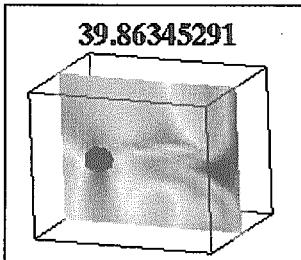
[OutputImage]
Image Filename: ファイル名
File Format: 保存できるファイルフォーマット
(av, image, GIF, JPEG, PBM, TIFF)
Write File: クリックするとファイルに書き込まれる
flip: イメージの反転

 **Particle_Advector**
ベクトルデータに対してパーティクル・トレースを行なう。



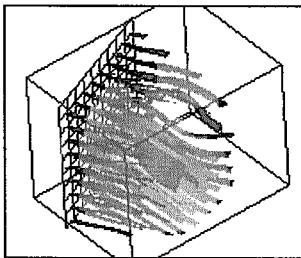
[Particle_Advector]
Probe on/off: トレース開始位置の表示/非表示
Probe parameter: トレース開始位置の形状
Glyph parameter: グリフ形状
Run: オンの場合、パーティクル・トレースが開始される

 **Probe**
指定された位置のデータを表示する。
位置の指定は Ctrl+マウス左ボタンで行なう。



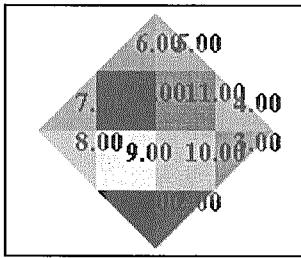
[Probe]
probe scale: クリック位置に表示するオブジェクトのスケール
value: クリックされた点のデータ

 **Streamline**
ベクトルデータに沿った流線表示を行なう。



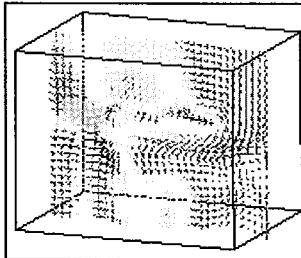
[Streamline]
Probe on/off: 流線の開始位置の表示/非表示
start point: 流線の開始位置の形状

 **Text_Glyph**
入力されたデータ各店にデータ値を表示する。



[Text_Glyph]
Display Data: 表示データ
Select Data Component: ラベル表示するデータ

 **Vector**
ベクトルデータに対して矢印ベクトル表示を行なう。



[Vector]
Glyph Type: 矢印の形状
scale: 矢印のスケール

7. AVS/ExpressMPE の操作手順

AVS/ExpressMPE (Multi-Pipe-Edition) は、AVS/Express を用いて作成した可視化結果を、立体視表示装置に出力するためのアプリケーションです。

AVS/ExpressMPE の利用は、画像サーバである Onyx コンソールにおいて行い、利用に際しては、情報システム管理課 利用支援係において予約を必要とします。

7.1 ハードウェアの準備

7.1.1 画像サーバ (Onyx3200) の設定確認

AVS/ExpressMPE を利用するには、画像サーバのキーボード設定が、“1key” になっている必要があります。

“1key” とは、2つのグラフィックス・パイプラインを専有する状態であり、2セットあるコンソール端末を1つのキーボード及びマウスで操作します。

通常運用中は、“2key” になっていますので、システム管理者に連絡し、“1key” の設定に変更します。

7.1.2 周辺ハードウェアの準備

(1) VR 制御用 PC のモニタ電源を投入します。PC 本体の電源は常時投入されていますので操作は不要です。(スイッチは、液晶ディスプレイ下側にあります。図 7-1 左)

(2) VR 制御用 PC のタッチパネル上で [3D] ボタンを選択し、[実行] ボタンを押します。

(図 7-1 右)

この操作により、タッチパネル右下に「処理中に付き、しばらくお待ちください」のメッセージが表示されます。数秒すると、画像サーバの画面解像度が切り替わり、プロジェクタの電源が投入されます。

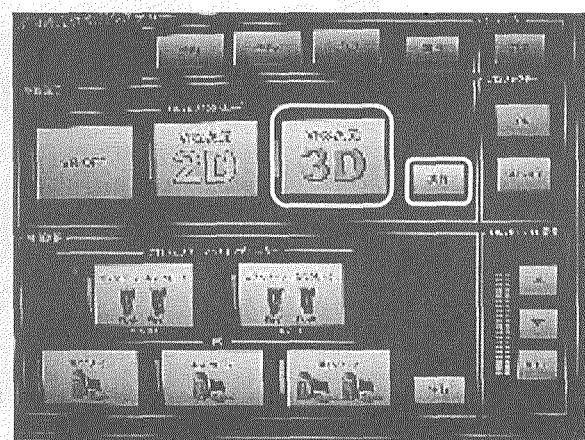
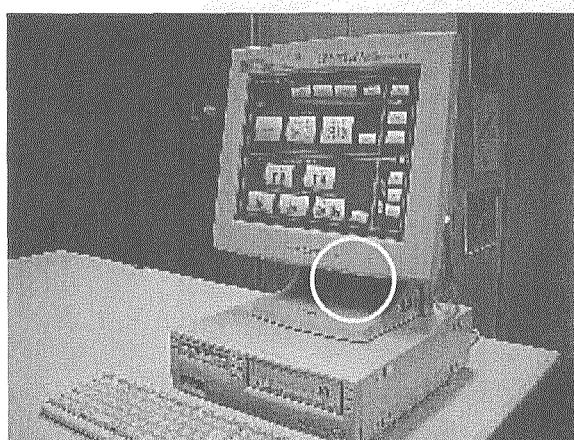


図 7-1 VR 制御用 PC と タッチパネル画面

- (3) 磁気センサー (Fastrack) の電源を投入します。(図 7-2 左)
 背面インジケータランプが緑色に点灯していることを確認してください。(図 7-2 右)

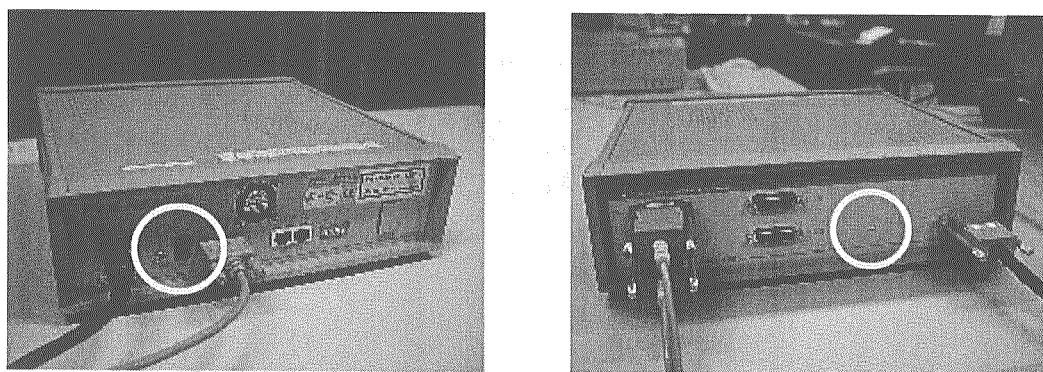


図 7-2 磁気センサー (Fastrack)

7.1.3 センシングデータ取得用デーモン (Trackd) の起動

- (1) 各人のユーザアカウントで、画像サーバにログインしてください。
 - (2) /home/vr-cnt にある AVS/ExpressMPE 用の Trackd を起動します。
`% /home/vr-cnt/Trackdvxp`
 - (3) Trackd を起動した Window に Trackd の Information (取得しているデータ) が表示されます。(図 7-3)
- Information 内の数値に変動がない場合は、一旦 Trackd を終了し再起動してください。
 Trackd を再起動する場合は、Fastrack の電源も入れ直してください。
 (Fastrack の電源再投入は、電源切断後 10 秒ほど待った後に行ってください)

```
端末 - wintern
A = 109.1
trackd[2060819]: [tra02] x= -2.43 y= 6.85 z= 4.26 azim=-133.5 elev= -67.7 roll=
= 172.4

trackd[2060819]: [con01] val01: 0.000 val02: 0.000
trackd[2060819]: but01: 0 but02: 0 but03: 0

trackd[2060819]: [tra01] x= -4.12 y= 3.03 z= 1.40 azim=-115.7 elev= -26.7 roll=
= 109.1
trackd[2060819]: [tra02] x= -2.49 y= 6.85 z= 4.27 azim=-132.7 elev= -67.7 roll=
= 171.6

trackd[2060819]: [con01] val01: 0.000 val02: 0.000
trackd[2060819]: but01: 0 but02: 0 but03: 0

trackd[2060819]: [tra01] x= -4.12 y= 3.03 z= 1.40 azim=-115.7 elev= -26.7 roll=
= 109.1
trackd[2060819]: [tra02] x= -2.49 y= 6.85 z= 4.27 azim=-133.2 elev= -67.7 roll=
= 172.1

trackd[2060819]: [con01] val01: 0.000 val02: 0.000
trackd[2060819]: but01: 0 but02: 0 but03: 0

trackd[2060819]: [tra01] x= -4.12 y= 3.03 z= 1.40 azim=-115.6 elev= -26.9 roll=
= 109.1
trackd[2060819]: [tra02] x= -2.49 y= 6.85 z= 4.27 azim=-132.9 elev= -67.7 roll=
= 171.8

trackd[2060819]: [con01] val01: 0.000 val02: 0.000
trackd[2060819]: but01: 0 but02: 0 but03: 0

trackd[2060819]: [tra01] x= -4.12 y= 3.03 z= 1.39 azim=-115.7 elev= -26.7 roll=
= 109.1
trackd[2060819]: [tra02] x= -2.49 y= 6.85 z= 4.27 azim=-133.0 elev= -67.7 roll=
= 171.9

trackd[2060819]: [con01] val01: 0.000 val02: 0.000
trackd[2060819]: but01: 0 but02: 0 but03: 0
```

図 7-3 Trackd 実行画面

7.2 AVS/ExpressMPE の利用方法

7.2.1 AVS/ExpressMPE の起動

(1) AVS/ExpressMPE を起動します。

```
% /appli/AVS-MPE/runmpe
```

(2) 通常の AVS/Express と同様にネットワークを作成します。

AVS/Express で作成した既存ネットワークを読み込むこともできます。

7.2.2 MPE 専用ビューワモジュールのインスタンス

AVS/ExpressMPE では、専用のビューワ (WandUviewer3D) を使います。

起動するアプリケーション・タイプによっては、Uviewer3D がデフォルトでインスタンスされていますので、そのような場合は、Uviewer3D を削除したのち WandUviewer3D をインスタンスします。

(1) ネットワーク・エディタのモジュールライブラリから [JAERIwand] を選択します。

(図 7-4)

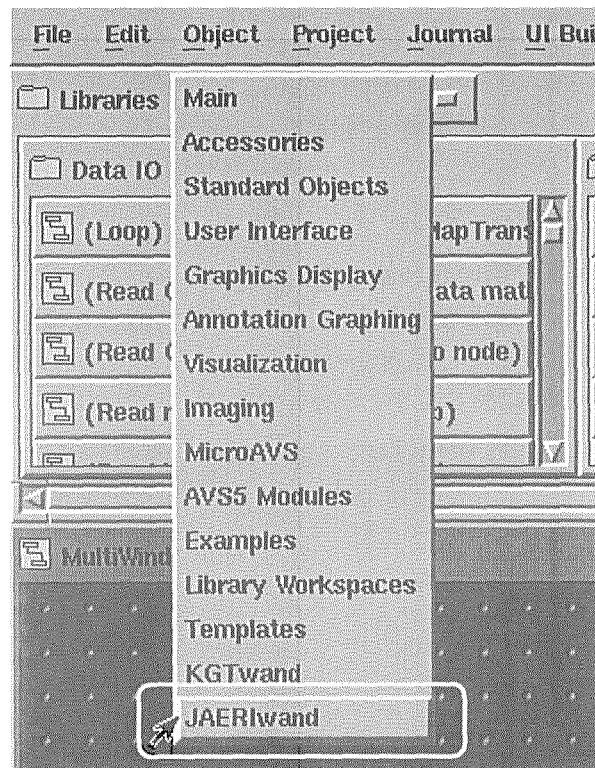


図 7-4 モジュールライブラリ選択画面

(2) wand カテゴリにある [WandUviewer3D2] を選択し、ワークスペースにインスタンスします。

(図 7-5)



図 7-5 [WandUviewer3D2] モジュールのインスタンス

(3) 作成したネットワークを [WandUviewer3D2] に接続します。

7.2.3 ステータス表示の停止

MPE の負荷を軽減するため、ステータス表示をオフにします。(図 7-6)

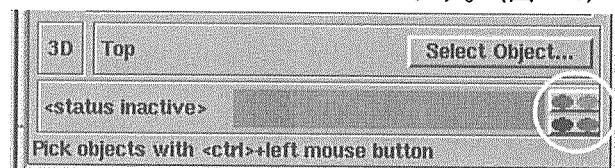


図 7-6 ステータス表示

7.2.4 Wand パラメータの設定変更と起動

MPE の負荷に応じて「Interval(ms)」パラメータを変更します。
このパラメータは、Wnad の情報を問い合わせる間隔です。
間隔が小さいほど、操作はスムーズになりますが、負荷が高くなります。
(基本的には、デフォルト設定のままで問題ありません)
「Start」ボタンを押し、Wand を起動します。(図 7-7)

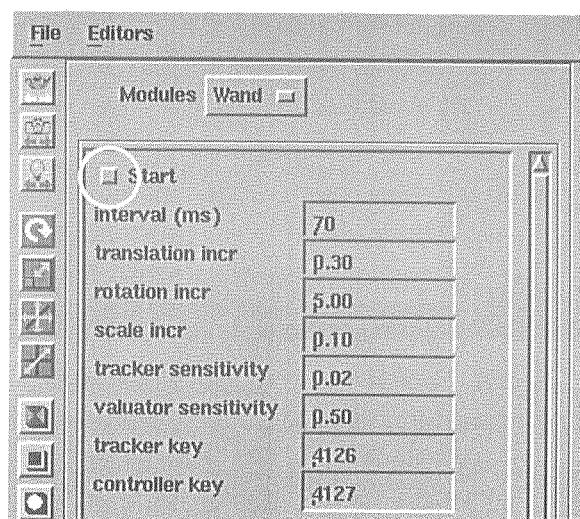


図 7-7 [WandUviewer3D2] モジュールコントロール・パネル

7.2.5 レンダリングモードの切り替え

コントロールパネル [Editors] - [View] を選択します。

Renderer を「OpenGL」から「MPU」に変更します。(図 7-8)

この時点から、[WandUviewer3D2] の可視化データがプロジェクタに表示されます。

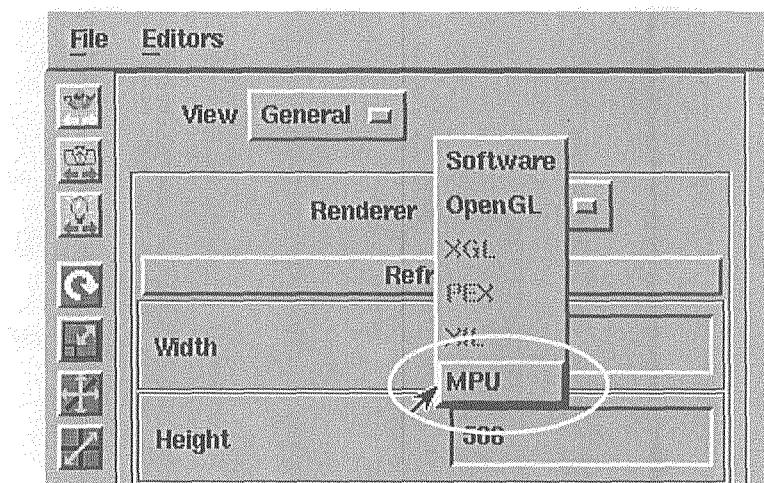


図 7-8 レンダリングモードの切り替え画面

7.3 AVS/ExpressMPE コントロール方法

7.3.1 立体視メガネの装着

プロジェクタに表示されるステレオ画像を立体的に見るために、立体視用メガネ (CrystalEyes) を装着します。(図 7-9)

立体視用メガネの電源は、メガネのツルを開くことでオン状態になります。

(使用しない時は、ツルを閉じた状態にしてください)

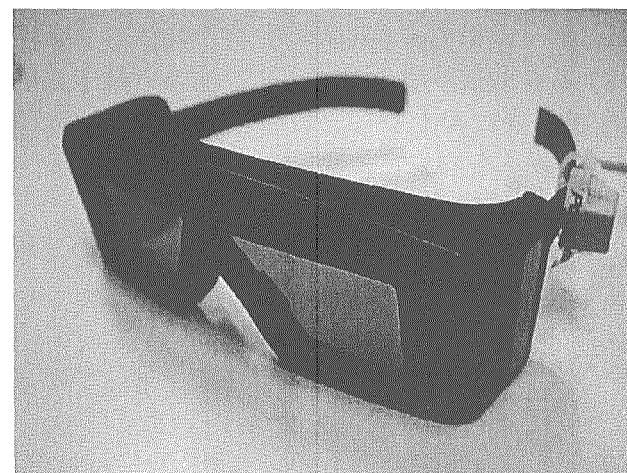


図 7-9 立体視用メガネ (CrystalEyes)

7.3.2 WANDA コントローラの操作方法

プロジェクトに表示される可視化データを操作するには、WANDA コントローラ (WANDA) を使用します。(図 7-10)

3つのボタン (青・黄・赤) がある方をスクリーン側に向けた状態で WAND を操作します。

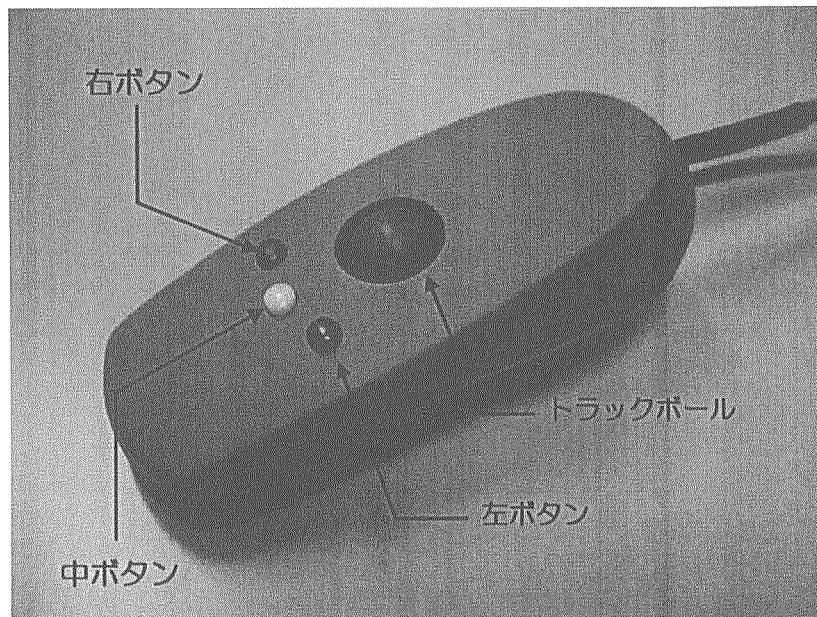


図 7-10 WANDA コントローラ

■ボタンアサイン

- 左ボタン (青) を押しながら WANDA を移動：
→ WANDA を移動した方向に物体を移動します
- 中ボタン (黄) を押しながら WANDA を X の正方向 (右) に移動：
→ 物体を拡大します
- 中ボタン (黄) を押しながら WANDA を X の負方向 (左) に移動：
→ 物体を縮小します
- 右ボタン (赤) を押す：
→ トラックボールを左／右に押した時、物体が Y 軸周りに回転するか Z 軸周りに回転するかを切り替えます (デフォルトは Y 軸周り)
- トラックボールを手前／奥に押す：
→ X 軸周りに物体を回転します
- トラックボールを左／右に押す：
→ Y 軸または Z 軸周りに物体を回転します (Y 軸／Z 軸の切り替えは右ボタンで切り替えます。デフォルトは Y 軸)
- 中ボタン (黄) と右ボタン (赤) を同時に押す：
→ “物体の位置のリセット”、“大きさのノーマライズ”、“回転中心を物体の中心にセット”を行います
- 左ボタン (青) と中ボタン (黄) と右ボタン (赤) を同時に押す：
→ [WandUviewer3D2] モジュールの処理を停止します
再度起動するには、7.4.2 (2) のオペレーションを行います

7.4 AVS/ExpressMPE の終了

7.4.1 Window の切り替え

[Alt] + [F3] キーを押し、現在の表示 Window を最背面に移動します。

7.4.2 レンダリングモードの切り替え

- (1) コントロールパネル [Editors] - [View] を選択します。
- (2) Renderer を「MPU」から「OpenGL」に変更します。(図 7-8)

7.4.3 [WandUviewer3D2] モジュールの停止

コントロールパネル [Editors] - [Modules] - [Wand] を選択します。
「Start」ボタンを押し、Wand を停止します。(図 7-7)

7.4.4 AVS/ExpressMPE の終了

コントロールパネル [File] - [Exit] を選択します。
図 7-11 のようなパネルが表示されるので、[OK] ボタンを押してください。

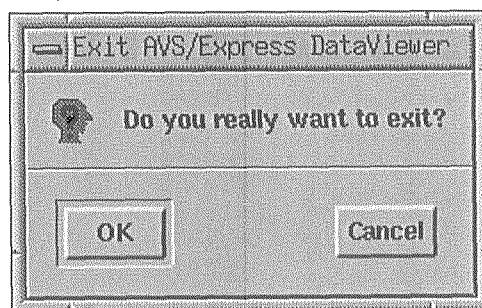


図 7-11 終了パネル

7.5 センシングデータ取得用デーモン (Trackd) の終了

Trackd を起動した Window 内で [Ctrl] + [c] キーを押し、Trackd デーモンを停止します。

謝辞

本報告書を執筆する機会を与えてくださいました計算科学技術推進センター長 矢川元基氏、次長 平山俊雄氏及び情報システム管理課長 山田孝行氏に深く感謝致します。また、本報告書の作成にあたり御指導、御助言をいただきました数値実験技術開発グループ 渡辺 正氏、(株)富士通長野システムエンジニアリングの方々に深く感謝致します。

参考文献

- [1] 株式会社ケイ・ジー・ティー：“AVS／Express ユーザーズ・ガイド”、株式会社ケイ・ジー・ティー、(2000年).
- [2] 株式会社ケイ・ジー・ティー：“AVS／Express ディベロッパーズ・ガイド”、株式会社ケイ・ジー・ティー、(2000年).
- [3] 増子献児、五來一夫、加藤克海、山崎和彦：“AVS(Application Visualization System)利用手引書”、JAERI-Tech 96-008(1996).

付録 索引

2

2D 4

3

3D 4

A

Accessories 6

Ambient 27

Application 4

Arrange 13

AutoNormalize 29

AVS/Express 1, 2

AVS/ExpressMPE 1, 105

AVS_CCompat 6

AVS5 Modules 6

B

base コマンド 75

C

Camera 18

coord 39

CrystalEyes 109

D

Data IO 6

Datamap 18

DefaultObject 7

Diffuse 27

dim 35

DO 41

E

Edit 6

ENDDO 41

EOT 41

Examples 6

express プロセス 74

F

Fastrack 106

field 36

Field 7

Field Mappers 6

Field データ 33

File 6

filetype 38

Filters 6

Fit 13

G

Geometries 6

Gloss 28

Graph 18

Graphics Display 6

H

Help 6

Hue 26

HW/SW 30

I

Imaging 6

irregular 36

J

JAERIwand 107

Journal 6

L

label 36

Library Workspaces 6

Light 18

Load application 4

M

Main 6

Mappers 6

Metal 28

Modules 18

ModuleStack	4
MPU	109
Multi-window DataViewer	4

N

ndim	35
None	4
nspcace	35
nstep	37

O

Object	6, 18
offset	38
Opacity	28
OpenGL	109
Option	6

P

Perspective	28
Print	18
Project	6

R

rectilinear	36
Rendering	24

S

Saturation	26
Scaling	13
Scene	18
Scratch Pad	4
Select Object	22
Single-window DataViewer	4
skip	38
Specular	27
Standard Objects	6
stride	38

T

Templates	6
time	40
Trackd	106
Transform	18

U

UCD データ	33
UI_Builder	6
UIConnection	7
UI モジュール	63
uniform	36
unit	36
User Interface	6
user プロセス	74

V

Value	26
variable	38
VCP	58
veclen	35
View	18
Viewers	6
Visualization	6
▽コマンド	57
▽ステートメント	57
▽言語	57

W

WANDA	110
WandUviewer3D	107

Z

Zoom	13
------------	----

あ

アプリケーション・タイプ	4
アプリケーションの削除	15
アプリケーションの保存	14
アプリケーションの読み込み	15
アルターネート	19

い

移動	21, 110
イメージデータ	33
インスタンス	10

お

オブジェクト	18
--------------	----

オブジェクトの幾何変換	21	デモ	88
オブジェクトの選択	22	ト	
オブジェクトの属性	23	投影法	28
オブジェクトの読み込み	20	透視表示	19
オンライン・ドキュメント	88	に	
か			
回転	18, 21, 110	入出力ポート	7
拡大	18, 21, 110	ね	
画像サーバ	2	ネットワーク	5
カメラ	18	ネットワーク・エディタ	5
カメラの属性	28	は	
環境設定	2	背景色	30
環境変数	2	パイプライン	105
く			
組み込み関数	57	ひ	
グループ・オブジェクト	60	非構造格子データ	33
こ			
構造格子データ	33	ビューワ・タイプ	4
コントロールパネル	17	ふ	
し			
磁気センサー	106	フィールドデータ	33
自作モジュール	65	プリミティブ	7
ジッター機能	19	プリミティブ・オブジェクト	60
縮小	18, 21, 110	プルダウンメニュー	18
す			
ステータス・バー	6	プロジェクト	66
せ			
正規化	19	ヘ	
そ			
ソースファイル	73	ベースタイプ	60
つ			
ツールバー	18	ほ	
て			
データビューワ	16	ボリュームデータ	33
も			
モジュール	1, 5, 10, 20	マクロ・オブジェクト	60
モジュール・アイコン	7	モジュール	
モジュール・オブジェクト	60	モジュール・アイコン	7
モジュール情報	88	モジュール・オブジェクト	60
モジュールの移動	12	モジュール情報	88

モジュールのインスタンス	10	ライン	25
モジュールの開閉	12	り	
モジュールの検索	9	リセット	19, 110
モジュールの消去	11	立体視用メガネ	109
モジュールの情報	13	リンク・オブジェクト	60
モジュールの接続	11	れ	
モジュールの切断	12	レンダリング	30
モジュールの保存	72	わ	
モジュール名の変更	11	ワークスペース	6
モジュールライブラリ	6	ワイルドカード	9
ら			
ライト	18		
ライブラリ	6		

国際単位系(SI)と換算表

表1 SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表3 固有の名称をもつSI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s^{-1}
力	ニュートン	N	$m \cdot kg/s^2$
圧力、応力	パスカル	Pa	N/m^2
エネルギー、仕事、熱量	ジュール	J	$N \cdot m$
工率、放射束	ワット	W	J/s
電気量、電荷	クーロン	C	$A \cdot s$
電位、電圧、起電力	ボルト	V	W/A
静電容量	ファラード	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンス	S	A/V
磁束	ウェーバ	Wb	$V \cdot s$
磁束密度	テスラ	T	Wb/m^2
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	$^{\circ}C$	
光束度	ルーメン	lm	$cd \cdot sr$
照度	ルクス	lx	lm/m^2
放射能	ベクレル	Bq	s^{-1}
吸収線量	グレイ	Gy	J/kg
線量当量	シーベルト	Sv	J/kg

表2 SIと併用される単位

名称	記号
分、時、日	min, h, d
度、分、秒	°, ', "
リットル	l, L
トントン	t
電子ボルト	eV
原子質量単位	u

$$1 \text{ eV} = 1.60218 \times 10^{-19} \text{ J}$$

$$1 \text{ u} = 1.66054 \times 10^{-27} \text{ kg}$$

表5 SI接頭語

倍数	接頭語	記号
10^{18}	エクサ	E
10^{15}	ペタ	P
10^{12}	テラ	T
10^9	ギガ	G
10^6	メガ	M
10^3	キロ	k
10^2	ヘクト	h
10^1	デカ	da
10^{-1}	デシ	d
10^{-2}	センチ	c
10^{-3}	ミリ	m
10^{-6}	マイクロ	μ
10^{-9}	ナノ	n
10^{-12}	ピコ	p
10^{-15}	フェムト	f
10^{-18}	アト	a

(注)

- 表1～5は「国際単位系」第5版、国際度量衡局1985年刊行による。ただし、1 eVおよび1 uの値はCODATAの1986年推奨値によった。
- 表4には海里、ノット、アール、ヘクタールも含まれているが日常の単位なのでここでは省略した。
- barは、JISでは流体の圧力を表す場合に限り表2のカテゴリに分類されている。
- EC閣僚理事会指令ではbar, barnおよび「血圧の単位」mmHgを表2のカテゴリに入れている。

換算表

圧	MPa(=10 bar)	kgf/cm ²	atm	mmHg(Torr)	lbf/in ² (psi)
力	1	10.1972	9.86923	7.50062×10^3	145.038
	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322×10^{-4}	1.35951×10^{-3}	1.31579×10^{-3}	1	1.93368×10^{-2}
	6.89476×10^{-3}	7.03070×10^{-2}	6.80460×10^{-2}	51.7149	1

粘度 $1 \text{ Pa} \cdot \text{s} (\text{N} \cdot \text{s}/\text{m}^2) = 10 \text{ P} (\text{ボアズ}) (\text{g}/(\text{cm} \cdot \text{s}))$

動粘度 $1 \text{ m}^2/\text{s} = 10^4 \text{ St} (\text{ストークス}) (\text{cm}^2/\text{s})$

エネルギー・仕事・熱量	J($=10^7 \text{ erg}$)	kgf·m	kW·h	cal(計量法)	Btu	ft · lbf	eV	1 cal = 4.18605 J(計量法)
	1	0.101972	2.77778×10^{-7}	0.238889	9.47813×10^{-4}	0.737562	6.24150×10^{-18}	$= 4.184 \text{ J} (\text{熱化学})$
	9.80665	1	2.72407×10^{-6}	2.34270	9.29487×10^{-3}	7.23301	6.12082×10^{-19}	$= 4.1855 \text{ J} (15^{\circ}\text{C})$
	3.6×10^6	3.67098×10^5	1	8.59999×10^5	3412.13	2.65522×10^6	2.24694×10^{25}	$= 4.1868 \text{ J} (\text{国際蒸気表})$
	4.18605	0.426858	1.16279×10^{-6}	1	3.96759×10^{-3}	3.08747	2.61272×10^{-19}	仕事率 1 PS(仏馬力)
	1055.06	107.586	2.93072×10^{-4}	252.042	1	778.172	6.58515×10^{21}	$= 75 \text{ kgf} \cdot \text{m/s}$
	1.35582	0.138255	3.76616×10^{-7}	0.323890	1.28506×10^{-3}	1	8.46233×10^{18}	$= 735.499 \text{ W}$
	1.60218×10^{-19}	1.63377×10^{-20}	4.45050×10^{-26}	3.82743×10^{-20}	1.51857×10^{-22}	1.18171×10^{-19}	1	

放射能	Bq	Ci	吸収線量	Gy	rad
	1	2.70270×10^{-11}		1	100
	3.7×10^{10}	1		0.01	1

照射線量	C/kg	R
	1	3876
	2.58×10^{-4}	1

線量当量	Sv	rem
	1	100
	0.01	1

(86年12月26日現在)

AVS/Express (Application Visualization System) 液壓油示範

