

JAERI-Tech  
2003-050



JP0350227



有限要素解析ソフトの利用環境整備  
—ADVENTUREシステム利用手引書—

2003年5月

山崎 一郎・吉村 忍\*

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）で複写による実費領布をお  
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, Ibaraki-ken 〒319-1195, Japan.

# 有限要素解析ソフトの利用環境整備 － ADVENTURE システム利用手引書 －

日本原子力研究所計算科学技術推進センター

山崎 一郎・吉村 忍\*

(2003年2月25日受理)

ITBL 利用推進室では、ITBL の利用・普及を目的として、共有化ソフトウェア・データベースの整備や利用マニュアルの整備などを実施しており、その一環として超並列環境で高い並列処理性能を持つ設計用大規模並列有限要素法解析システム：ADVENTURE（代表開発者：東京大学教授、吉村 忍）の利用環境整備を行った。

本報告書では、ADVENTURE システムの一連のプロセスの操作方法や PC クラスタへのインストール方法について報告する。

---

日本原子力研究所（関西駐在）：〒619-0215 京都府相楽郡木津町梅美台 8-1

\*東京大学

# Environmental Equipment for Usages of FEM Software – ADVENTURE System User's Guide –

Ichirou YAMASAKI and Shinobu YOSHIMURA \*

Center for Promotion of Computational Science and Engineering  
(Kansai Site)  
Japan Atomic Energy Research Institute  
Kizu-cho, Souraku-gun, Kyoto

(Received February 25, 2003)

The community softwares, databases, and other various tools have been installed in the ITBL environment by the Office of ITBL Promotion as a common utility property for each research field. Among those softwares, Finite Element Method (FEM) code, ADVENTURE (which was originally developed by Prof. Yoshimura, the University of Tokyo), is provided as one of structure analysis programs for ITBL users. The code is well known to possess a high performance ability for parallel processing, especially for massively parallel environments. In this report, a chain of processes for usages of the system as well as the installation method to PC cluster are described.

keywords: ITBL, FEM, ADVENTURE

---

\*The University of Tokyo

## 目次

1.	はじめに .....	1
2.	ADVENTURE システム概要 .....	2
2.1.	ADVENTURE とは? .....	2
2.2.	ADVENTURE システム構成 .....	3
2.3.	ADVENTURE システム全体のモジュールフロー .....	5
2.4.	インターフェースサンプル .....	6
3.	ADV_TriPatch の使い方 .....	7
3.1.	概要 .....	7
3.2.	操作手順 .....	8
3.3.	ADV_TriPatch の実行形式 .....	8
3.4.	実行例 1 .....	9
3.5.	実行例 2 .....	9
4.	ADV_TetMesh の使い方 .....	10
4.1.	概要 .....	10
4.2.	操作手順 .....	11
4.3.	TetMesh_P の実行形式 .....	11
4.4.	TetMesh_P の実行例 .....	11
4.5.	TetMesh_M の実行形式 .....	12
4.6.	TetMesh_M の実行例 .....	12
5.	ADV_BCtool の使い方 .....	13
5.1.	概要 .....	13
5.2.	機能 .....	13
5.3.	処理フロー .....	14
5.4.	ステップ 1 : メッシュの表面の抽出 .....	15
5.5.	ステップ 2 : GUI による境界条件の設定 .....	16
5.6.	ステップ 3 : 一体型解析モデルファイルの作成 .....	18
6.	ADV_Metis の使い方 .....	20
6.1.	概要 .....	20
7.	ADV_Solid の使い方 .....	23
7.1.	概要 (ADV_Solid プログラム説明書より抜粋) .....	23
7.2.	実行環境 .....	25
7.3.	実行形式 .....	25
7.4.	実行例 .....	26

8.	ADV_Visual の使い方 .....	29
8.1.	概要 .....	29
8.2.	プログラムの起動.....	30
8.3.	プログラムの初期操作 .....	31
8.4.	プログラムの基本操作 .....	34
8.5.	解析モデル形状の表示 .....	35
9.	インストールについて .....	41
9.1.	プログラムの入手方法 .....	41
9.2.	インストール時の留意点.....	42
9.3.	ADVENTURE_IO のインストール方法.....	43
9.4.	ADVENTURE_TriPatch のインストール .....	44
9.5.	ADVENTURE_TetMesh のインストール.....	45
9.6.	ADVENTURE_BCtool のインストール.....	47
9.7.	ADVENTURE_Metis のインストール.....	49
9.8.	ADVENTURE_Solid のインストール.....	52
9.9.	ADVENTURE_Visual のインストール .....	55
9.10.	MPICH のインストール.....	57
10.	おわりに .....	57
	謝辞 .....	58
	参考文献 .....	58

## Contents

1.	Introduction .....	1
2.	Outline of ADVENTURE System.....	2
2.1.	What is the ADVENTURE System?.....	2
2.2.	ADVENTURE System Configuration .....	3
2.3.	Flow of Modules of the ADVENTURE System .....	5
2.4.	Use Samples of GUI.....	6
3.	How to Use of ADV_TriPatch .....	7
3.1.	Outline .....	7
3.2.	Operation.....	8
3.3.	Starting Method of ADV_TriPatch .....	8
3.4.	Example 1 .....	9
3.5.	Example 2 .....	9
4.	How to Use of ADV_TetMesh .....	10
4.1.	Outline .....	10
4.2.	Operation.....	11
4.3.	Starting Method of TetMesh_P .....	11
4.4.	Example of TetMesh_P .....	11
4.5.	Starting Method of TetMesh_M.....	12
4.6.	Example of TetMesh_M.....	12
5.	How to Use of ADV_BCtool .....	13
5.1.	Outline .....	13
5.2.	Function.....	13
5.3.	Processing Flow .....	14
5.4.	Step1: Extraction of Surface of Mesh .....	15
5.5.	Step2: Setting of Boundary Condition by GUI .....	16
5.6.	Step3: Making of All-in-one Design Analytical Model .....	18
6.	How to Use of ADV_Metis .....	20
6.1.	Outline .....	20
7.	How to Use of ADV_Solid .....	23
7.1.	Outline(Excerpt from Program's Manual on ADV_Solid) .....	23
7.2.	Environment .....	25
7.3.	Starting Method.....	25
7.4.	Example .....	26

8.	How to Use of ADV_Visual.....	29
8.1.	Outline.....	29
8.2.	Starting Method.....	30
8.3.	Initial Operation .....	31
8.4.	Basic Operation.....	34
8.5.	Display of Analytical Model .....	35
9.	Installation .....	41
9.1.	Software Download .....	41
9.2.	Notes .....	42
9.3.	Installation of ADVENTURE_IO.....	43
9.4.	Installation of ADVENTURE_TriPatch .....	44
9.5.	Installation of ADVENTURE_TetMesh .....	45
9.6.	Installation of ADVENTURE_BCtool .....	47
9.7.	Installation of ADVENTURE_Metis .....	49
9.8.	Installation of ADVENTURE_Solid .....	52
9.9.	Installation of ADVENTURE_Visual .....	55
9.10.	Installation of MPICH .....	57
10.	Summary.....	57
	Acknowledgment .....	58
	Reference.....	58

## 1. はじめに

ITBL 環境でのコミュニティソフトとして、有限要素解析ソフトのユーザ利用環境整備を進めている。従来の汎用解析計算ソフトは、100 万自由度が上限であり、主に逐次処理計算機向けで、単一現象解析にしか対応できない。小さな規模の解析しかできないと大型構造物・機器の解析計算において、全体挙動の評価を行うか、もしくは局部挙動の評価を行うかの二者択一を迫られる。全体挙動に注目すると簡略化されたモデルのため、局部構造とのインテラクションが分らず、局部挙動の評価だけでは、境界条件の与え方、またどの局部構造に注目すべきか分からぬ。

設計用大規模並列有限要素法解析システム : ADVENTURE<sup>1)</sup> (東京大学、吉村教授) は、これら従来の汎用解析ソフトの問題を解決すべく設計・開発され、超並列環境で高い並列処理性能を持ち、単一の現象の解析ばかりでなく、各種連成解析や設計解析の 3 次元丸ごと詳細解析機能により、大規模な解析計算を可能とする。また、ソースは無料公開され、改良や商用利用にも原則として制限を設けておらず、CAD データとのインターフェースや大規模並列有限要素法（領域分割）のための IO を標準化・公開するなど、ITBL 環境でのコミュニティソフトに適したソフトと言える。

ADVENTURE を ITBL 環境に導入する前に、ADVENTURE システムの基本的調査と動作確認を実施した。実際に教育用 PC クラスタへ移植を行い、メッシュ生成・領域分割・並列計算・並列可視化など一連の動作を確認した。

本報告書は、これら作業の成果をまとめたもので、ADVENTURE システムの各モジュールの操作方法を手順に従って解説する。また、PC クラスタ (Compaq Proliant DL360: 2CPU×36 ノード) へのインストール方法等について報告する。

## 2. ADVENTURE システム概要

### 2.1. ADVENTURE とは？

ADVENTURE とは、日本学術振興会未来開拓学術研究推進事業「計算科学」分野の設計用大規模計算力学システム開発プロジェクト（吉村教授、東京大学大学院・新領域創成科学研究科）から生まれたソフトウェアで、正式名を「設計用大規模並列有限要素法解析システム（ADvanced Engineering analysis Tool for Ultra large Real world）」という。

プロジェクトは、以下の事項<sup>2)</sup>を目標にかかげる。

- 産業界の現場において、100万～1億自由度の実用問題の解析を、従来の100～1千自由度の解析程度の労力で実現できる CAE システムを開発する。
- CAD 世界から独立するために、IGES ファイルの受け渡し方式を採用、さらに、移植性を高めるために、GUI は最小限に留める。
- ソースを無料公開し、改良や商用利用に原則として一切制限を設けない（ADVENTURE の成果を利用している旨は明記する）…並列 FEM の産業界への普及を促進する。
- 大規模並列有限要素法（領域分割型）のための IO を標準化し、公開する … デファクトスタンダードを目指す。
- 比較的廉価な PC クラスター+フリーOS（Linux）+フリー通信ライブラリー（MPI）に加えて、フリー並列 FEM ソフトの登場により、真の意味で並列処理がパーソナルな技術となる。

## 2.2. ADVENTURE システム構成

ADVENTURE システムのモジュール構成を図 2.1 に示す。

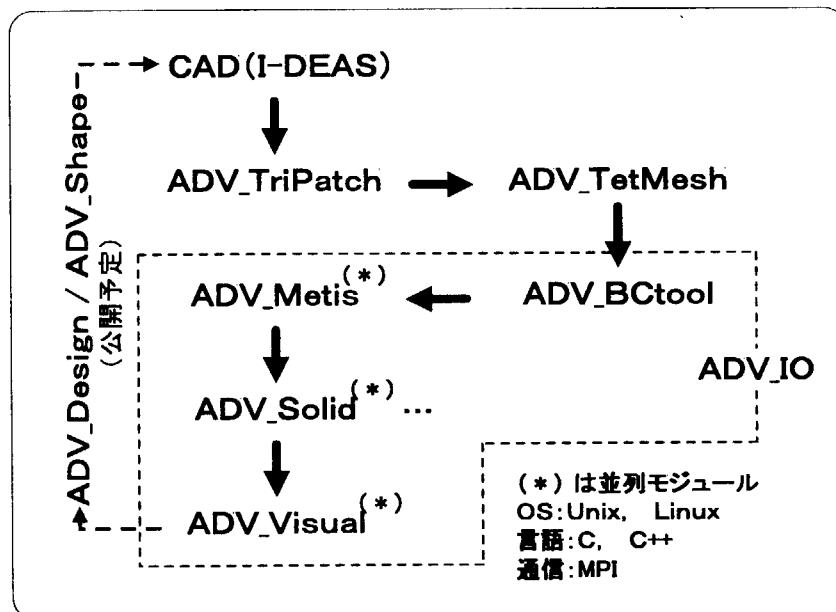


図 2.1 モジュール構成<sup>2)</sup>

表 2.1 は、リリースされた ADVENTURE システムの Ver1.0 のモジュール群である。

表 2.1 ADVENTURE システム Ver1.0 のモジュール一覧<sup>1)</sup>

モジュール名	機能	開発・実行環境
ADV_TriPatch	IGES フォーマットの B-rep 表現ソリッドモデル(単一材料、複数材料)から 3 角形表面パッチデータを生成する。	OS: Linux コンパイラ: C++
ADV_TetMesh	3 角形表面パッチデータから 4 面体メッシュを自動生成する。	OS: Unix, Linux コンパイラ: C++, Fortran90
ADV_BCTool	メッシュに境界条件及び物性値を貼付し、有限要素解析データを作成する作業を支援する。	OS: Unix, Linux コンパイラ: C++ ライブラリ: OpenGL, Motif
ADV_Metis	メッシュを単層の部分領域あるいは 2 階層のパートと部分領域に分割する処理を並列に実行する。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Solid	静弾性 / 静弾塑性 / 大変形応力解析。 対角スケーリングなし BBD 前処理付き階層型領域分割法版、及び対角スケーリング付き並列 CG 版。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Visual	解析結果を並列に可視化する。変形図、表面・断面におけるスカラー分布表示。	OS: Unix, Linux コンパイラ: C++ グラフィックス: OpenGL, Motif
ADV_IO	ADVENTURE モジュール用 IO フォーマットとライブラリ。	OS: Unix, Linux コンパイラ: C, C++
※ ADV_BaseSourcePack	ADVENTURE モジュール(Ver 1.x)の全アーカイブをまとめたもの	

リリースされた ADVENTURE システムの β 版のモジュールは、表 2.2 のとおり。

表 2.2 ADVENTURE システム β 版モジュール一覧<sup>1)</sup>

モジュール名	機能	開発・実行環境
ADV_Fluid	非圧縮性流体解析機能。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Impact	階層型領域分割による陽解法動弾性解析 / 接触解析。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Magnetic	線形渦電流解析 / 非線形静磁場解析。 階層型領域分割法版。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Forge	素材と工具の接触・摩擦機能を持った剛塑性解析。 階層型領域分割法版。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Thermal	固体中の定常 / 非定常熱伝導解析。 階層型領域分割法版。	OS: Unix, Linux コンパイラ: C 通信ライブラリ: MPI
ADV_Shape	力法に基づくノンパラメトリック形状最適化 / 密度法によるトポロジー最適化。	OS: Unix, Linux コンパイラ: C
ADV_Opt	逐次 2 次計画法 / 実数値遺伝的アルゴリズムに基づくパラメトリック最適化。	OS: Unix, Linux コンパイラ: C
ADV_CAD	ポリゴン中心のソリッドモデルの自動作成。2 次元形状の押し出し・回転による 3 次元形状定義 / 3 次元形状のブール演算。	OS: Unix, Linux コンパイラ: C++
ADV_Auto	バッチ処理モードで自動解析や自動最適化を行うためのユーティリティ。境界条件を設定する場所の自動判別機能 / 解析結果から最大応力などの代表値を抽出する機能。	OS: Unix, Linux コンパイラ: C++
ADV_iAgent	操作プランの生成とユーザナビゲーションによる統合支援。解析プランの提示 / ホストマシンの稼動状況表示 / ユーザの状況に応じたプッシュ型情報提供。	OS: Unix, Linux コンパイラ: Java 通信ライブラリ: Java RMI(Remote Method Invocation)
ADV_Material	実験データから構成方程式のパラメータを同定する。	OS: Unix, Linux コンパイラ: Java
ADV_FEMAPtool	NASTRAN インプット …> ADVENTURE メッシュ変換[nas2adv]。 ADVENTURE の解析結果(Solid/Thermal) …> FEMAP neutral ファイル変換[adv2femap]。	OS: Unix, Linux コンパイラ: C

### 2.3. ADVENTURE システム全体のモジュールフロー

ADVENTURE システム全体のモジュールフローを図 2.2 に示す。

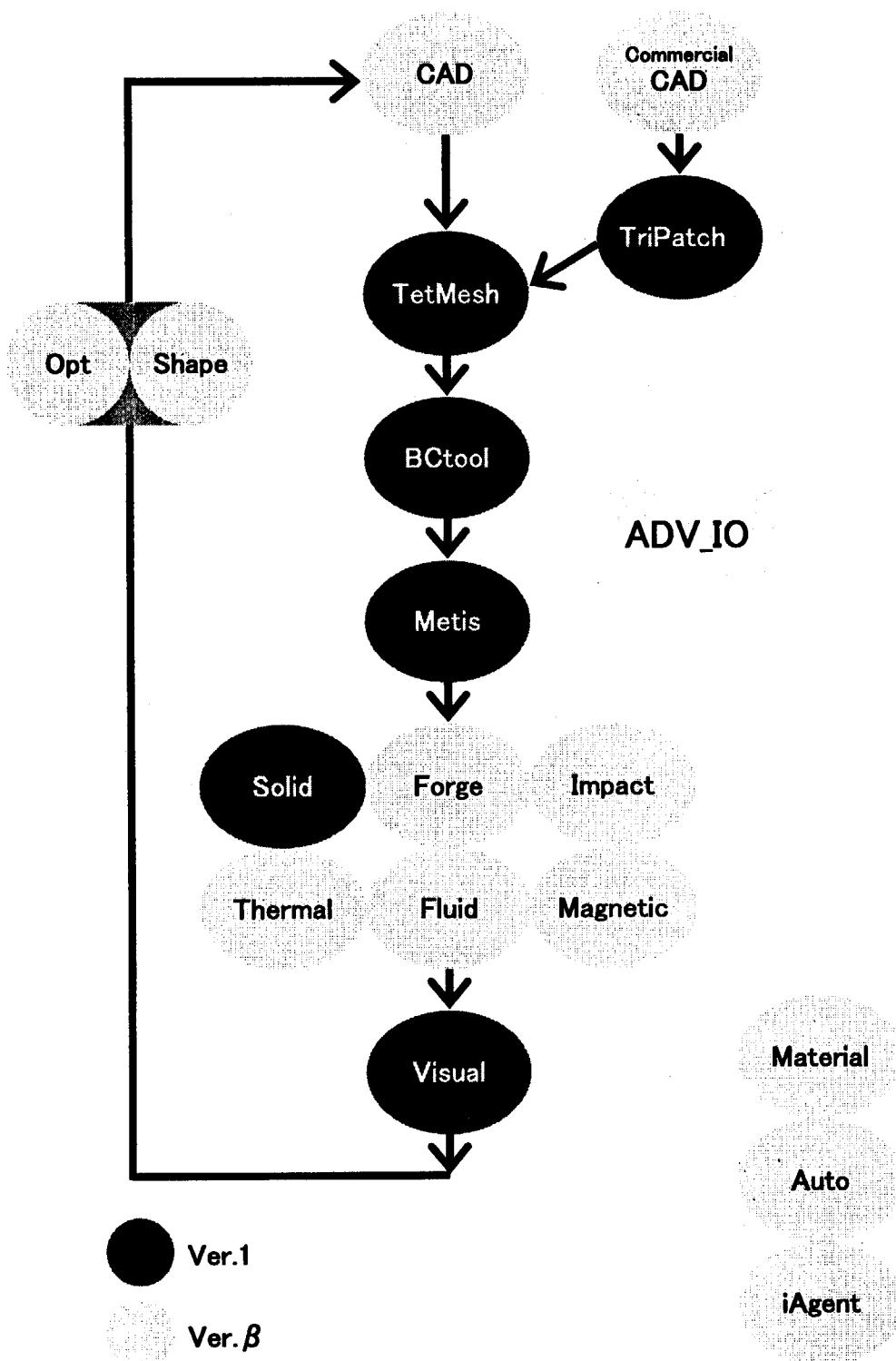


図 2.2 ADVENTURE システムのモジュールフロー<sup>1)</sup>

## 2.4. インターフェースサンプル

ADVENTURE システムが提供するグラフィカルインターフェース（図 2.3～図 2.6）を 2 点（ADV\_BCtool と ADV\_Visual）紹介する。

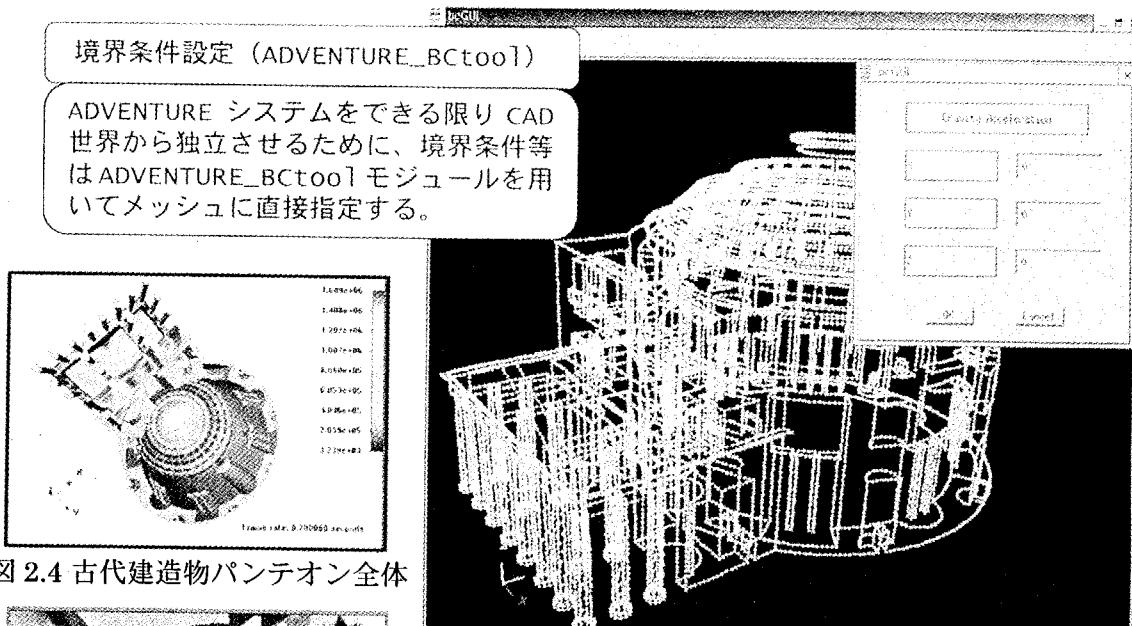


図 2.4 古代建造物パンテオン全体

図 2.3 ADVENTURE\_BCtool

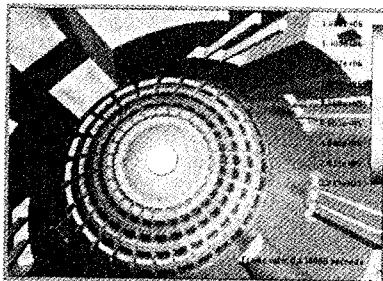


図 2.5 古代建造物パンテオン拡大

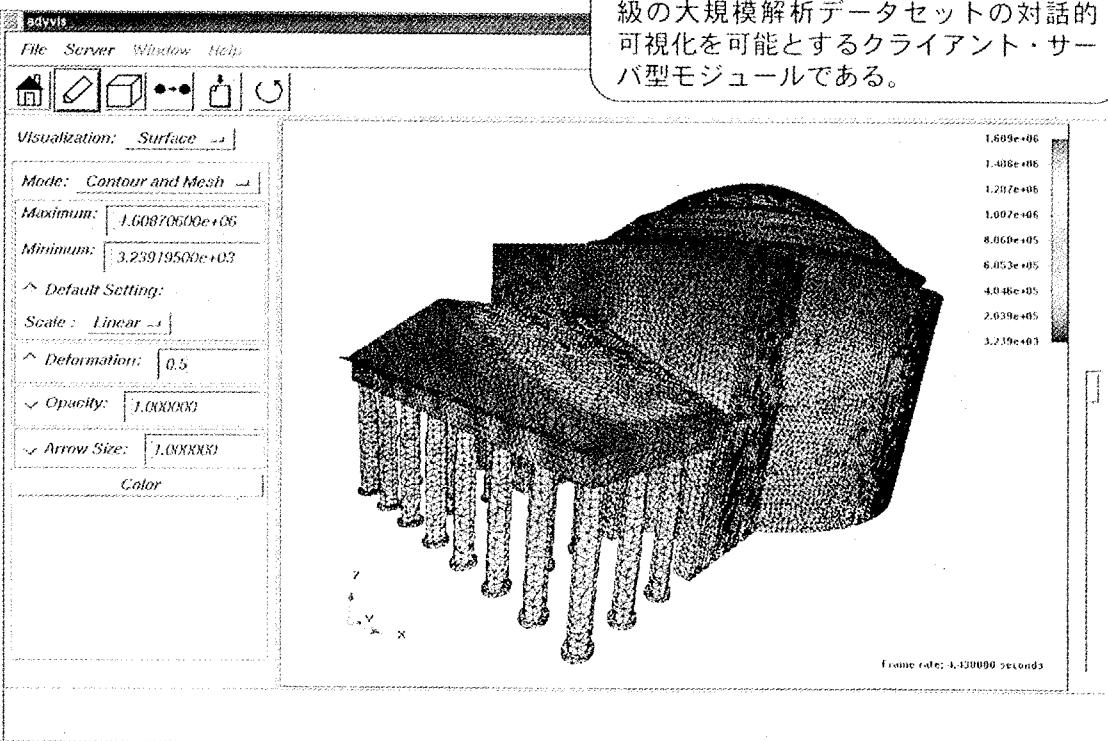


図 2.6 ADVENTURE\_Visual

### 3. ADV\_TriPatch の使い方

#### 3.1. 概要

本プログラムは、IGES 形式のファイルフォーマット（NURBS 曲面ベースのソリッドデータ）を入力し、3 角形の表面パッチを作成する。作成された 3 角形の表面パッチ情報は、以下のファイルに出力される。図 3.1 に ADV\_TriPatch<sup>3)</sup>の IPO (Input/Process/Output) を示す。

- (1) 表面パッチデータファイル（ファイル拡張子 : pcm）※旧フォーマットの場合は、pch  
表面パッチの座標情報、コネクティビティデータを含むデータを保有
- (2) 表面パッチ VRML ファイル（ファイル拡張子 : wrl）  
表面パッチを VRML 形式に変換したデータを保有
- (3) 表面パッチグループデータファイル（ファイル拡張子 : pcg）  
表面パッチをグループ化した情報を含むデータを保有

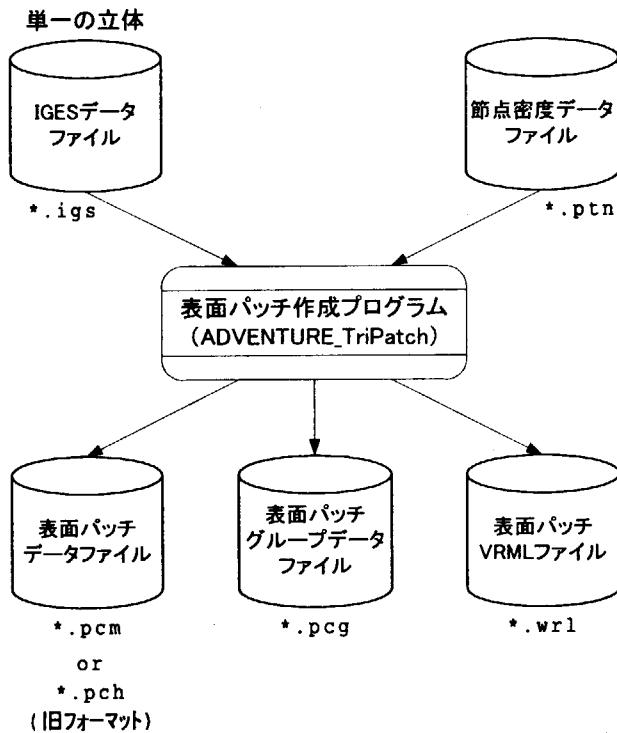


図 3.1 ADV\_TriPatch の IPO (Input/Process/Output)

### 3.2. 操作手順

- ① IGES ファイルの用意 (\*.igs)
- ② 節点密度ファイルの作成 (\*.ptn)
- ③ プログラムの実行

表面パッチ作成プログラム (ADV\_TriPatch) は、上記手順にて実行する。基本的に入力データファイルが準備できれば、プログラムを実行するだけである。が、もう一つ、表面パッチ結合プログラム (mrpach) が用意されている。mrpach は、複数の表面パッチデータファイル (=複数の IGES ファイル) に含まれる領域間の整合化処理を行い、複数材料用の表面パッチデータを作成する。

### 3.3. ADV\_TriPatch の実行形式

- 呼び出し形式 :  

```
ADVENTURE_TriPatch igsFile ptnFile --out_pch_form
```
- 入力ファイル : ※拡張子を除いて入力。  

igsFile	: IGES データファイル	(拡張子は igs)
ptnFile	: 節点密度データファイル	(拡張子は ptn)
- オプション :  

--out_pch_form	: 旧フォーマットの表面パッチデータファイルを出力する。
	(拡張子は pch)
- 出力ファイル :  

pcmFile	: 表面パッチデータファイル	(拡張子は pcm)
pcgFile	: 表面パッチグループデータファイル	(拡張子は pcg)
wrlFile	: 表面パッチ VRML ファイル	(拡張子は wrl)

### 3.4. 実行例 1

表面パッチ作成プログラムの実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
# usage:
# \ADVENTURE_TriPatch sample[.igs] sample[.ptn] --out_pch_form
# output:
#   sample.pcm(or sample.pch) sample.pcg sample_c_mcp.wrl
ADVENTURE_TriPatch adventure_manual_data01 adventure_manual_data01 \
--out_pch_form
```

実際に実行すると、adventure\_manual\_data01.pch と adventure\_manual\_data01.pcg と adventure\_manual\_data01\_c\_mcp.wrl（図 3.2）が出力される。

### 3.5. 実行例 2

表面パッチ結合プログラムの実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
# usage:
#   mrpatch sample1.pcm sample1.pcg sample2.pcm sample2.pcg \
#             -o merge.pcm -g merge.pcg -v merge.wrl
mrpatch mat_in01.pcm mat_in01.pcg mat_in02.pcm mat_in02.pcg \
-o merge.pcm -g merge.pcg -v merge.wrl
```

二つの表面パッチデータファイル（mat\_in01.pcm と mat\_in02.pcm）を結合している。結合データは、merge.pcg と merge.wrl（図 3.3）となる。

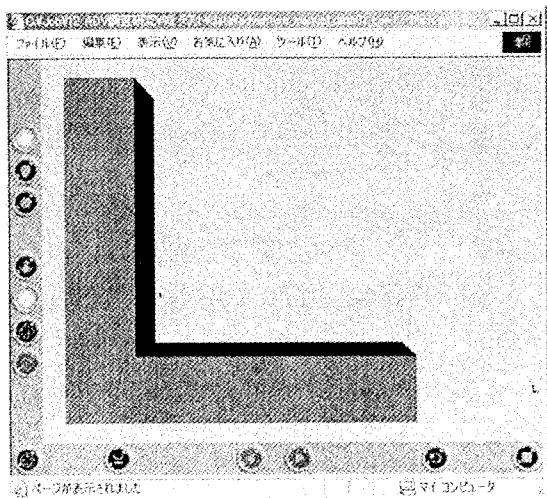


図 3.2 実行例 1  
adventure\_manual\_data01\_c\_mcp.wrl の形状

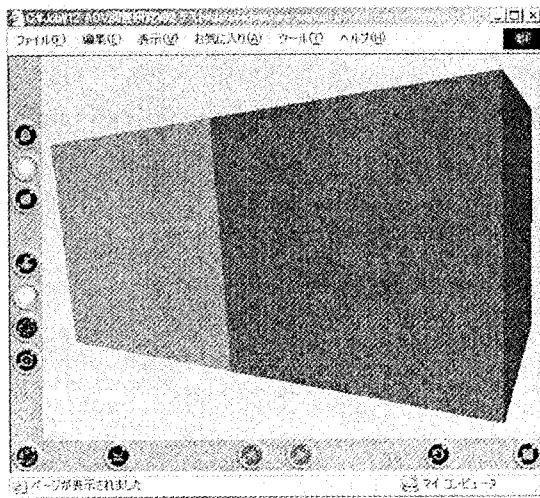


図 3.3 実行例 2  
merge.wrl の形状

## 4. ADV\_TetMesh の使い方

### 4.1. 概要

本プログラムは、3 角形表面パッチを入力し Delaunay 四面体分割により四面体メッシュを作成する。本プログラムは、表面パッチを Pliant Delaunay Retriangulation により平滑化と Delaunay 化するモジュール TetMesh\_P<sup>4)</sup>と、Delaunay 四面体分割により四面体メッシュを作成するモジュール TetMesh\_M<sup>4)</sup>の 2 つのモジュールで構成される。

作成された四面体メッシュの情報は、以下のファイルに出力される。

(1) 四面体メッシュデータファイル（ファイル拡張子：msh）

四面体メッシュの節点座標および要素コネクティビティ

(2) 表面 VRML ファイル（ファイル拡張子：wrl）

メッシュ表面を VRML 形式に変換したデータ（2 種類）

図 4.1 に ADV\_TetMesh<sup>4)</sup>の IPO を示す。

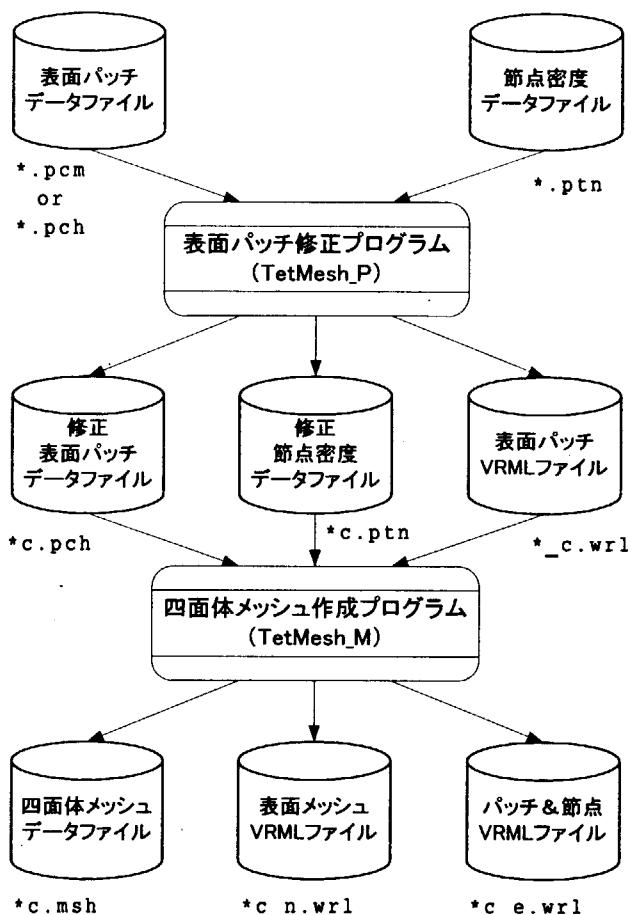


図 4.1 ADV\_TetMesh の IPO

## 4.2. 操作手順

- ① 表面パッチデータファイルの用意 (\*.pch)
- ② 節点密度ファイルの作成 (\*.ptn)
- ③ 表面パッチ修正プログラムの実行 (TetMesh\_P)
- ④ 四面体メッシュ作成プログラムの実行 (TetMesh\_M)

表面パッチデータファイルは、ADV\_TriPatch の出力を想定。ただし、ここでは旧フォーマットの pch ファイルとする。また、ADV\_TriPatch の出力を使用する場合、節点密度ファイルは、パッチ作成時に使用した節点密度ファイルをそのまま使用する。

## 4.3. TetMesh\_P の実行形式

- 呼び出し形式 :

```
advtmesh8p pchFile -d -p
```

- 入力ファイル :

pchFile	: 表面パッチデータファイル	(拡張子は pch)
---------	----------------	------------

- オプション :

-d	: 節点密度データファイルを使用する
----	--------------------

-p	: VRML 形式ファイルを出力する
----	--------------------

- 出力ファイル :

pchFile	: 修正表面パッチデータファイル	(拡張子は pch)
---------	------------------	------------

ptnFile	: 修正節点密度データファイル	(拡張子は ptn)
---------	-----------------	------------

wrlFile	: 表面パッチ VRML ファイル	(拡張子は wrl)
---------	-------------------	------------

## 4.4. TetMesh\_P の実行例

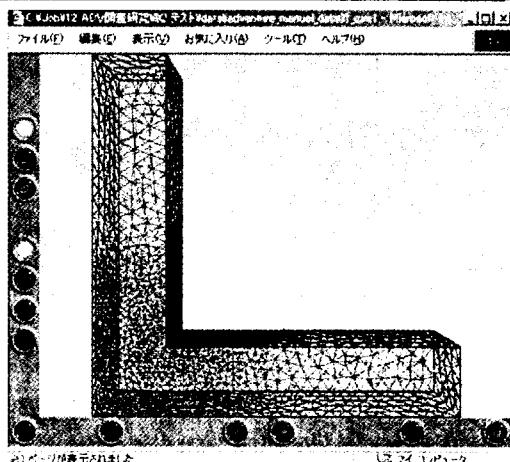
表面パッチ修正プログラムの実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
# usage:
#   advtmesh8p sample[.pch] -d -p
#   output) samplec.pch samplec.ptn sample_c.wrl
advtmesh8p adventure_manual_data01 -d -p
```

実際に実行すると、

adventure\_manual\_data01c.pch と  
adventure\_manual\_data01c.ptn と  
adventure\_manual\_data01\_c.wrl (図 4.2)  
が出力される。

図 4.2 adventure\_manual\_data01\_c.wrl の形状



#### 4.5. TetMesh\_M の実行形式

- 呼び出し形式:

```
advtmesh8m pchFile -s -p
```

- 入力ファイル:

pchFile : 修正表面パッチデータファイル (拡張子は pch)

- オプション:

-s : 四面体メッシュに 2 次節点を付加する

-p : VRML 形式ファイルを出力する

- 出力ファイル:

mshFile : 四面体メッシュデータファイル (拡張子は msh)

wrlFile : 表面メッシュ VRML ファイル (拡張子は wrl)

wrlFile : 表面メッシュ VRML ファイル (拡張子は wrl)

#### 4.6. TetMesh\_M の実行例

四面体メッシュ作成プログラムの実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
# usage:
#   advtmesh8m samplec[.pch] -s -p
#   output) samplec.msh samplec_n.wrl samplec_e.wrl
advtmesh8m adventure_manual_data01c -s -p
```

実際に実行すると、adventure\_manual\_data01c.msh と adventure\_manual\_data01c\_e.wrl (図 4.3) と adventure\_manual\_data01c\_n.wrl (図 4.4) が出力される。

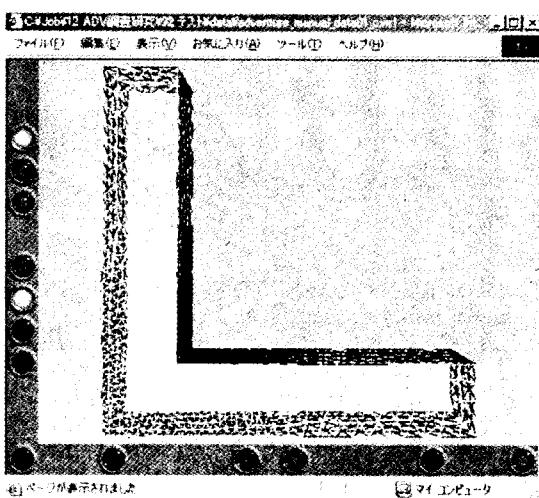


図 4.3 adventure\_manual\_data01c\_e.wrl

四面体メッシュの表面 VRML

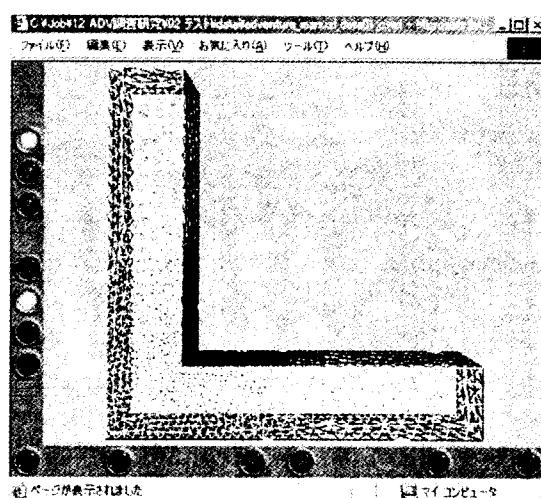


図 4.4 adventure\_manual\_data01c\_n.wrl

四面体メッシュの節点の VRML

## 5. ADV\_BCtool の使い方

### 5.1. 概要

ADVENTURE\_BCtool<sup>5)</sup>はメッシュに境界条件及び物性値を貼付し、ADVENTURE\_Solid のための有限要素解析データを作成するためのツールである。

### 5.2. 機能

#### 5.2.1. 対応している要素の種類

4面体1次要素、4面体2次要素、6面体1次要素、6面体2次要素

#### 5.2.2. 対応している解析の種類

弾性解析

弾塑性解析

熱応力解析

#### 5.2.3. 境界条件を設定できる場所

グループ化したメッシュ表面

グループ化したメッシュ表面の境界上の1次節点

#### 5.2.4. 設定できる境界条件

荷重 (X, Y, Z 方向、面に垂直方向)

変位 (X, Y, Z 方向、面に垂直方向)

重力加速度

#### 5.2.5. 設定できる物性値 (複数物性値に対応)

ヤング率

ポアソン比

加工硬化係数

初期降伏応力

質量密度

熱膨張係数

参照温度

### 5.3. 処理フロー

ADV\_BCtool<sup>5)</sup>の IPO を図 5.1 に示す。

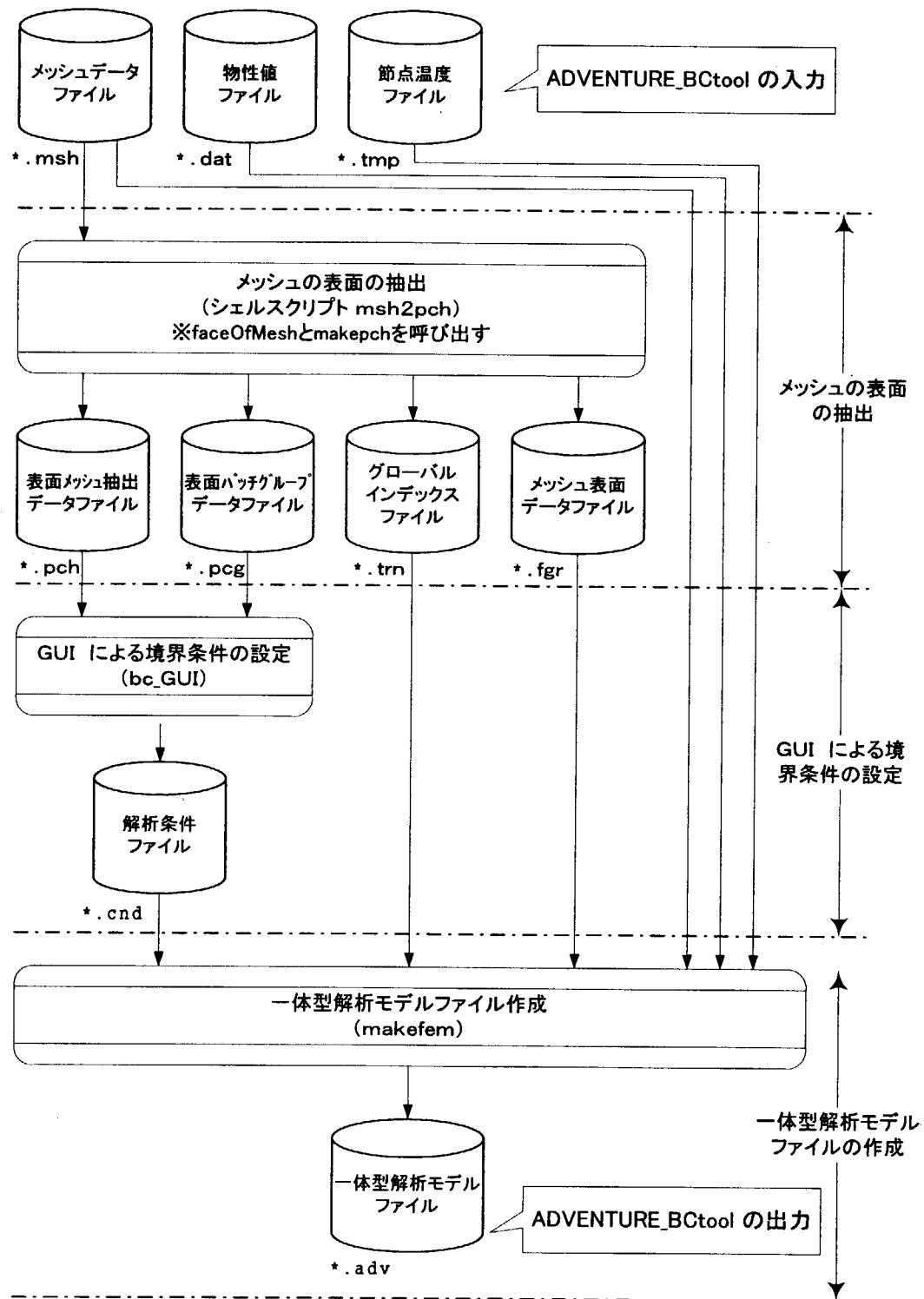


図 5.1 ADV\_BCtool の IPO

## 5.4. ステップ1：メッシュの表面の抽出

メッシュの表面を抽出およびグループ化し、GUI の入力フォーマットへ変換する。

### 5.4.1. 実行形式

- 呼び出し形式：

```
msh2pch mshFile div_n
```

- 入力ファイル：

**mshFile** : メッシュデータファイル (拡張子は **msh**)

**div\_n** : メッシュ表面のグループ化の基準となる 2 面狭角の指定。

2 面狭角が指定した角度以上の面は別のグループになる。

180 度の何分の 1 かで指定する。

例1) メッシュデータファイル名が **Model.msh**、2 面狭角が 60 度 (=180/3)

```
% msh2pch Model.msh 3
```

例2) メッシュデータファイル名が **Model.msh**、2 面狭角が 45 度 (=180/4)

```
% msh2pch Model.msh 4
```

- 出力ファイル：メッシュデータファイル **Model.msh**、2 面狭角が 180/N の場合。

**Model\_N.fgr** : メッシュ表面データファイル (拡張子は **fgr**)

**Model\_N.pch** : 表面メッシュ抽出データファイル (拡張子は **pch**)

**Model\_N.pcg** : 表面パッチグループデータファイル (拡張子は **pcg**)

**Model\_N.trn** : グローバルインデックスファイル (拡張子は **trn**)

### 5.4.2. 実行例

実行用シェルスクリプト **msh2pch** の実行例 (動作確認に使用したスクリプト)。

```
#!/bin/sh
# usage:
#   msh2pch sample.msh 2
# output:
#   sample_2.fgr sample_2.pch sample_2.pcg sample_2.trn
msh2pch adventure_manual_data01.msh 2
```

実際に実行すると、**adventure\_manual\_data01\_2.fgr** と **adventure\_manual\_data01\_2.pch** と **adventure\_manual\_data01\_2.pcg** と **adventure\_manual\_data01\_2.trn** が出力される。

## 5.5. ステップ2：GUIによる境界条件の設定

境界条件の設定には、GUIベースのモジュールである bcGUI を使用して、解析条件ファイルを作成する。

### 5.5.1. 実行形式

- 呼び出し形式：

```
bcGUI pchFile pcgFile [-icnd cndFile] [-ocnd outFile]
```

- 入力ファイル：

pchFile	：表面メッシュ抽出データファイル	(拡張子は pch)
---------	------------------	------------

pcgFile	：表面パッチグループデータファイル	(拡張子は pcg)
---------	-------------------	------------

- オプション：

-icnd	：起動時に読み込む解析条件ファイル
-------	-------------------

-ocnd	：終了時に書き込む解析条件ファイル
-------	-------------------

- 出力ファイル：

cndFile	：解析条件ファイル	(拡張子は cnd)
---------	-----------	------------

### 5.5.2. 実行例

境界条件設定用 bcGUI プログラムの実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
# usage:
#   bcGUI sample_2.pch sample_2.pcg [-icnd cndFile] [-ocnd outFile]
bcGUI adventure_manual_data01_2.pch adventure_manual_data01_2.pcg
```

bcGUI は、X Window System のクライアントアプリケーションである。起動の際には、事前に X 環境を準備する必要がある。本例では、Windows 2000 上で、X 端末エミュレータを使用して bcGUI を起動している。図 5.2 は起動直後の画面である。

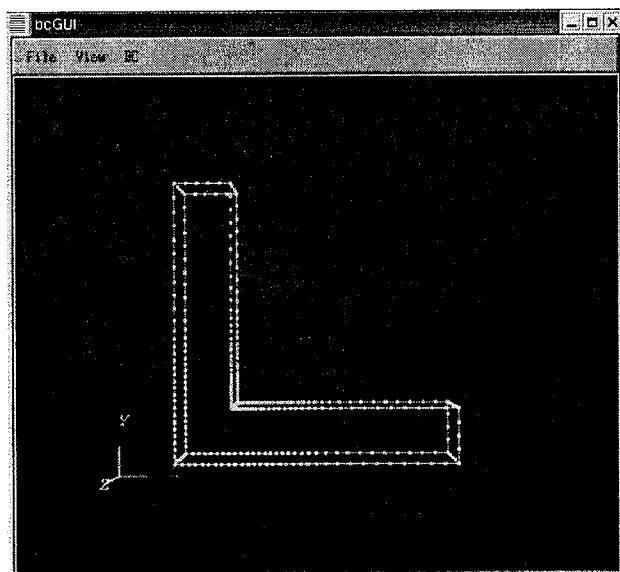


図 5.2 初期表示

画面操作のポイントは下述(図 5.3~図 5.9)。

### ●視点の移動方法

- 平行移動 : マウス左ボタン押下+ドラッグ
- 拡大／縮小 : マウス右ボタン押下+上下ドラッグ  
上方向で縮小、下方向で拡大
- 物体の回転 : マウス中ボタン押下+ドラッグ

### ●節点の選択

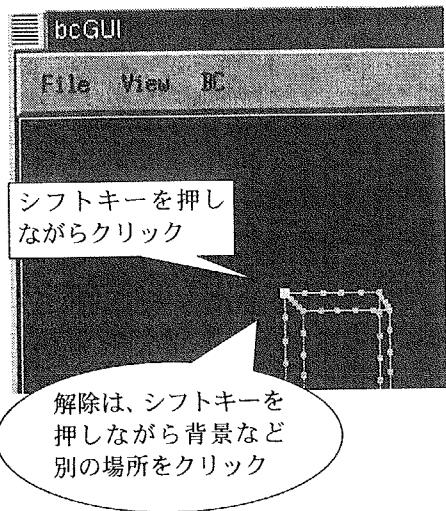


図 5.3 節点の選択

### ●面グループの選択

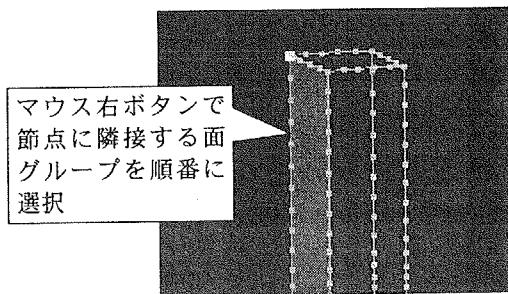


図 5.4 面グループの選択

### ●境界条件の設定 (荷重、変位)

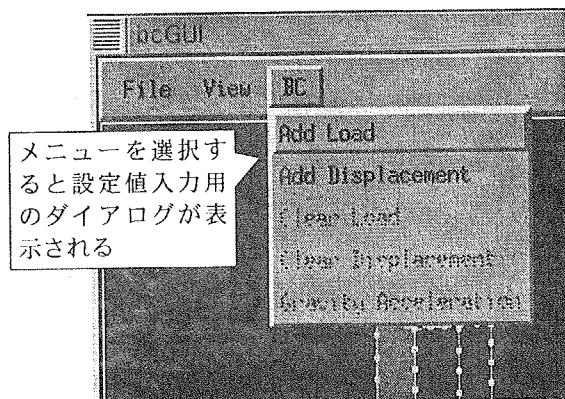


図 5.5 境界条件の設定

### ●重力加速度の設定方法

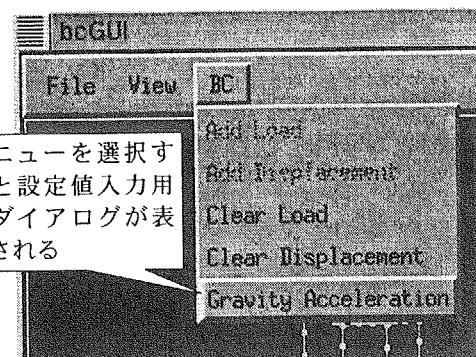


図 5.6 重力加速度の設定

### ●境界条件の表示方法

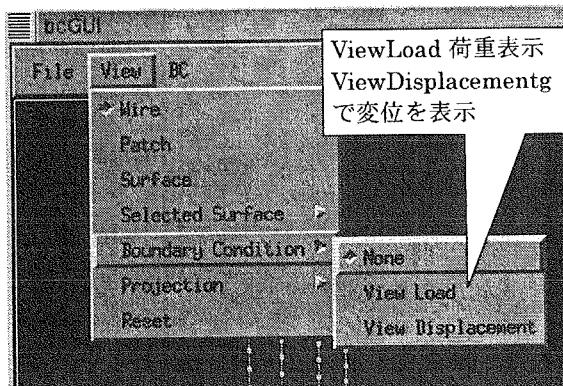


図 5.7 境界条件の表示方法

### ●境界条件のクリア

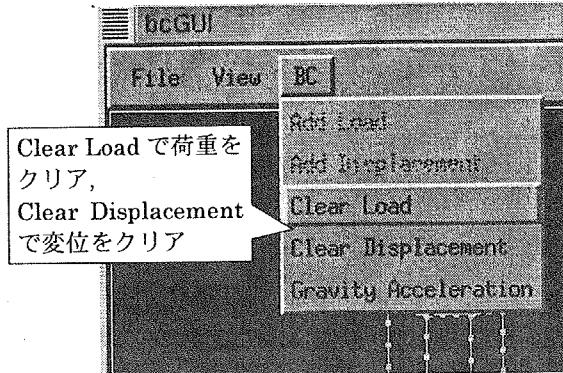


図 5.8 境界条件のクリア

### ●解析ファイルの出力

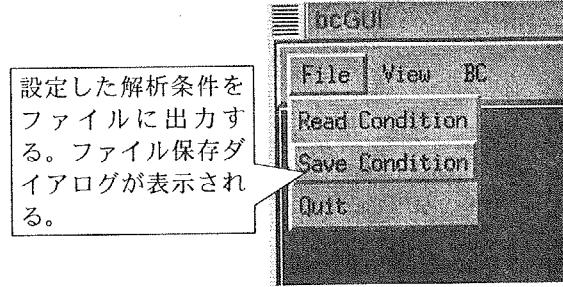


図 5.9 解析ファイルの出力

### 5.5.3. 解析条件ファイルの例

```

gravity 0 0 -9.8
boundary 10
loadOnVertex 271 0 10.5
loadOnVertex 271 1 10.5
loadOnVertex 271 2 10.5
dispOnVertex 8 0 0
dispOnVertex 8 1 0
dispOnVertex 8 2 0
loadOnFaceGroup 1 1 5.2
loadOnFaceGroup 2 0 2 -1.1
dispOnFaceGroup 3 0 0 0
dispOnFaceGroup 3 0 1 0

```

## 5.6. ステップ3：一体型解析モデルファイルの作成

メッシュに対して境界条件と物性値を貼り付け、ADVENTURE\_IO 形式の一体型解析モデルファイルを作成する。

### 5.6.1. 実行形式

- 呼び出し形式 :

```

makefem mshFile fgrFile cndFile matFile advFile
        [-t trnFile] [-temperature tmpFile] [-f femFile]

```

- 入力ファイル :

mshFile	: メッシュデータファイル	(拡張子は msh)
fgrFile	: メッシュ表面データファイル	(拡張子は fgr)
cndFile	: 解析条件ファイル	(拡張子は cnd)
matFile	: 物性値ファイル	(拡張子は dat)
advFile	: 一体型解析モデルファイル	(拡張子は adv)

- オプション :

-t	: グローバルインデックスファイルを指定	(拡張子は trn)
-temperature	: 節点温度ファイルを指定	(拡張子は tmp)

- 出力ファイル :

advFile	: 一体型解析モデルファイル	(拡張子は adv)
femFile	: 一体型解析モデルファイル (ascii 形式)	(拡張子は fem)

※bcGUI を使用して cndFile を作成した場合は、必ず trnFile を指定する。

※tmpFile は熱応力解析を行うときにのみ指定する。

※femFile はデバッグ用の出力。

### 5.6.2. 実行例

一体型解析モデルファイル作成プログラム makefem の実行例（サンプルスクリプト）。

```
#!/bin/sh
# usage:
#   makefem sample.msh sample_2.fgr sample.cnd sample.dat sample.adv \
#             [-t trnFile] [-temperature tmpFile] [-f femFile]
makefem sample.msh sample_2.fgr sample.cnd s_material.dat sample.adv
```

出力ファイルは、sample.adv。

### 5.6.3. 物性値ファイルの例

YoungModulus	21000.0
PoissonRatio	0.4
HardeningParameter	1000.0
YieldStress	500.0
Density	760.0

## 6. ADV\_Metis の使い方

### 6.1. 概要

ADVENTURE\_Metis<sup>6)</sup>は、ADVENTURE Projectにおいて開発中の、階層型領域分割法（HDDM）を用いた並列処理による有限要素法解析システム（ADVENTURE system）のための領域分割ツール（Domain Decomposer）である。このソフトウェアは、以下の特徴を持っている。

- (1) メッシュを単層に、あるいは2階層に分割する。
- (2) ミネソタ大学の開発したグラフ分割ライブラリ METIS と ParMETIS を用いている（これらは ADVENTURE\_Metis ソースアーカイブに含まれており、別途入手する必要はない）。
- (3) FEM メッシュデータ（一体型解析モデル）を入力とし、階層型領域分割法（HDDM）用入力データ（HDDM 型解析モデル）を出力する。入出力のデータフォーマットは ADVENTURE file format である。
- (4) 並列処理が可能である。並列化に MPI を用いている。コンパイルと実行には MPI が必要である。
- (5) ADVENTURE Project でリリースされるソルバに入力する解析モデル作成に用いることができる。

ADVENTURE システム全体における ADVENTURE\_Metris の位置づけについては固体解析ソルバ ADVENTURE\_Solid など、各ソルバのドキュメントを参照されたい。

#### 6.1.1. 実行環境

本書をまとめるにあたり、動作確認のため、PC クラスタを使用している。PC クラスタ（2 CPU × 36 ノード）はコンパック製（Compaq ProliantDL）で、並列処理環境は SCore Cluster System Software を用いている。並列プログラムの実行では、PBS というバッチシステムを使用する。

### 6.1.2. 実行形式

- 呼び出し形式：  
`mpirun [mpioptions] adventure_metis [options] advFile DATA div_num`
- PBS/Score 環境呼び出し形式：  
`sc_qsub [scoptions] adventure_metis [options] advFile DATA div_num`
- 必須引数：  

<code>advFile</code>	：一体型解析モデルファイル	(拡張子は <code>adv</code> )
<code>DATA</code>	：領域分割型解析モデルファイルの出力ディレクトリ	
<code>div_num</code>	：2段階目の領域分割数を指定する	

※1段階目の領域数は、部分数と呼び、ADVENTURE\_Metis を動作させるプロセス数。  
 ※2段階目の領域数は、各部分あたりの分割数。
- オプション：  
  - `-HDDM`  
 標準的な階層型領域分割を行う（デフォルト）。  
 ADVENTURE\_Solid への入力には、このモードを用いる。
  - `-MAGNETIC`  
 ADVENTURE\_Magnetic への入力向けの階層型領域分割を行う。
  - `-decompose-file filename`  
 領域分割後のファイル名を指定する。ただし、部分番号を表す数字とその前のアンダーバーと拡張子 (\*.adv) は除いたものを指定する。デフォルトは、advhddm\_in。
  - `-subdir_name directory`  
 ファイルを出力するディレクトリのサブディレクトリを指定する。領域分割後のファイルはこのサブディレクトリ中に出力される。デフォルトは、model である。
  - `-memlimit n`  
 各プロセッサが使用できるメモリ量を MByte 単位で指定する。動的に確保されたメモリの総計から解放された分を引いた値がこの値を超えるとプログラムは強制終了する。デフォルトでは無制限。
  - `-ls filename`  
 各プロセッサから出力されるログの出力先を filename に指定する。デフォルトでは、起動プロセッサの標準エラー出力。
  - `-ll n`  
 ll は、Log Level の略。このオプションは、現在の処理の段階や要した時間などについてのログの詳細の度合いを制御する。  
`-ll 0` : ログを出力しない。  
`-ll 1` : おまかなくーチン毎についてのみログを出力する。  
`-ll 2` : さらに細かいサブルーチンについてもログを出力する。  
`-ll 3` : 主なループ作業について進行状況をリアルタイムに表示する。
  - `-nin n`  
 入力ファイルを読み込むプロセス数を指定する。デフォルトでは 1 つのプロセスが入力ファイルを読み込むようになっている。
  - `-r, -nr`  
`-r` オプションをつけると、部分領域を生成するときに最適化リナンバリングを行う。`-nr` オプションをつけると最適化リナンバリングは行わない。デフォルトでは行う。
  - `-h, -help`  
 簡単なヘルプを表示する。

### 6.1.3. 実行例

領域分割プログラム adventure\_metis の実行例（動作確認に使用したスクリプト）。

```
#!/bin/sh
PROGRAM=adventure_metis
ADVFILE=/home/itblpg/j8513/job/adventure/test/sample.adv
DATADIR=/home/itblpg/j8513/job/adventure/test/SAMPLE
sc_qsub -q large -nodes=16 ${PROGRAM} ${ADVFILE} ${DATADIR} 100
```

SAMPLE/model ディレクトリ下に advhddm\_in\_XX.adv が output される。※XX は 0~15。

#### ●出力一部抜粋

```
SCOUT: Spawning epcsl6 epcsl5 epcsl4 epcsl3 epcsl2 epcsl1 epcsl0 epcsl9
epcsl8 epcsl7 epcsl6 epcsl5 epcsl4 epcsl3 epcsl2 done.
<0:0> SCORE: 16 nodes (16x1) ready.

[r00000]: Not Found "conf.pdiv(default filename)"...
[r00000]: All strings of ADV-Format were set to default.

[r00000]: start Read NOP data.

[r00000]: Read NOP data end. 0.150689 [sec]
[r00000]: (Read FEGA data end. 0.161328 [sec])
[r00000]:
[r00000]: start Pre-Filter.
[r00000]: Pre-Filter end. 0.095057 [sec]

[r00000]: start ParMETIS.
[r00000]: (ParMETIS start.)
[r00000]: (ParMETIS end.: 0.0685091 [sec])
[r00000]: ParMETIS end. 0.068992 [sec]
[r00000]:
[r00000]: start convert_metas.
[r00000]: (converter start.)
[r00000]: (converter end.: 0.016946 [sec])
[r00000]: (Distribute FEGA-by-Element start.)
[r00000]: (Distribute FEGA-by-Element end. 0.011750 [sec])
[r00000]: Convert_metas end. 0.028889 [sec]
[r00000]:
[r00000]: start METIS.

[r00000]: METIS end. 0.030403 [sec]
[r00000]:
[r00000]: start Post-Filter.

[r00000]: Post-Filter end. 2.791225 [sec]
[r00000]: Sum: 3.389449 [sec]
[r00000]:
[r00000]: MemUse = 0, MaxUse = 458229
[r00003]: MemUse = 0, MaxUse = 422076
[r00010]: MemUse = 0, MaxUse = 487444

[r00013]: MemUse = 0, MaxUse = 426244
[r00009]: MemUse = 0, MaxUse = 414904
[r00014]: MemUse = 0, MaxUse = 451812
SCOUT: Session done.
```

## 7. ADV\_Solid の使い方

### 7.1. 概要 (ADV\_Solid プログラム説明書より抜粋)

このドキュメントは ADVENTURE Project において開発された、固体静解析のための有限要素法解析ソルバ ADVENTURE Solid<sup>7)</sup> の使用マニュアルである。階層型領域分割法 (Hierarchical Domain Decomposition Method, 以下 HDDM) に基づいた線形方程式ソルバを採用しており、並列計算機環境に対応することで大規模な解析を可能としている。ADVENTURE Solid は以下のようない特徴を持っている。

- 線形方程式ソルバとして HDDM、並列 CG 法および BDD (Balancing DomainDecomposition) 前処理つき HDDM の 3 種類が使用可能である。
- HDDM ソルバによる並列実行では、動的な負荷分散が可能である。
- 解析種類は弾性/弾塑性/幾何学的非線形応力解析。
- 弹塑性/幾何学的非線形解析は荷重/変位制御による増分法。
- 非線形解析には、後退型 Euler 法による応力積分とコンシスティント接線剛性を使用。
- 4 面体、6 面体のそれぞれ 1 次、2 次ソリッド要素に対応。
- 対応プラットフォームは Unix, Linux。
- 並列処理ライブラリには MPI [8] を使用し、MPP やネットワークにより接続された PC またはワークステーションといった多様な並列環境に対応する。もちろん単一プロセッサ上の実行も可能である。

ADVENTURE システムにおいて ADVENTURE Solid およびその前後における処理の流れは図 7.1 のようになっている。

#### (1) メッシュデータの作成(ADVENTURE TetMesh)

ADVENTURE TetMesh を用い、解析対象に対してメッシュ分割を行う。他のメッシュ分割プログラムや、手動で作成したメッシュも、フォーマットを変換することで取り込むことが可能。

#### (2) 境界条件の設定(ADVENTURE BCtool)

ADVENTURE BCtool を用い、解析対象のメッシュに対して境界条件を付加する。

物性値の設定もここで行う。

#### (3) 領域分割(ADVENTURE Metis)

ADVENTURE Metis を用い、一体型の解析モデルより階層型に領域分割された解析モデルを作成する。並列処理が可能である。

#### (4) FEM 解析(ADVENTURE Solid)

領域分割された解析モデルを入力として、ADVENTURE Solid により有限要素法解析を行う。並列処理が可能である。

#### (5) ポストシステム(ADVENTURE Visual)

ADVENTURE Visual を用いて解析結果を可視化する。並列処理が可能である。

なお、動作環境は Linux および Unix であり、並列計算には MPI が必要である。MPI には種々の実装が存在するが、フリーなものとしては MPICH 等がある。MPICH は非

常に多くの環境をサポートしているため、MPI が用意されていない環境でも多くの場合これをインストールすることで並列版の ADVENTURE\_Solid を使用出来るであろう。

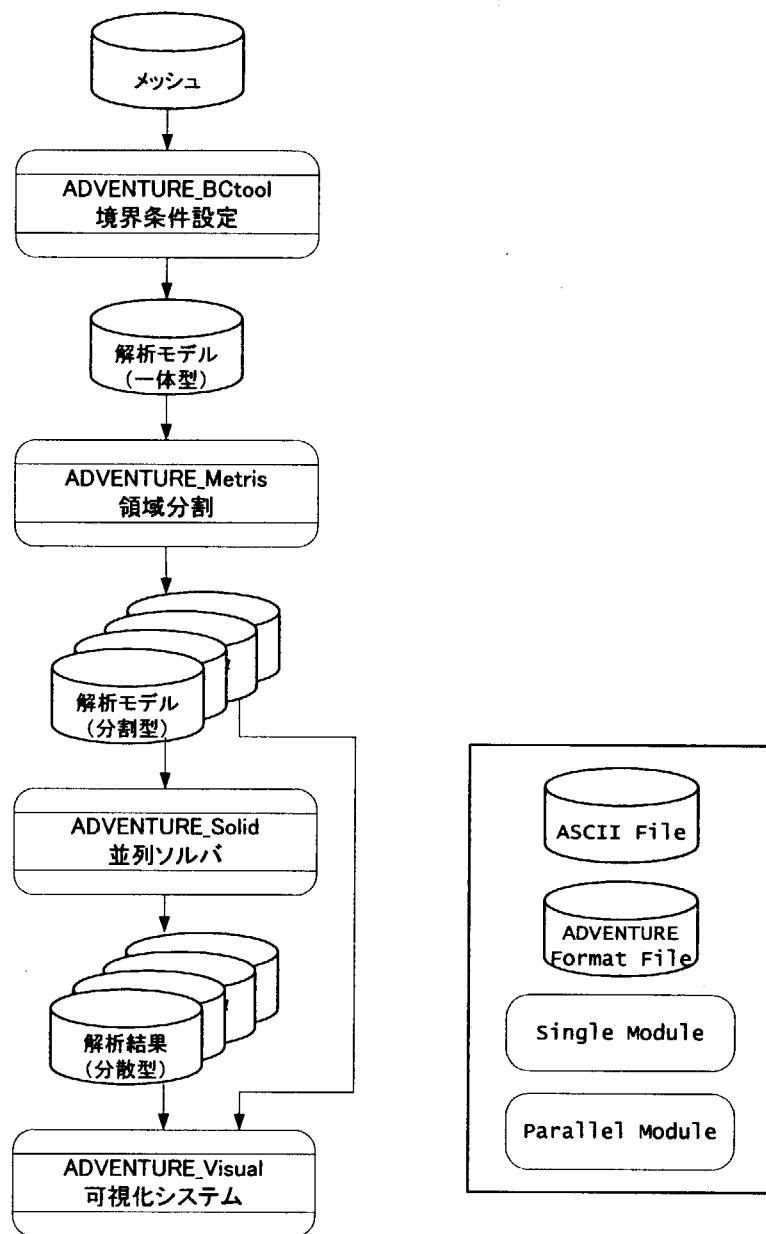


図 7.1 ADV\_Solid の前後の IPO<sup>7)</sup>

## 7.2. 実行環境

本書をまとめるにあたり、動作確認のため、PC クラスタを使用している。PC クラスタ（2 CPU×36 ノード）はコンパック製（Compaq ProliantDL）で、並列処理環境は SCore Cluster System Software を用いている。並列プログラムの実行では、PBS というバッチシステムを使用する。

## 7.3. 実行形式

ADVENTURE\_Solid では、並列方式により 3 つの実行バイナリ（表 7.1）が用意されている。

表 7.1 ADV\_Solid の実行バイナリ<sup>7)</sup>

並列方式	コマンド名	対応ソルバ	領域分割における部分数
シングル版	<code>advsolid-s</code>	HDDM, CG	任意
静的付加分散版	<code>advsolid-p</code>	HDDM, CG, BDD	部分数 = 使用ノード数
動的付加分散版	<code>advsolid-h</code>	HDDM, BDD	部分数 = 親プロセス数

本書では、並列版の `advsolid-p` を実行する。

- 呼び出し形式：  
`mpirun [mpioptions] advsolid-p [options] data_dir`
- PBS/Score 環境呼び出し形式：  
`sc_qsub [scoptions] advsolid-p [options] data_dir`
- 必須引数：  
`data_dir` : 入出力データファイルのあるトップディレクトリ

実行時に可能なオプションは、「解析種類の指定」、「要素に関するオプション」、「出力データ指定オプション」、「増分ステップコントロールオプション」、「反復法のコントロールオプション」、「入出力ファイル名の変更オプション」、「ソルバ指定オプション」等あるが、ここでは、解析種類の指定オプションについてのみ触れる。詳細は、ADVENTURE\_Solid のプログラム使用マニュアルを参照のこと。

### ◆ 解析種類の指定：

下記の指定がなければ、線形弾性解析を行う。

- `-ep`  
 弹塑性解析を行う。モデル作成時に加工硬化係数と初期降伏応力を指定しておく必要がある。
- `-tl`  
 Total Lagrange 法による幾何学的非線形解析を行う場合指定する。弾性解析、弾塑性解析ともに有効であり、大変位微小歪問題を扱うことができる。下の`-ul` オプションとは共存しないのでいずれかを指定する。

## ● -u1

Updated Lagrange 法による幾何学的非線形解析を行う場合指定する。弾性解析、弾塑性解析とともに有効であり、大変位大歪問題を扱うことができる。上の-tl オプションとは共存しないのでいずれかを指定する。

可能な解析と指定オプションの関係は表 7.2 のようになる。

表 7.2 解析の種類とオプション<sup>7)</sup>

解析の種類	オプション
線形弾性解析	
弾性大変位微小歪み解析	-tl
弾性大変位大歪み解析	-u1
弾塑性解析	-ep
弾塑性大変位微小歪み解析	-ep -tl
弾塑性大変位大歪み解析	-ep -u1

## 7.4. 実行例

実行例として、サンプルデータを用いて動作確認した際のスクリプトを記述する。

```
#!/bin/sh
PROGRAM=advsolid-p
DATADIR=/home/itblpg/j8513/job/adventure/test/SAMPLE
sc_qsub -q large -nodes=16 ${PROGRAM} -result --disp --str --stra \
${DATADIR}
```

## ●出力一部抜粋

```

SCOUT: Spawning epcsl6 epcsl5 epcsl4 epcsl3 epcsl2 epcsl1 epcsl0 epcsl9
epcsl8 epcsl7 epcsl6 epcsl5 epcsl4 epcsl3 epcsl2 done.
<0:0> SCORE: 16 nodes (16x1) ready.
Starting /home/itblpg/j8513/ADVENTURE/bin/advsolid-p (Version: 1.0)
  Date: Tue Oct  8 16:44:35 2002
  Options:      -result      --disp      --str      --stra
/home/itblpg/j8513/job/adventure/test/SAMPLE

[0] Start reading /home/.../SAMPLE/model/advhddm_in_0.adv, time 0.049901
[1] Start reading /home/.../SAMPLE/model/advhddm_in_1.adv, time 0.10157
[2] Start reading /home/.../SAMPLE/model/advhddm_in_2.adv, time
0.153327

[0] Start re-reading /home/.../SAMPLE/model/advhddm_in_0.adv, time
0.856727
[0] Finish re-reading /home/.../SAMPLE/model/advhddm_in_0.adv, time
0.860791
[0] Start sending sets of global params , time 0.860952
[1] Got 1 sets of global params
[2] Got 1 sets of global params
Start solver, time 0.897961
CGloop 0: Conv 1.000000e+00, Norm 1.547540e+02, time 1.136843
CGloop 1: Conv 9.344270e-01, Norm 1.446063e+02, time 1.171814
CGloop 2: Conv 1.014517e+00, Norm 1.570007e+02, time 1.200869

CGloop 1873: Conv 1.215649e-06, Norm 1.881265e-04, time 57.064643
CGloop 1874: Conv 1.090338e-06, Norm 1.687343e-04, time 57.095825
CGloop 1875: Conv 9.446254e-07, Norm 1.461846e-04, time 57.128781
CGloop 1876: End CG_LAST, time 57.197147
Finish solver, time 57.1975

[0] Start writing /home/.../SAMPLE/result/advhddm_out_0.adv, time 57.9789
[1] Start writing /home/.../SAMPLE/result/advhddm_out_1.adv, time 59.465
[2] Start writing /home/.../SAMPLE/result/advhddm_out_2.adv, time 59.353

[0] Finish writing /home/.../SAMPLE/result/advhddm_out_0.adv, time
64.1724
[1] Finish writing /home/.../SAMPLE/result/advhddm_out_1.adv, time
64.2183
[2] Finish writing /home/.../SAMPLE/result/advhddm_out_2.adv, time
64.2526

[0] Memory usage [byte]: current 0, max 6728944
[1] Memory usage [byte]: current 0, max 6660780
[2] Memory usage [byte]: current 0, max 6596220

[0] CPU usage [s]: elapsed 64.8444, user 60.87, system 0.26
[1] CPU usage [s]: elapsed 64.8445, user 59.64, system 0.07
[2] CPU usage [s]: elapsed 64.8446, user 59.76, system 0.03

Finish date: Tue Oct  8 16:45:40 2002
SCOUT: Session done.

```

**●実行結果**

SAMPLE ディレクトリ下に結果ファイル（表 7.3）が出力される。

表 7.3 解析結果ファイル

解析モデルファイル	最終解析結果ファイル
advhddm_in_0.adv	advhddm_out_0.adv
advhddm_in_1.adv	advhddm_out_1.adv
advhddm_in_2.adv	advhddm_out_2.adv
advhddm_in_3.adv	advhddm_out_3.adv
advhddm_in_4.adv	advhddm_out_4.adv
advhddm_in_5.adv	advhddm_out_5.adv
advhddm_in_6.adv	advhddm_out_6.adv
advhddm_in_7.adv	advhddm_out_7.adv
advhddm_in_8.adv	advhddm_out_8.adv
advhddm_in_9.adv	advhddm_out_9.adv
advhddm_in_10.adv	advhddm_out_10.adv
advhddm_in_11.adv	advhddm_out_11.adv
advhddm_in_12.adv	advhddm_out_12.adv
advhddm_in_13.adv	advhddm_out_13.adv
advhddm_in_14.adv	advhddm_out_14.adv
advhddm_in_15.adv	advhddm_out_15.adv

## 8. ADV\_Visual の使い方

### 8.1. 概要

ADVENTURE\_Visual<sup>8)</sup>はPCクラスタ上で1千万から1億自由度クラスの大規模解析結果の可視化をインタラクティブに行うことを目的としたクライアント・サーバ型の可視化システムである。PCクラスタ上で稼動するサーバは解析結果の読み込みから可視化情報生成までの演算を並列に実行する。ユーザは、GUIを含むクライアント側のウィンドウ上で操作によって3次元グラフィックスによる可視化表示を得ることができる。

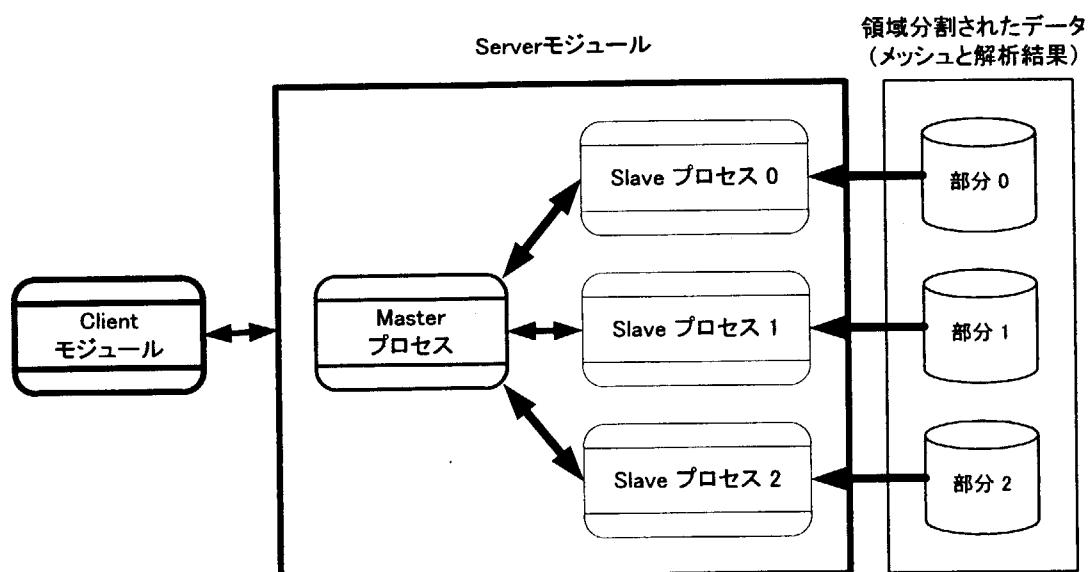


図 8.1 Server-Client 構成<sup>8)</sup>

図8.1のようにGUIを含むClientモジュールとServerモジュールで構成される。ServerモジュールはClientモジュールから起動され、複数のホスト上でMaster・Slave型の分散並列処理を行う。Slaveプロセスは階層型領域分割の各部分の可視化情報の抽出を担当する。したがって、Slaveプロセスの数は階層型領域分割の部分数と同じである。

利用者は、Clientモジュール(advvis)をX端末上で起動して使用する。Serverモジュール(advvis\_server)は上述のとおり、Clientモジュールから起動されることになるが、そのため、事前にServerモジュール環境の設定ファイル(adv-visual.sys)を準備しておく必要がある。

## 8.2. プログラムの起動

可視化プログラム advvis を起動する前に、2つの設定ファイルを準備する。

### 8.2.1. Server モジュール環境の設定

Client モジュール（advvis）を起動するディレクトリに Server モジュール環境の設定ファイル（adv-visual.sys）を作成する。このファイルは advvis 起動時に読み込まれる。

#### adv-visual.sys の例

```

11111  <= Server モジュールの通信ポート番号
33333  <= Client モジュールと Server モジュール間の通信ポート番号
33      <= 登録するホストの数
epcs00                      <= ホスト名
/home/.../ADVENTURE/bin/advvis_server  <= Server モジュールのパス
epcs01
/home/.../ADVENTURE/bin/advvis_server
epcs02
/home/.../ADVENTURE/bin/advvis_server
...
epcs32
/home/.../ADVENTURE/bin/advvis_server

```

### 8.2.2. 解析データセットの設定

可視化する解析データファイルのパスを記述した解析データセット設定ファイル（ファイル名は任意）を作成する。任意のディレクトリに作成できる。

#### 解析データセット設定ファイルの例

```

16
/home/.../adventure/data/DATA/model/advhddm_in_0.adv
/home/.../adventure/data/DATA/result/advhddm_out_0.adv
/home/.../adventure/data/DATA/model/advhddm_in_1.adv
/home/.../adventure/data/DATA/result/advhddm_out_1.adv
/home/.../adventure/data/DATA/model/advhddm_in_2.adv
/home/.../adventure/data/DATA/result/advhddm_out_2.adv
...
/home/.../adventure/data/DATA/model/advhddm_in_15.adv
/home/.../adventure/data/DATA/result/advhddm_out_15.adv

```

### 8.2.3. プログラムの起動

X 端末上で、Client モジュール advvis を起動する。

```
% advvis
```

### 8.3. プログラムの初期操作

#### ①解析データセット設定ファイルの指定

メニュー項目「File-Open」を選択して、ダイアログ画面から事前に作成しておいた解析データセット設定ファイル (\*.inp) を選択して、OKボタンを押す（図 8.2）。

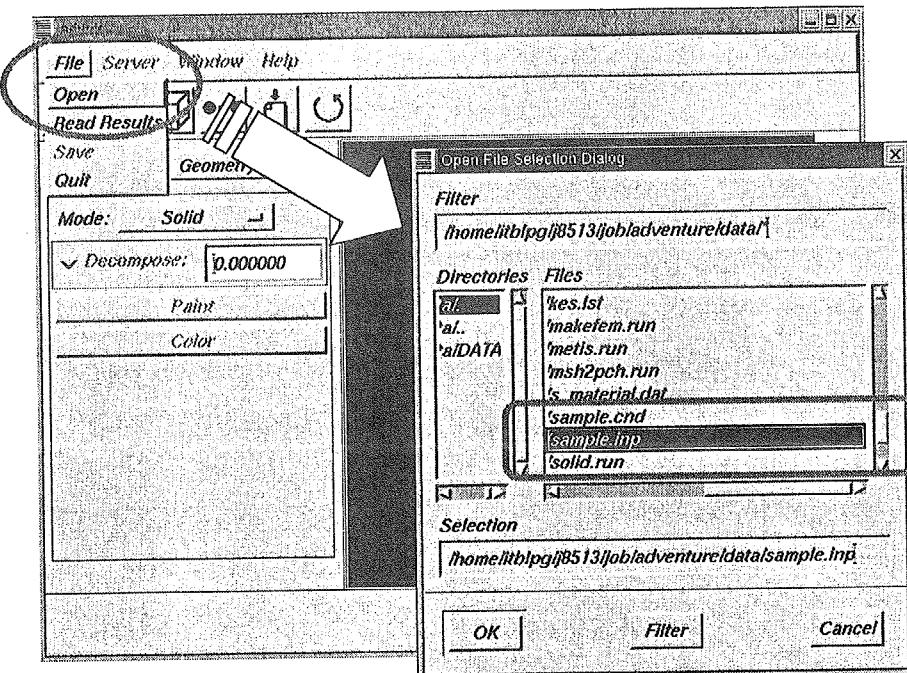


図 8.2 解析データセット設定ファイルの指定

#### ②Server モジュールの設定

メニュー項目「Server-Initialize」を選択して、ダイアログ画面にて Master サーバの指定と Slave サーバの指定を行う（図 8.3）。Slave サーバは、部分数と同じ数のサーバ指定が必要。

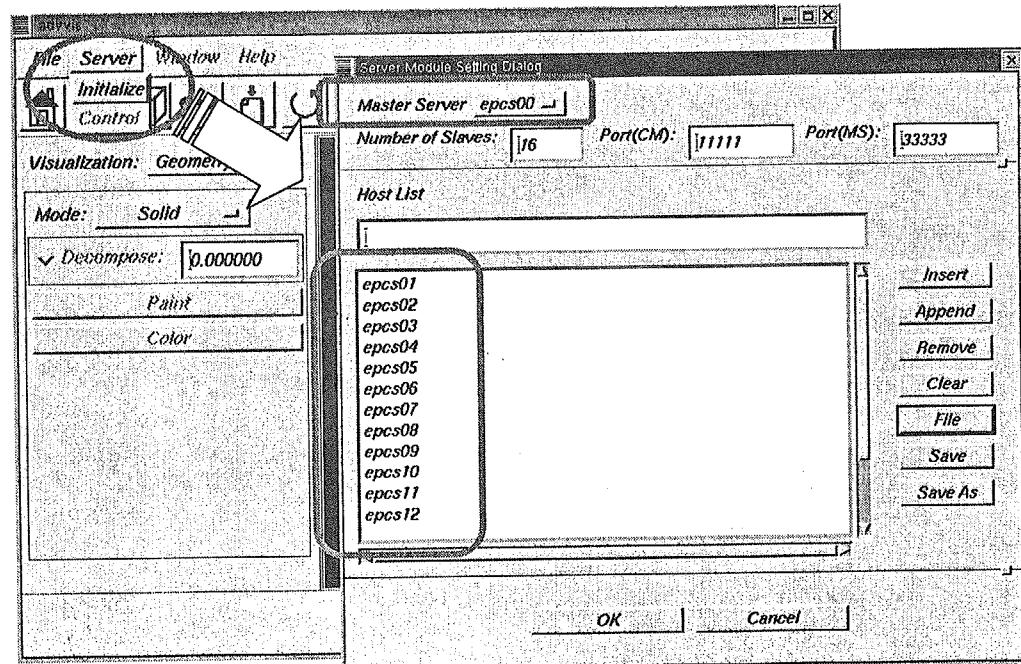


図 8.3 Master サーバと Slave サーバの指定

### ③Server モジュールの起動および接続（図 8.4～図 8.6）

「Server- Control」メニューを選択し、接続設定ダイアログを表示する（図 8.4）。

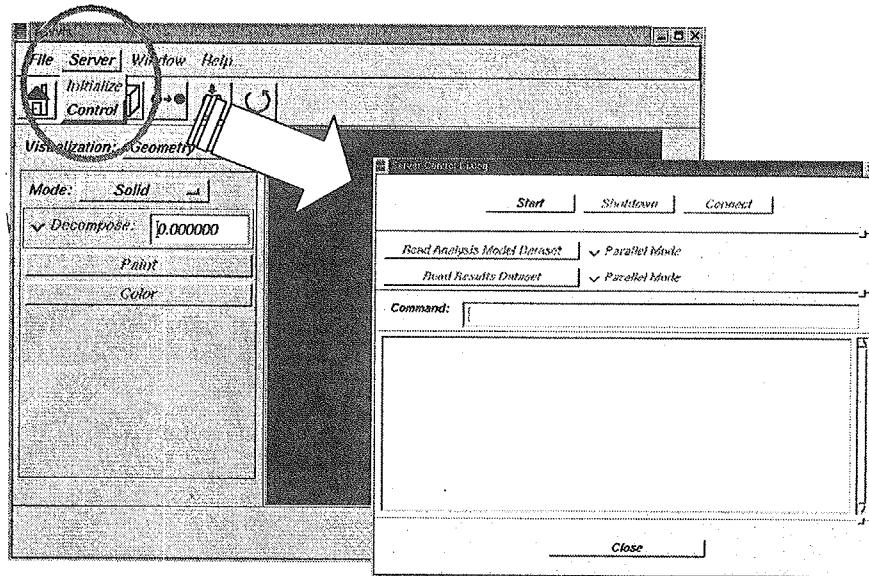


図 8.4 接続設定ダイアログの表示

接続設定ダイアログの「Start」ボタンをクリックする（図 8.5）。

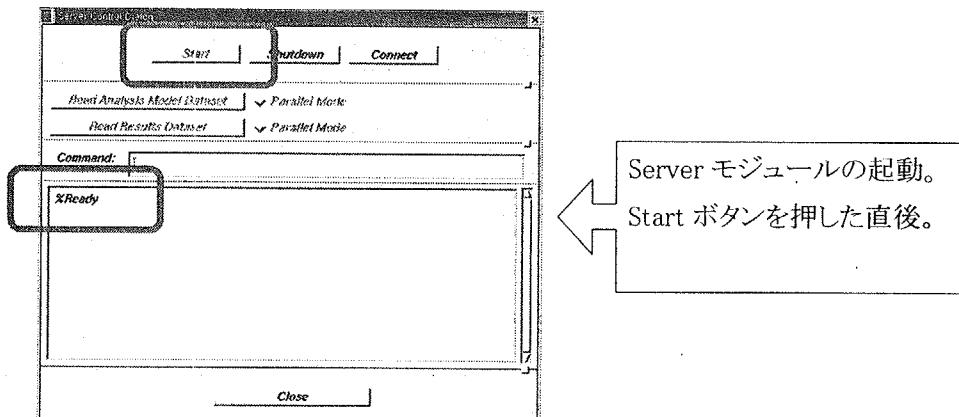


図 8.5 「Start」ボタンクリック直後

接続設定ダイアログの「Connect」ボタンをクリックして Client サーバと接続（図 8.6）。

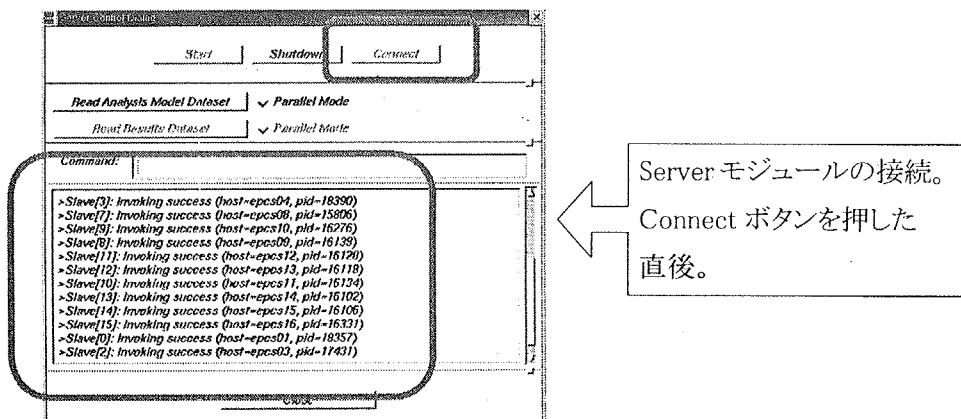


図 8.6 「Connect」ボタンクリック直後

#### ④解析モデルデータセットの読み込み

接続設定ダイアログ画面から Read Analysis Dataset を選択する（図 8.7）。データの入力（取り込み）が完了するとグラフィック表示される。

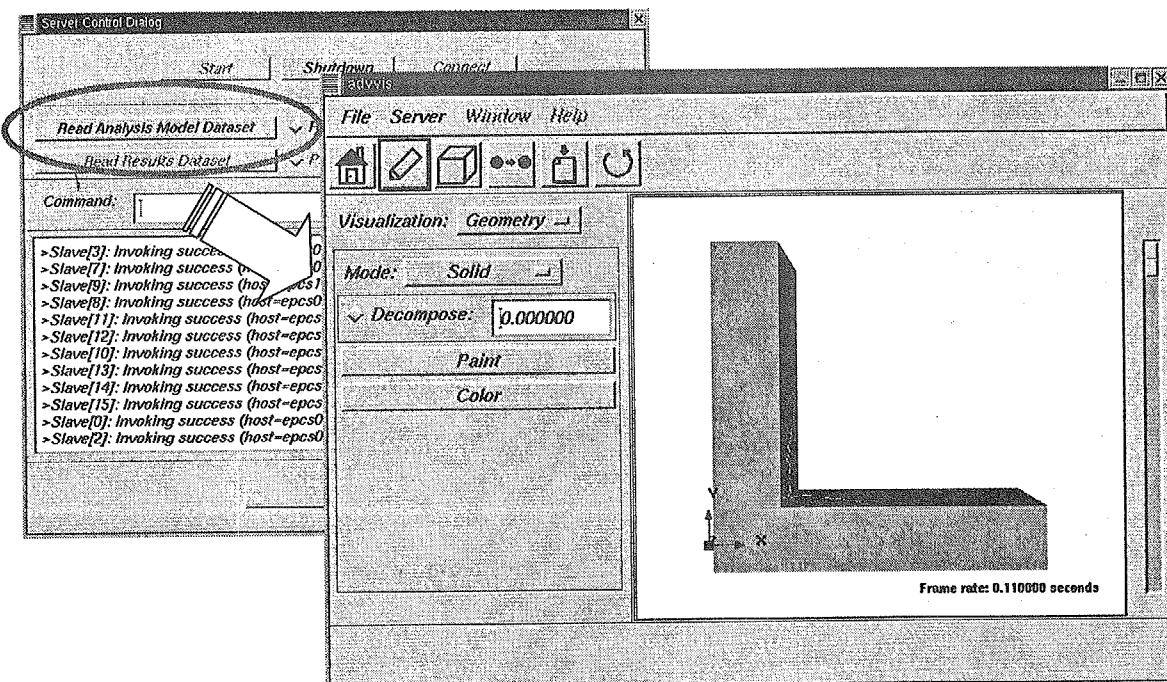


図 8.7 解析モデルの表示

#### ⑤解析結果データセットの読み込み

解析結果データセット入力ダイアログには解析結果の変数が表示される。ここで入力する解析結果の変数を選択して、Load ボタンを押す（図 8.8）。

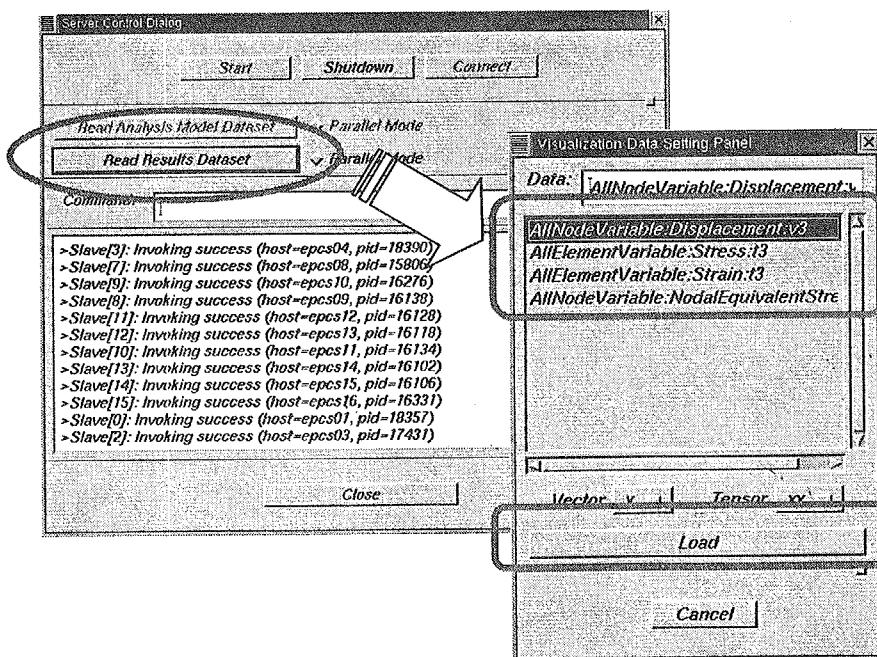


図 8.8 解析結果モデルの表示

## 8.4. プログラムの基本操作

### 8.4.1. ツールボタンの機能 (図 8.9~図 8.11)

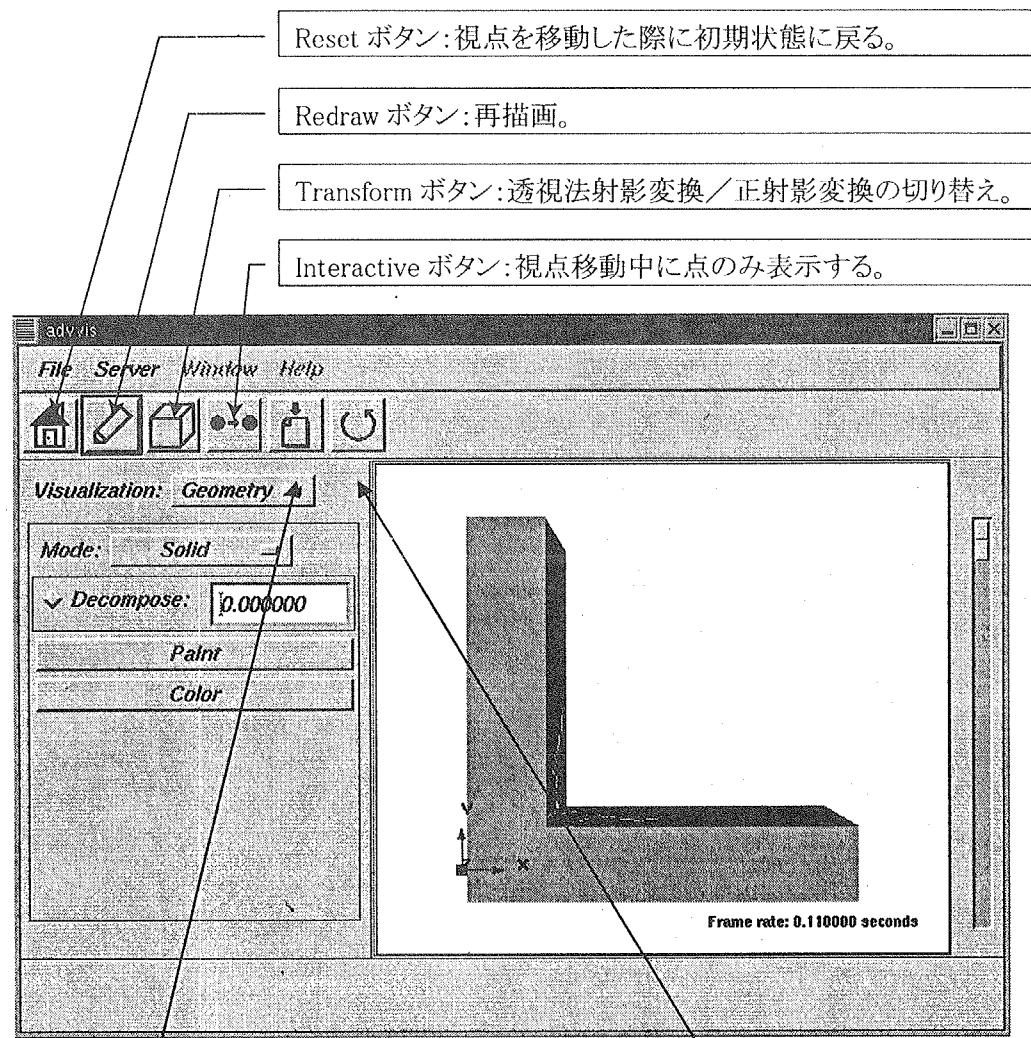


図 8.9 ツールボタンの機能

Save ボタン: 表示内容を画像ファイルとして保存する。

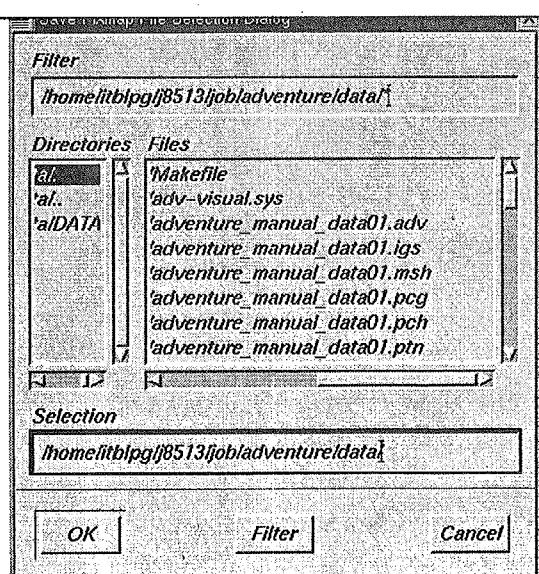


図 8.10 画像ファイル保存

Rotate ボタン: 回転操作を行う。

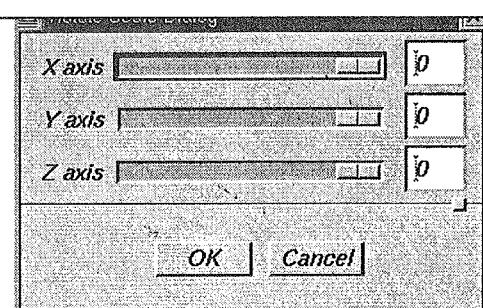


図 8.11 画像回転操作

#### 8.4.2. 視点の移動および拡大表示

視点を移動するにはマウスボタンを押しながらカーソルを動かす。視点の移動とマウスボタンとの対応は表 8.1 のとおり。

表 8.1 視点移動とマウスボタンの対応<sup>8)</sup>

視点の移動	ボタン
左右	Left
上下	Left
前後	Right
物体を中心とした水平方向の回転	Middle
物体を中心とした垂直方向の回転	Middle
両面に垂直な軸周りの回転	Left+Shift キー

### 8.5. 解析モデル形状の表示

表示例を数点掲載するが、詳細は ADVENTURE\_Visual ユーザマニュアルを参照のこと。

#### 8.5.1. ソリッド表示

Mode コンボボックスから「Solid」を選択する（図 8.12）。

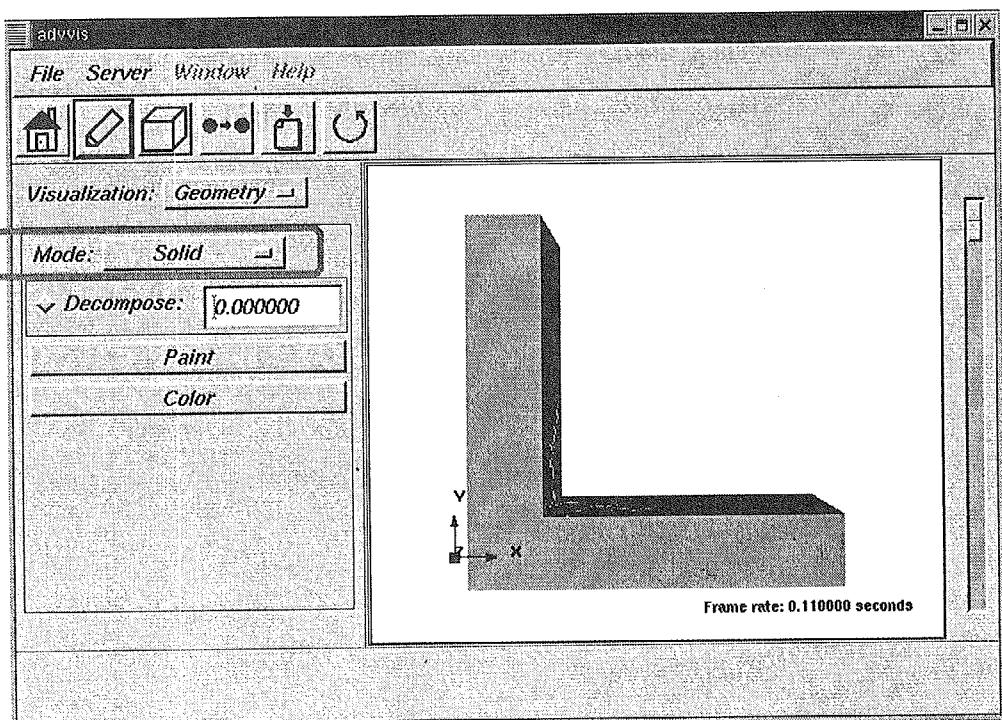


図 8.12 ソリッド表示

### 8.5.2. メッシュ表示

Mode コンボボックスから「Mesh」を選択する（図 8.13）。

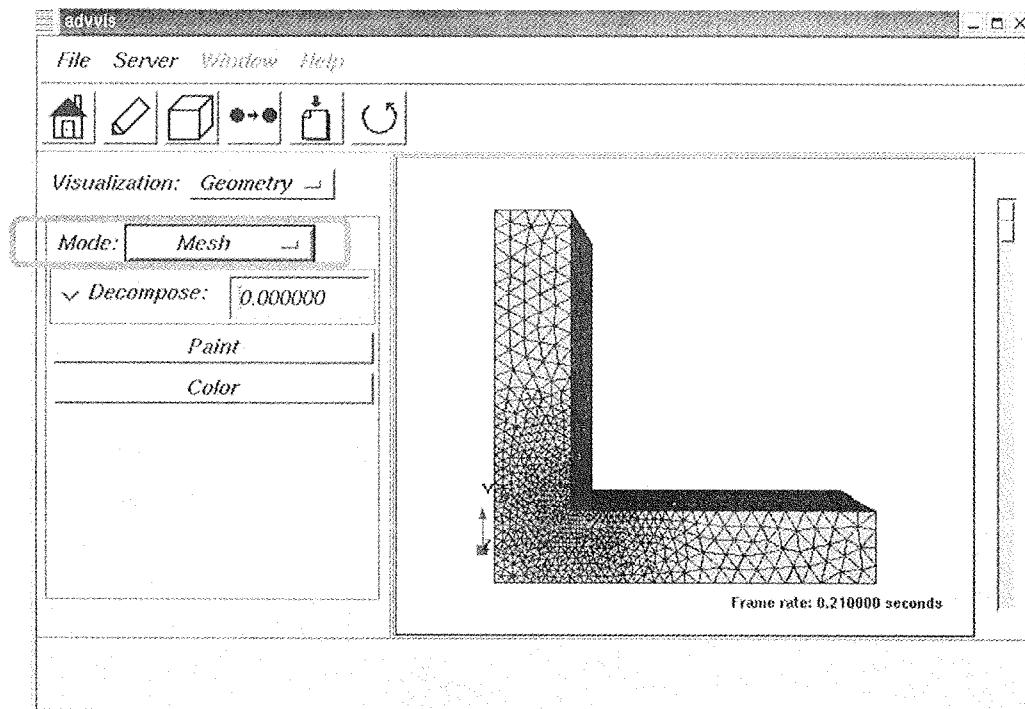


図 8.13 メッシュ表示

### 8.5.3. 領域分割表示

「Paint」ボタンをクリックして色分けにより領域分割を表示する（図 8.14）。

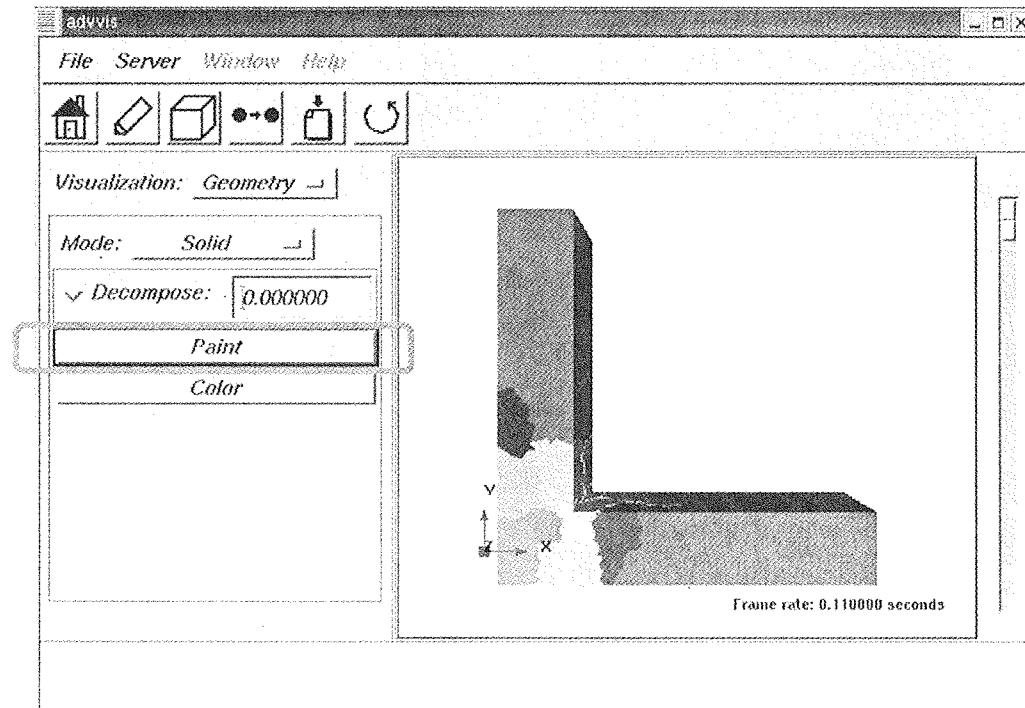


図 8.14 領域分割表示（1）

#### 8.5.4. 領域分割表示 (Decompose factor = 0.1)

領域間の間隔を 0.1 にして表示する (図 8.15)。

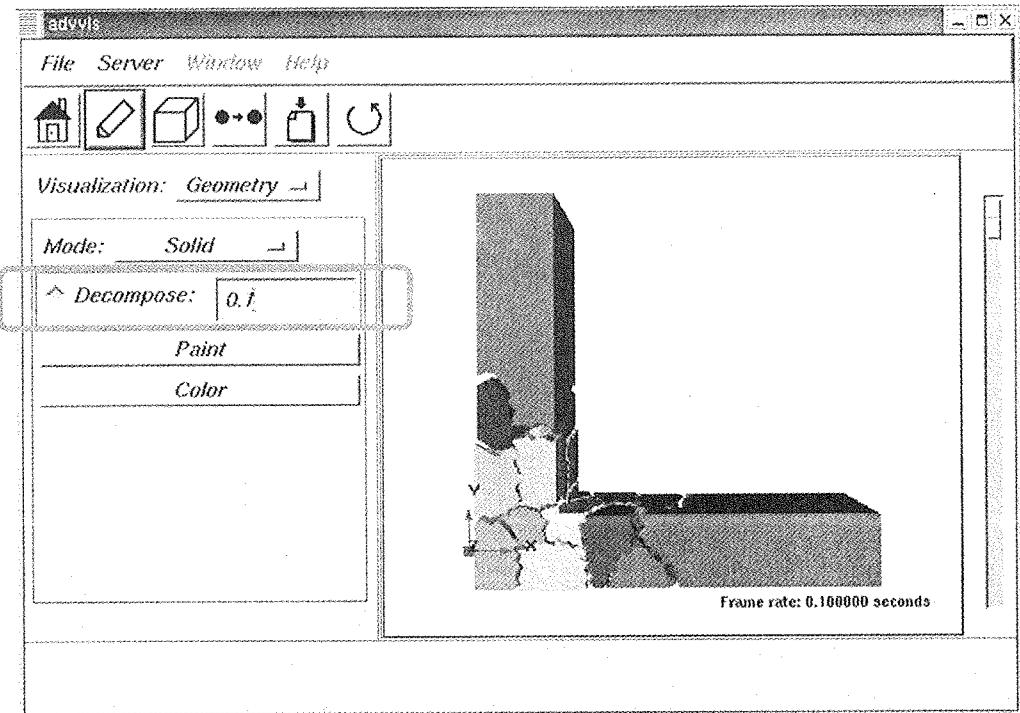


図 8.15 領域分割表示 (2)

#### 8.5.5. 領域分割表示 (Decompose factor=0.5)

領域間の間隔を 0.1 にして表示する (図 8.16)。

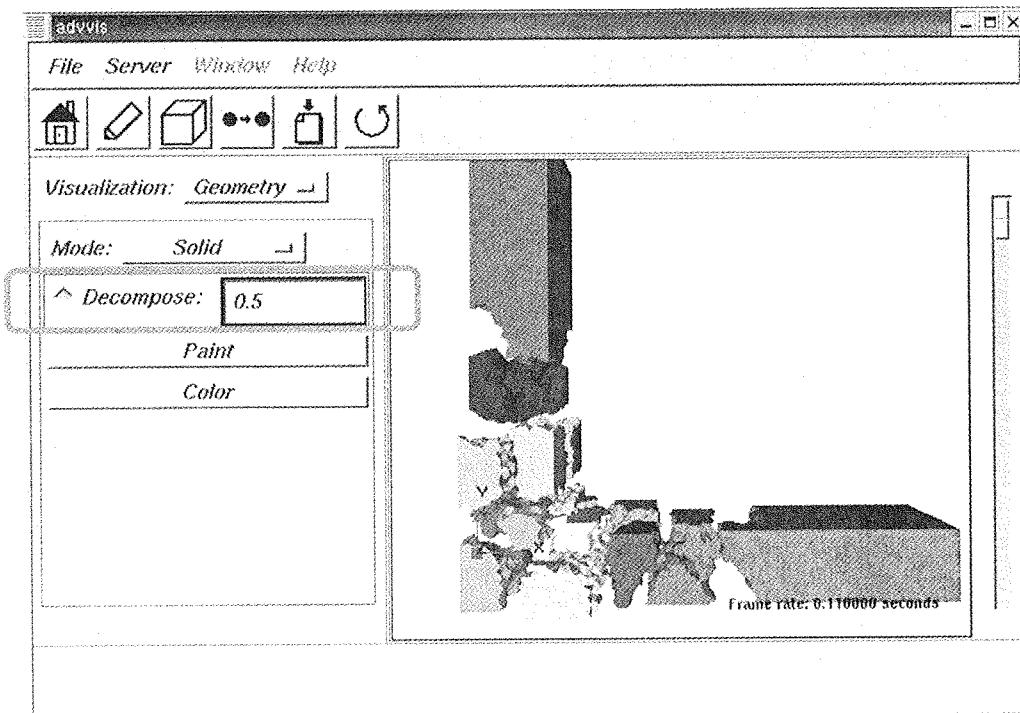


図 8.16 領域分割表示 (3)

### 8.5.6. ワイヤフレーム表示

Mode コンボボックスから「WireFrame」を選択する（図 8.17）。

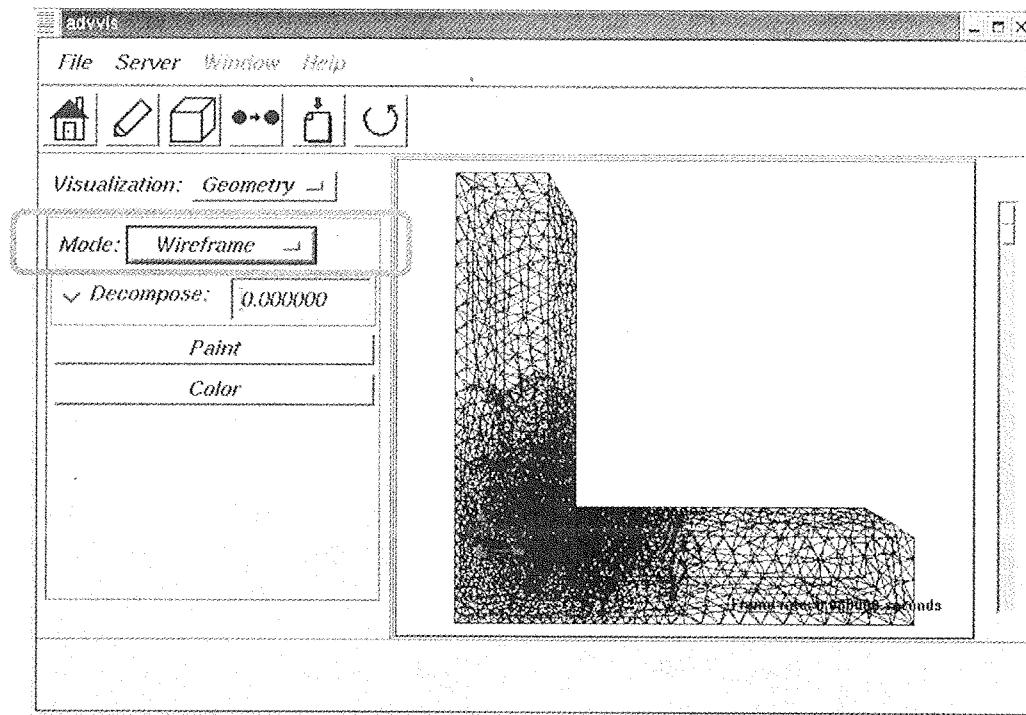


図 8.17 ワイヤフレーム表示

### 8.5.7. バウンディングボックス表示

Mode コンボボックスから「Bounding Box」を選択する（図 8.18）。

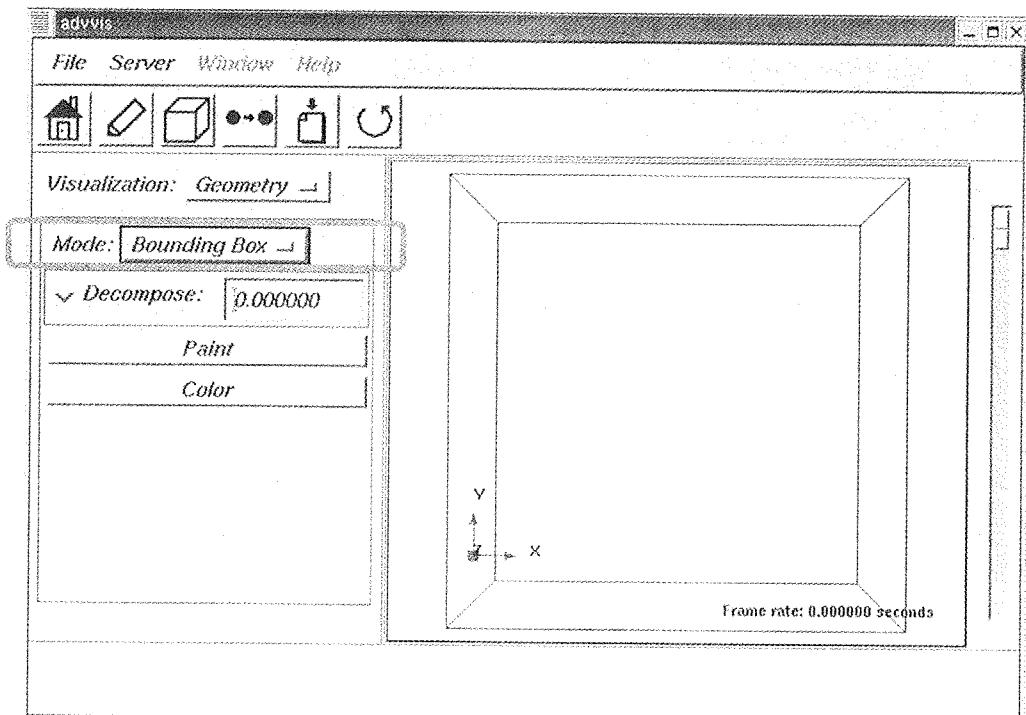


図 8.18 バウンディングボックス表示

### 8.5.8. 表面スカラ分布表示

Mode コンボボックスから「Contour」を選択する（図 8.19）。

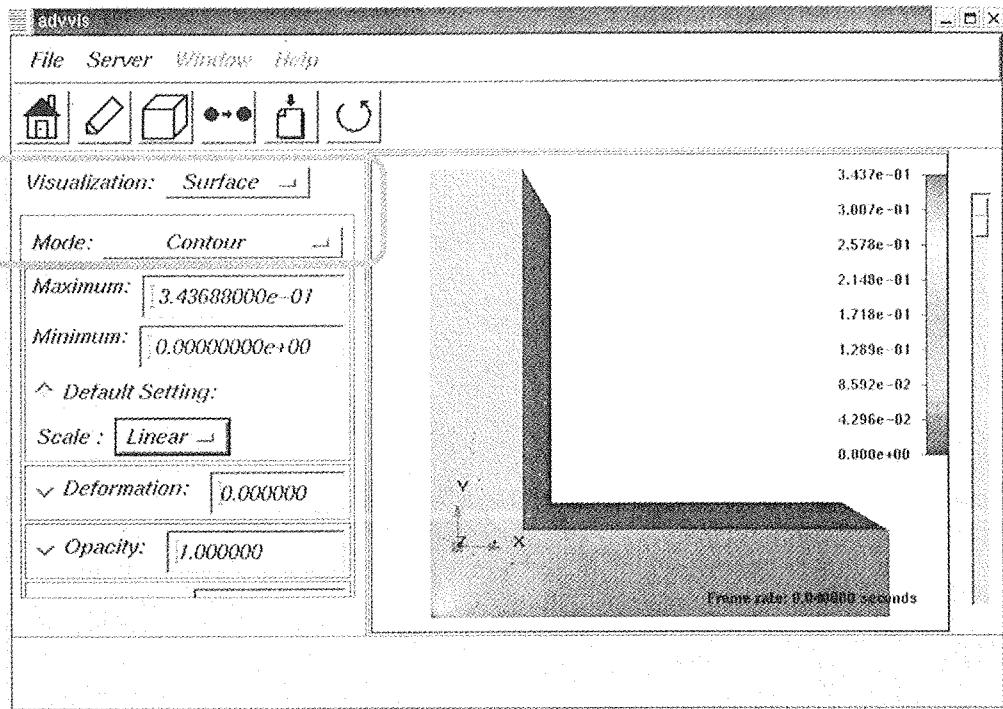


図 8.19 表面スカラ分布表示

### 8.5.9. メッシュ表示

Mode コンボボックスから「Contour and Mesh」を選択する（図 8.20）。

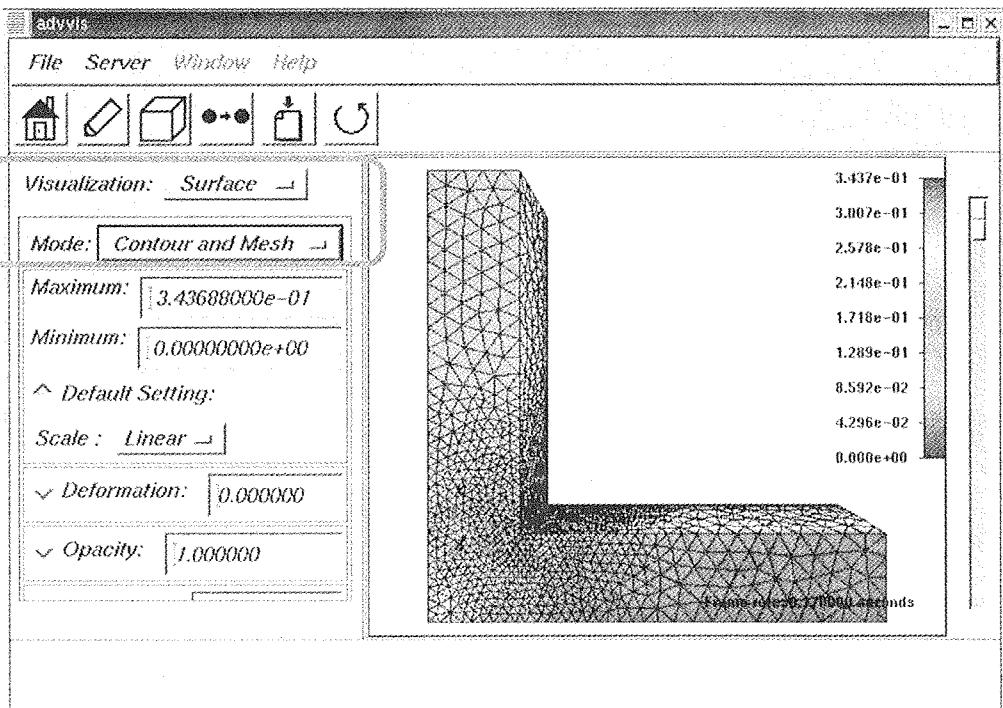


図 8.20 メッシュ表示

### 8.5.10. 変位表示

Mode コンボボックスから「Contour and Vector」を選択する（図 8.21）。

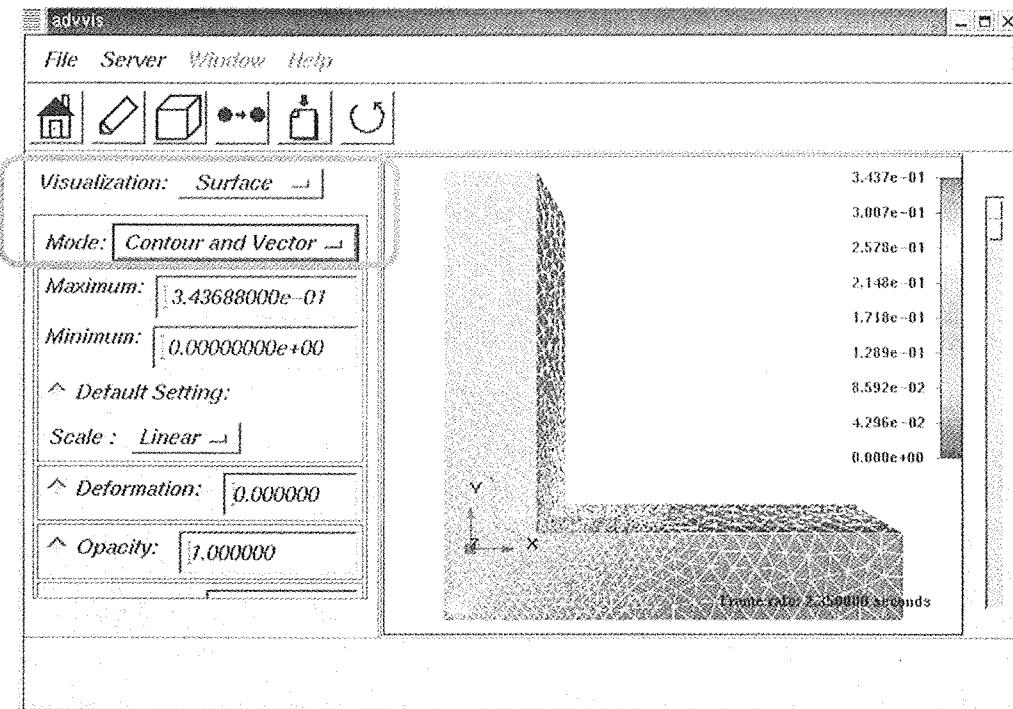


図 8.21 変位表示

### 8.5.11. 断面スカラ分布表示

Visualization コンボボックスから「Cut Plane」を選択する（図 8.22）。

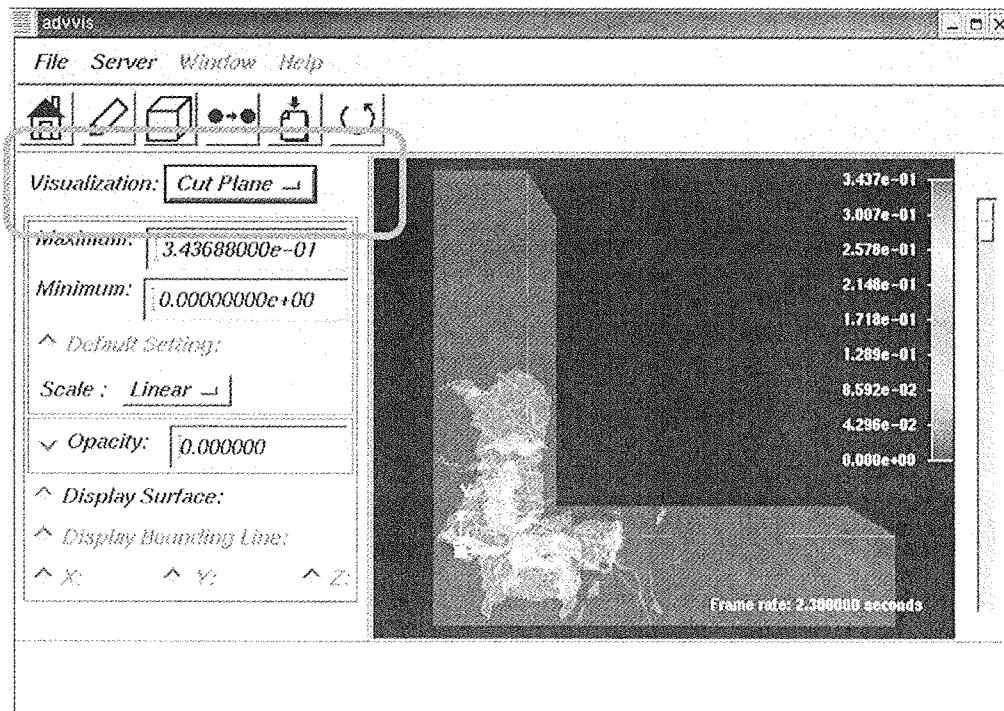


図 8.22 断面スカラ表示

## 9. インストールについて

### 9.1. プログラムの入手方法

ADVENTURE プロジェクト<sup>1)</sup>の以下のサイトよりダウンロードする。

ダウンロードに際しては、ユーザ登録が必要になる。

○ADVENTURE プロジェクトホームページ <http://adventure.q.t.u-tokyo.ac.jp/jp/>

※2002年8月21日現在のVer 1.0のリリース情報

No	モジュール名	バージョン	更新	備考
1	ADVENTURE_Solid	Ver 1.0	2002/03/01	
2	ADVENTURE_Metis	Ver 1.0	2002/03/01	
3	ADVENTURE_IO	Ver 1.0	2002/03/01	
4	ADVENTURE_TriPatch	Ver 1.2	2002/04/18	
5	ADVENTURE_BCtool	Ver 1.0	2002/03/04	
6	ADVENTURE_Visual	Ver 1.0	2002/03/05	
7	ADVENTURE_TetMesh	beta-0.83	2002/03/08	
*	Base Source Package	Ver 1.1	2002/05/08	

※2002年8月21日現在のβ版の情報

No	モジュール名	バージョン	更新	備考
1	ADVENTURE_Fluid	beta		
2	ADVENTURE_Impact	beta-0.82	2002/03/26	
3	ADVENTURE_Magnetic	beta-0.22	2002/04/01	
4	ADVENTURE_Forge	beta-0.4	2002/03/08	
5	ADVENTURE_Thermal	beta-0.4	2002/03/25	
6	ADVENTURE_Shape	beta-0.11	2002/03/19	
7	ADVENTURE_Opt	beta-0.1	2002/03/01	
8	ADVENTURE_CAD	beta-0.1	2002/03/01	
9	ADVENTURE_Auto	beta-0.1	2002/03/01	
10	ADVENTURE_iAgent	beta-0.3	2002/04/05	
11	ADVENTURE_Material	beta-0.8	2002/03/01	

※2002年8月21日現在のユーティリティモジュールの情報

No	モジュール名	バージョン	更新	備考
1	ADVENTURE_FEMAPtool	Ver 1.0	2002/07/25	

## 9.2. インストール時の留意点

- ADVENTURE モジュールのインストール及び実行環境としては、Intel-Linux(Red Hat 7.2)を推奨。
- ADVENTURE の多くのモジュールは、デフォルトで以下にインストールされる。
  - \$HOME/ADVENTURE/bin : 実行モジュール
  - \$HOME/ADVENTURE/doc : ドキュメント関連
  - \$HOME/ADVENTURE/include : ADV\_IO を用いたプログラムのコンパイルに必要なヘッダーファイル
  - \$HOME/ADVENTURE/lib : ADV\_IO を用いたプログラムのコンパイルに必要な静的ライブラリ
- ADV\_IO (ADVENTURE\_IO)からインストールする。  
\$HOME/ADVENTURE/include 及び\$HOME/ADVENTURE/lib は ADV\_IO をインストールした場合のみ作成される。
- コンパイルに必要な開発環境
  - C, C++コンパイラー (gcc 2.95.3 推奨)
  - F90 (g77 不可)
  - make (GNU make 推奨)
  - Motif (Lesstif 可)
  - OpenGL (Mesa 可)
  - MPI (MMPICH, LAM 可)
    - ✧ MPI(Message Passing Interface)は並列計算用の API 規格
    - ✧ MPICH(Portable Implementation of MPI)は Argonne National Lab.によるフリーな MPI 実装系

### 9.3. ADVENTURE\_IO のインストール方法

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvIO-1.0.tar.gz | tar xvf -
```

ADV\_IO のトップディレクトリ (AdvIO-Version) に移り、以下のコマンドを実行する。

```
% cd AdvIO-1.0  
% ./configure  
% make
```

make が正常に終了すると、以下のライブラリファイルができる。

- Base/libAdvBase.a
- DocIO/libAdvDocIO.a
- FileIO/libAdvFileIO.a

ライブラリとヘッダーファイルを所定のディレクトリにインストールする。

```
% make install
```

デフォルトでは、\$HOME/ADVENTURE にインストールされる。インストール先を指定する場合は、先の configure 時にオプション --prefix=install\_dir にて指定する。たとえば、

```
% ./configure --prefix=/usr/local
```

とすれば、make install 時に /usr/local 以下にインストールされる。

configure にはいくつかのオプションがあり、以下のコマンドでその一覧が表示される。

```
% ./configure --help
```

## 9.4. ADVENTURE\_TriPatch のインストール

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvTriPatch-1.2.tar.gz | tar xvf -
```

ADV\_TriPatch のトップディレクトリ (AdvTriPatch-*Version*) に移り、以下のコマンドを実行する。

```
% cd AdvTriPatch-1.2
% ./configure
% make
```

`make` が正常に終了すると、以下のプログラムファイルができる。

- IGES\_V5.3/patch/maskMelon
- tools/mcpach/mcpach
- IGES\_V5.3/patch/ckpatch
- tools/mrpach/mrpach
- IGES\_V5.3/ADVENTURE\_TriPatch

プログラムファイルとドキュメントを所定のディレクトリにインストールする。

```
% make install
```

デフォルトでは、\$HOME/ADVENTURE にインストールされる。インストール先を指定する場合は、先の `configure` 時にオプション `--prefix=install_dir` にて指定する。たとえば、

```
% ./configure --prefix=/usr/local
```

とすれば、`make install` 時に /usr/local 以下にインストールされる。

`configurfe` にはいくつかのオプションがあり、以下のコマンドでその一覧が表示される。

```
% ./configure --help
```

## 9.5. ADVENTURE\_TetMesh のインストール

【注意】2002年8月21日現在、βバージョンのため、将来手順変更の可能性あり。

### 9.5.1. (1) 表面パッチ作成プログラムのインストール

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvTetMesh-0.83b.tar.gz | tar xvf -
```

ADV\_TetMesh の表面パッチ作成プログラムディレクトリ (*AdvTetMesh-Version/TetMesh\_p*) に移り、以下のコマンドを実行する。

```
% cd AdvTetMesh-0.83b/TetMesh_P
% vi Makefile (Makefile を編集)
```

コンパイラ（フラグ）、リンクフラグを変更する場合、以下の内容を編集する。

```
F90 = f90           # --> pgf90 ...
FFLAGS = -fast
LDFLAGS = -non_shared # --> nothing
```

`make` を実行する。

```
% make
```

`make` が正常に終了すると、以下のプログラムファイルができる。

➤ `AdvTetMesh-0.8b/bin/advtmesh8p`

プログラムファイルを所定のディレクトリにインストールする。

```
% make install
```

インストール先のディレクトリはデフォルトでは\$HOME/ADVENTURE/bin であるが、別のディレクトリにしたい場合は、`Makefile` の `PREFIX` を便宜変更してから `make install` する。

### 9.5.2. (2) 四面体メッシュ作成プログラムのインストール

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

※ 上記(1)で既に展開している場合は、不要。

```
% gunzip -c AdvTetMesh-0.83b.tar.gz | tar xvf -
```

ADV\_TetMesh の四面体メッシュ作成プログラムディレクトリ (*AdvTetMesh·Version/TetMesh\_M*) に移り、以下のコマンドを実行する。

```
% cd AdvTetMesh-0.83b/TetMesh_M
% vi Makefile  (Makefile を編集)
```

コンパイラ（フラグ）、リンクフラグを変更する場合、以下の内容を編集する。

```
CXX = cxx
LDFLAGS = -non_shared
CXXFLAGS = -wall -O
```

make を実行する。

```
% make
```

make が正常に終了すると、以下のプログラムファイルができる。

➤ AdvTetMesh-0.8b/bin/advtmesh8m

プログラムファイルを所定のディレクトリにインストールする。

```
% make install
```

インストール先のディレクトリはデフォルトでは\$HOME/ADVENTURE/bin であるが、別のディレクトリにしたい場合は、Makefile の PREFIX を便宜変更してから make install する。

## 9.6. ADVENTURE\_BCtool のインストール

### 【構成】

ADVENTURE\_BCtool は、以下の 7 つのモジュールで構成されている。

bcGUI	; C++, Motif(or LessTif), OpenGL(or Mesa)が必要
faceOfMesh	; C++が必要
makefem	; C++, ADVENTURE_IO が必要
makepch	; C++が必要
makepcm	; C++が必要
msh2pch	; シェルスクリプト
msh2pcm	; シェルスクリプト

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvBCtool-1.0.tar.gz | tar xvf -
```

Adv\_BCtool-1.0/Makefile を環境に合わせて編集する。

```
% cd AdvBCtool-1.0
% vi Makefile (Makefile を編集)
```

### ◇変更箇所 1：インストールディレクトリに関して

ADVIO_PREFIX # ADVENTURE_IO をインストールしたディレクトリ
PREFIX # ADVENTURE_BCtool をインストールするディレクトリ

### ◇変更箇所 2：コンパイル、リンクに関して

CXX # C++コンパイラ
CXXFLAGS # コンパイルオプション
GL_INC # OpenGL (または Mesa) のインクルードファイルの指定
GL_LIB # OpenGL (または Mesa) のライブラリの指定
X11_INC # X11 のインクルードファイルの指定
X11_LIB # X11 のライブラリの指定
MOTIF_INC # Motif (または LessTif) のインクルードファイルの指定
MOTIF_LIB # Motif (または LessTif) のライブラリの指定

◇変更箇所 3 : OpenGL について

```
GLWFLAGS      # bcGUI 用のオプション  
GLW          # GLW ウィジェットのライブラリの指定
```

OpenGL の代わりに Mesa を使用する場合

```
GLWFLAGS      = -DGLW_HDR_ANOTHER_LOCATION -DnoMotifGLwidget  
GLW          = -lMesaGLw
```

OpenGL を使用する場合

```
GLWFLAGS      =  
GLW          = -lGLW
```

Makefile の編集後、以下のコマンドを実行する。

```
% make  
% make install
```

インストール先のディレクトリはデフォルトでは\$HOME/ADVENTURE/bin であるが、別のディレクトリにしたい場合は、Makefile の PREFIX を便宜変更してから make install する。

## 9.7. ADVENTURE\_Metis のインストール

### 【準備】

あらかじめ、MPI コンパイル環境と ADVENTURE\_IO モジュールをコンパイルしておくこと。

MPI がインストールされていない環境では、フリーの MPICH 等を利用する。

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvMetis-1.0.tar.gz | tar xvf -
```

ADV\_Metis のトップディレクトリ (AdvMetis-*Version*) に移り、以下のコマンドを実行する。

```
% cd AdvMetis-1.0
% ./configure
% make
```

### 【注意】

以下に `configure` に関して、吉村先生（東大院、新領域）のコメントが詳しかったのでそのまま記載する。

`configure` は環境に依存する部分を解決し、適切な Makefile を作成するためのシェルスクリプトである。`configure` に渡せる主なオプションを以下に示す。ただし、以下で用いるディレクトリ名には絶対パスを指定すること。

- `--prefix=install_dir`  
インストール先のトップディレクトリを `install_dir` にする。デフォルトは `$HOME/ADVENTURE` である。
- `--with-advio=install_dir`  
ADVENTURE\_IO が上記で指定する（あるいはデフォルトの）`install_dir` 以外にインストールされている場合に、そのインストール先ディレクトリを指定する。
- `--with-mpicc=command`  
MPI C コンパイラ名を示す。デフォルトは `mpicc` である。ADVENTURE\_Metis のソースのコンパイルには、すべてこれを用いる。見つからない場合は、コンパイルできない。
- `--with-mpi-cflags=CFLAGS`  
MPI プログラムをコンパイルする場合必要な C のコンパイルオプションを指定する。たとえば MPI のインクルードファイル（ディレクトリ）を指定する必要がある場合は、  
`--with-mpi-cflags="-I/usr/local/include/mpi"`などと指定する。ログインシェル(たとえば C シェル) の環境変数とあわせて用いることができる。

- **--with-mpi-libs=LIBS**

プログラムをリンクする場合必要なオプションを指定する。たとえば MPI のライブラリを明示する必要がある場合は、`--with-mpi-libs="-L/usr/local/lib/mpi -lmpi"`などと指定する。ログインシェルの環境変数とあわせて用いる事ができる。

- **--enable-optimize**

コンパイル時に最適化を行う。これにより設定されるオプション以外の最適化オプションをつけてコンパイルしたい場合は、下記の書式を使用する。

- **--enable-optimize=CFLAGS**

`CFLAGS` を最適化用のオプションとして、コンパイル時に最適化を行う。

また、コンパイルに使用される MPI C コンパイラを変更するには、以下の環境変数が使用できる。

- **MPICC**

MPI C コンパイラ名を設定する。

- **CFLAGS**

MPI C コンパイラに対するオプションを設定する。

- **LIBS**

リンクする必要のあるライブラリが他にあれば指定する。

これらは、`./configure` 実行前に`./configure` を実行するシェル中にて設定しておく。

たとえば、C シェルの場合、

```
% setenv MPICC /usr/local/bin/mpicc
% setenv CFLAGS "-O2 -g -Wall"
% ./configure
```

等とし、Bourne シェルの場合、

```
$ MPICC=/usr/local/bin/mpicc
$ export MPICC
$ CFLAGS="-O2 -g -Wall"
$ export CFLAGS
$ ./configure
```

等として設定する。

`configurfe` のオプションは、以下のコマンドでその一覧が表示される。

```
% ./configure --help
```

`configure` スクリプトを使用することで、多くの環境ではコンパイル可能と思われるが、うまくいかない場合には、`Makefile` のサンプルが各ディレクトリに用意してあるのでそれを用いてコンパイルする。それぞれのディレクトリにて `Makefile.sample` を `Makefile` にコピーする。また、トップディレクトリにある `Makefile.sample.in` を各環境に合わせて編集してトップディレクトリにて `make` を実行する。

コンパイル終了後、以下のコマンドで、所定のディレクトリにインストールする。  
configure を用いず make した場合は、各ディレクトリにて実行する。

```
% make install
```

デフォルトでは、\$HOME/ADVENTURE にインストールされる。インストール先を指定する場合は、先の configure 時にオプション --prefix=*install\_dir* にて指定する。

以下の、プログラムとドキュメントがインストールされる。

- bin/adventure\_metris ADVENTURE\_Metis バイナリ
- doc/AdvMetris/manual-jp.pdf  
                  日本語マニュアル
- doc/AdvMetris/README, README.eucJP, copyright  
                  テキスト形式のドキュメント

## 9.8. ADVENTURE\_Solid のインストール

### 【準備】

あらかじめ、MPI コンパイル環境と ADVENTURE\_IO モジュールをコンパイルしておくこと。

MPI がインストールされていない環境では、フリーの MPICH 等を利用する。ただし、シングル版のみを使用する場合は、MPI がなくてもコンパイル、実行することが可能である。また、簡単なログ解析用ツールが perl で記述されているため、必須ではないが perl もインストールされているのが望ましい。

以下のコマンドで、ダウンロードした tar ファイルのパッケージを開く。

```
% gunzip -c AdvSolid-1.0.tar.gz | tar xvf -
```

ADV\_Solid のトップディレクトリ (AdvSolid-*Version*) に移り、以下のコマンドを実行する。

```
% cd AdvSolid-1.0
% ./configure
% make
```

### 【注意】

以下に `configure` に関して、吉村先生（東大院、新領域）のコメントが詳しきったのでそのまま記載する。

`configure` は環境に依存する部分を解決し、適切な Makefile を作成するためのシェルスクリプトである。`configure` に渡せる主なオプションを以下に示す。ただし、以下で用いるディレクトリ名には絶対パスを指定すること。

- `--prefix=install_dir`  
インストール先のトップディレクトリを `install_dir` にする。デフォルトは `$HOME/ADVENTURE` である。
- `--with-advio=install_dir`  
ADVENTURE\_IO が上記で指定する（あるいはデフォルトの）`install_dir` 以外にインストールされている場合に、そのインストール先ディレクトリを指定する。
- `--with-mpicc=command`  
MPI プログラム用の C コンパイラ名を示す。デフォルトは `mpicc` である。見つからない場合は、並列版のコンパイルは行わない。
- `--with-mpi-cflags=CFLAGS`  
MPI プログラムをコンパイルする場合必要な C のコンパイルオプションを指定する。例えば MPI のインクルードファイル（ディレクトリ）を指定する必要がある場合は、`--with-mpi-cflags="-I/usr/local/include/mpi"` などと指定する。MPI プログラムのコンパイル時には、ここで指定したものに加えシングル版用に設定されて

いるものの両方が用いられる。

- **--with-mpi-libs=LIBS**

MPI プログラムをリンクする場合必要なオプションを指定する。たとえば MPI のライブラリを明示する必要がある場合は、**--with-mpi-libs = "-L/usr/local/lib/mpi -lmpi"**などと指定する。MPI プログラムのリンク時には、ここで指定したものに加え、シングル版用に設定されているものの両方が用いられる。

- **--enable-optimize**

コンパイル時に最適化を行う。これにより設定されるオプション以外の最適化オプションをつけてコンパイルしたい場合は、下記の書式を使用する。

- **--enable-optimize=CFLAGS**

**CFLAGS** を最適化用のオプションとして、コンパイル時に最適化を行う。

また、シングル版や、シングル・並列版共通部分のコンパイルに使用される C コンパイラを変更するには、以下の環境変数が使用できる。

また、コンパイルに使用される MPI C コンパイラを変更するには、以下の環境変数が使用できる。

- **CC**

C コンパイラ名を設定する。

- **CFLAGS**

C コンパイラに対するオプションを設定する。

- **LIBS**

リンクする必要のあるライブラリが他にあれば指定する。

これらは、**./configure** 実行前に**./configure** を実行するシェル中にて設定しておく。

たとえば、C シェルの場合、

```
% setenv CC /usr/local/bin/cc
% setenv CFLAGS "-O2 -g -Wall"
% ./configure
```

等とし、Bourne シェルの場合、

```
$ CC=/usr/local/bin/mpicc
$ export CC
$ CFLAGS="-O2 -g -Wall"
$ export CFLAGS
$ ./configure
```

等として設定する。

**configurfe** のオプションは、以下のコマンドでその一覧が表示される。

```
% ./configure --help
```

`configure` スクリプトを使用することで、多くの環境ではコンパイル可能と思われるが、うまくいかない場合には、`Makefile` のサンプルが各ディレクトリに用意してあるのでそれを用いてコンパイルする。それぞれのディレクトリにて `Makefile.sample` を `Makefile` にコピーする。また、トップディレクトリにある `Makefile.sample.in` を各環境に合わせて編集してトップディレクトリにて `make` を実行する。

ただし、`solver/` の下をコンパイルする前に `libfem/` の下をコンパイルしておく必要がある。

コンパイル終了後、以下のコマンドで、所定のディレクトリにインストールする。

`configure` を用いず `make` した場合は、`libfem` を除く各ディレクトリにて実行する。

```
% make install
```

デフォルトでは、`$HOME/ADVENTURE` にインストールされる。インストール先を指定する場合は、先の `configure` 時にオプション `--prefix=install_dir` にて指定する。

以下の、プログラムとドキュメントがインストールされる。

- `bin/advsolid`      ADVENTURE\_Solid 実行スクリプト
- `bin/advsolid-s`    シングル版 ADVENTURE\_Solid
- `bin/advsolid-p`    並列・静的負荷分散 ADVENTURE\_Solid
- `bin/advsolid-h`    並列・動的負荷分散 ADVENTURE\_Solid
- `bin/log2cnv-cg, log2cnv-nr, log2info`                            ADVENTURE\_Solid のログファイル解析スクリプト
- `bin/advshow`        ADVENTURE\_Format File のテキスト化ツール
- `bin/hddmmrg`        解析結果データの一体型へのマージツール
- `doc/AdvSolid/manual-jp.pdf`                                    日本語マニュアル
- `doc/AdvSolid/README, README.eucJP, eucJP, copyright`    テキスト形式のドキュメント

## 9.9. ADVENTURE\_Visual のインストール

【参考：動作環境】

- オペレーティングシステム  
Unix, Linux, FreeBSD
- X Window システム (XFree86 を含む)
- ハードウェア  
Server モジュール: PC クラスタ  
Client モジュール: OpenGL 対応グラフィックスカード推奨
- コンパイラ及び開発ツール  
`g++` (gcc 2.95.1, または 2.95.3)  
`GNU make`
- 必要なライブラリ
  - (1) グラフィックス・ライブラリ  
OpenGL (または Mesa), Motif (または LessTif)  
Mesa: <http://www.mesa3d.org/>  
LessTif: <http://www.lesstif.org/>  
が利用できる。
  - (2) ADVENTURE\_IO ライブラリ

以下のコマンドで、ダウンロードした tar ファイルのパッケージを展開する。

```
% gunzip -c AdvVisual-1.0.tar.gz | tar xvf -
```

展開した AdvVisual-1.0 の構成は以下のとおり。

サブディレクトリ名	内容
client	Client モジュール用ソースファイル
server	Server モジュール用ソースファイル
doc	ドキュメント類
examples	サンプルデータ類

#### ◇Server モジュールのコンパイル

ADV\_Visual のトップディレクトリ (AdvVisual-Version) 下の Server ディレクトリに移り、以下のコマンドを実行する。

```
% cd Advvisual-1.0
% cd server
% ./configure --with-advio=<ADVENTURE_IO をインストールしたディレクト
リ ${HOME}/ADVENTURE/bin>
% make
```

インストール。

**make install** 未整備のため、\${HOME}ADVENTURE/bin に単純コピー

#### ◇Client モジュールのコンパイル

ADV\_Visual のトップディレクトリ (AdvVisual-Version) 下の Client ディレクトリに移り、以下のコマンドを実行する。

```
% cd Advvisual-1.0
% cd client
% vi Makefile      (Makefile の編集)
```

Makefile 中、以下のマクロを編集する。

```
INCDIR : Motif および OpenGL 用 include ファイルのディレクトリ
LIBDIR : Motif および OpenGL 用 library ファイルのディレクトリ
LIBS : Motif および OpenGL 用 library ファイル
```

Makefile 編集後、以下のコマンドを実行する。

**% make**

インストール。

**make install** 未整備のため、\${HOME}ADVENTURE/bin に単純コピー

## 9.10. MPICH のインストール

### ■バイナリパッケージからインストールする

RedHat や Debian などのディストリビューションではコンパイル済みのパッケージが用意されているので、それをインストールする。

### ■ソースからコンパイルする

```
% gunzip -c mpich.tar.gz | tar xvf -
% cd mpich-1.2.1
% ./configure --prefix=/usr/local/mpich
% make
# make install
```

### □使い方

#### ■コンパイル

```
% mpicc [option] program.c
```

#### ■実行方法

```
% mpirun [-np nproc] [-machinefile machinefile] program [option]
```

## 10. おわりに

本書では、ADVENTURE システムのベーシックモジュール（Ver1.0）を中心に、基本的な操作方法、実行方法について述べた。動作確認に用いたサンプルデータは、シンプルなもので ADVENTURE プロジェクトのホームページからダウンロードしたソースモジュールにパックされている。ここで紹介した 7 つの基本モジュール以外に、現在 12 個のモジュールが  $\alpha$  版として公開されている。今後引き続き、ITBL 環境におけるコミュニティソフトとして整備を進めていく方針である。また、今回は、調査・試験的意味もあり PC クラスターへの移植に留まったが、現在ベーシックモジュールを ITBL 計算機に移植作業中である。ITBL 環境に移植後は、大規模超並列解析に挑戦し、ITBL 活用の有益性を実証していく考えである。

ITBL 環境において、基盤ソフトやネットワーク技術、並列処理技術の重要性は言うまでもないが、応用ソフトの整備やその利用・普及推進等の活動も同様に重要課題である。また、有限要素解析システムのような数値シミュレーションでは、解析結果の可視化環境も大変重要であり、ITBL 環境における可視化環境整備も合わせて実施していく方針である。

## 謝辞

本報告書の執筆の機会を与えて下さった ITBL 利用推進室の相川室長、ITBL 利用推進室の皆様に感謝いたします。また、ITBL と ADVENTURE システムの環境構築にいろいろご協力と助言をいただきました情報システム管理課の皆様、ADVENTURE プロジェクトの皆様に感謝いたします。

## 参考文献

- 1) ADVENTURE プロジェクトホームページ <http://adventure.q.t.u-tokyo.ac.jp/jp/>
- 2) ADVENTURE システム講習会資料 (2001/5/15)  
(ADVENTURE プロジェクト、(株)アライドエンジニアリング)
- 3) ADVENTURE\_TriPatch プログラム使用マニュアル (ADVENTURE プロジェクト)
- 4) ADVENTURE\_TetMesh プログラム使用マニュアル (ADVENTURE プロジェクト)
- 5) ADVENTURE\_BCtool プログラム使用マニュアル (ADVENTURE プロジェクト)
- 6) ADVENTURE\_Mmetis プログラム使用マニュアル (ADVENTURE プロジェクト)
- 7) ADVENTURE\_Solid プログラム使用マニュアル (ADVENTURE プロジェクト)
- 8) ADVENTURE\_Visual プログラム使用マニュアル (ADVENTURE プロジェクト)
- 9) ADVENTURE\_IO プログラム使用マニュアル (ADVENTURE プロジェクト)

# 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	$\Omega$	$V/A$
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	°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	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	$\mu$
$10^{-9}$	ナノ	n
$10^{-12}$	ピコ	p
$10^{-15}$	フェムト	f
$10^{-18}$	アト	a

(注)

1. 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、1eVおよび1uの値はCODATAの1986年推奨値によった。

2. 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。

3. barは、JISでは流体の圧力を表す場合に限り表2のカテゴリーに分類されている。

4. EC閣僚理事会指令ではbar、barnおよび「血圧の単位」mmHgを表2のカテゴリーに入れている。

## 換 算 表

力	N(=10 <sup>5</sup> dyn)	kgf	lbf
1	0.101972	0.224809	
9.80665	1	2.20462	
4.44822	0.453592	1	

粘度 1 Pa·s(N·s/m<sup>2</sup>) = 10 P(ポアズ)(g/(cm·s))

動粘度 1 m<sup>2</sup>/s = 10<sup>4</sup>St(ストークス)(cm<sup>2</sup>/s)

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

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

放射能	Bq	Ci	吸収線量	Gy	rad	照射線量	C/kg	R	線量当量	Sv	rem
1	$2.70270 \times 10^{-11}$	1	1	100	1	3876					
$3.7 \times 10^{10}$	1	0.01	1	$2.58 \times 10^{-4}$	1	1					

(86年12月26日現在)

R100

古紙配合率100%  
白色度70%再生紙を使用しています。