

JAERI-Data/Code
2002-016



JP0250403



PATRAS の地球シミュレータ用ソフトウェア
CHIKAKU DYNAMIC への適用・評価

2002 年 9 月

小林 穰

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。

入手の問い合わせは、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、お申し越し下さい。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行っております。

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.

© Japan Atomic Energy Research Institute, 2002

編集兼発行 日本原子力研究所

PATRAS の地球シミュレータ用ソフトウェア CHIKAKU DYNAMIC への適用・評価

日本原子力研究所計算科学技術推進センター

小林 穰

(2002年7月1日 受理)

日本原子力研究所計算科学技術推進センターが開発してきた PATRAS の非構造六面体格子の機能を理化学研究所が開発中の地球シミュレータ向けソフトウェアである CHIKAKU SYSTEM に適用した。PATRAS を CHIKAKU DYNAMIC に適用するにあたり、分離要素に対応するとともに、あらたに 2 次元テンソル量表示機能とカラーバー表示機能を追加した。新機能に対応した PATRAS を CHIKAKU DYNAMIC に組み込んだ実時間可視化版を開発した。実時間可視化版には、AVS の UCD 形式のデータを出力する機能も開発した。この可視化用データを基に、ポスト可視化用コードもあわせて開発した。開発した実時間版およびポスト可視化版コードを使用し、分離要素対応と 2 次元テンソル量表示が CHIKAKU DYNAMIC の解析結果を可視化する上で有効であることを確認した。あわせて、原研内の 3 種類の並列計算機で可視化速度を測定し、性能を評価した。

Application to the Software for Earth Simulator, CHIKAKU DYNAMIC and Evaluation
using PATRAS

Minoru KOBAYASHI

Center for Promotion of Computational Science and Engineering
Japan Atomic Energy Research Institute
Higashiueno, Taito-ku, Tokyo

(Received July 1, 2002)

Center for Promotion of Computational Science and Engineering in Japan Atomic Energy Research Institute has been developing PATRAS, and a function for structured hexahedral grid of the PATRAS was adapted to CHIKAKU DYNAMIC, the software for Earth Simulator developed by Institute of Physical and Chemical Research(RIKEN). To adapt to CHIKAKU DYNAMIC, it is now available for separated elements and has two new functions of 2-dimensional tensor display and color bar display. The new version of PATRAS for real time visualization system was developed with the new functions. The new version can output UCD data format of AVS. Post visualization codes were also developed with using the UCD data. It was tested and confirmed that the function for separated elements and display of 2-dimentional tensor are effective to visualize the results of CHIKAKU DYNAMIC. The performance of visualization on three parallel computers in the Japan Atomic Energy Research Institute are measured and evaluated.

Keywords : Real Time Visualization, PATRAS, Tensor

目 次

1. はじめに.....	1
2. PATRAS の概要.....	2
2.1 PATRAS の実時間可視化機能.....	2
2.2 PATRAS のポスト可視化機能.....	3
3. 新機能及び追加機能の概要.....	5
3.1 2次元テンソル表示対応.....	5
3.2 分離要素対応.....	5
3.3 マルチプラットフォーム対応.....	6
3.4 非構造六面体要素可視化の高速化.....	6
4. PATRAS の CHIKAKU DYNAMIC への適用.....	7
4.1 CHIKAKU SYSTEM の概要.....	7
4.2 実時間可視化機能の適用.....	8
4.3 ポスト可視化機能の適用.....	9
4.4 可視化例.....	10
5. PATRAS の可視化性能.....	12
5.1 可視化速度の設定条件.....	12
5.2 図種毎の性能.....	13
5.3 並列処理の性能.....	15
6. PATRAS の評価.....	17
6.1 分離要素の可視化.....	17
6.2 2次元テンソル量の可視化.....	18
6.3 PATRAS の機能.....	19
7. 議論.....	20
8. まとめ.....	21
謝辞.....	21
参考文献.....	22
付録.....	23
A. テンソルの計算式.....	23
A.1 テンソルの計算式.....	23
B. テンソル量表示のユーザズ・マニュアル.....	25
B.1 テンソル可視化機能の使用法.....	25
B.2 カラーバー表示機能の使用法.....	29
C. テンソル量表示のプログラマーズ・マニュアル.....	31

C.1	RVS_INIT の仕様.....	31
C.2	RVS_HEX1 の仕様.....	32
C.3	作業領域のサイズ.....	33
C.4	rvs_colorbar の仕様.....	34
D.	マルチプラットフォーム化.....	35
D.1	マルチプラットフォーム化の基本的考え方.....	35
D.2	ソースコードの修正.....	35
D.3	計算機個別の対応内容.....	38
D.4	ライブラリの作成方法.....	40

Contents

1. Preface.....	1
2. Outline of PATRAS.....	2
2.1 Real-time Visualization of PATRAS.....	2
2.2 Post Visualization of PATRAS.....	3
3. Outline of New Functions.....	5
3.1 2-dimentional Tensor Expression.....	5
3.2 Applied to Separated Elements.....	5
3.3 Applied to Multi Platform.....	6
3.4 Speedup of Visualization for Unstructured Hexahedral Elements.....	6
4. Application to PATRAS on CHIKAKU DYNAMIC.....	7
4.1 Outline of CHIKAKU SYSTEM.....	7
4.2 Applied to Real Time Visualization Function.....	8
4.3 Applied to Post Visualization Function.....	9
4.4 Sample Output of Visualization.....	10
5. Performance Evaluation of PATRAS.....	12
5.1 Condition of Visualization Speed Measurement.....	12
5.2 Performance of Various Drawing Method.....	13
5.3 Performance of Parallel Processing.....	15
6. Evaluation of PATRAS.....	17
6.1 Visualization of Separated Element.....	17
6.2 Visualization of 2-D Tensor Quantity.....	18
6.3 Function of PATRAS.....	19
7. Discussion.....	20
8. Summary.....	21
Acknowledgements.....	21
References.....	22
Appendix.....	23
A. Definition of Tensor.....	23
A.1 Definition of Tensor.....	23
B. User's Manual for Tensor Visualization.....	25
B.1 Usage of Tensor Visualization.....	25
B.2 Usage of Colorbar.....	29
C. Programmer's Manual for Tensor Visualization.....	31
C.1 Specification of RVS_INTT.....	31

C.2 Specification of RVS_HEX1.....	32
C.3 Size of Work Area.....	33
C.4 Specification of rvs_colorbar.....	34
D. Modification for Multi-platform.....	35
D.1 Basic Concept for Multi-platform.....	35
D.2 Modification of Source Code.....	35
D.3 Treatment of Each Machine.....	38
D.4 How to Make Library.....	40

1. はじめに

実時間可視化システムである PATRAS (Parallel Tracking and Steering system) は、地球シミュレータで出力された大規模な解析データを地球シミュレータによる効率的な可視化を目的として開発され、BFC (Boundary Fitted Curvilinear) 格子、非構造四面体要素および非構造六面体要素に対応している ([1])。

これまで、BFC 格子は、緊急時のおける放射能放出源推定システム ([5]) に適用されている ([3])。しかしながら、これまで、PATRAS の非構造四面体および非構造六面体要素に適用したシステムが無く、本機能に関する評価がなされていなかった。

そこで、理研が開発中の地球シミュレータ用ソフトウェアである CHIKAKU SYSTEM ([6] 等を参照のこと) の 1 機能である CHIKAKU DYNAMIC に PATRAS の非構造六面体要素の機能を適用し、その評価を行うことにした。

なお、本報告書は、PATRAS の概要、新機能の概要、CHIKAKU SYSTEM への適用、PATRAS の可視化性能および評価を記述している。あわせて、付録に新規に開発した機能についてのユーザーズ・マニュアルとプログラマーズ・マニュアル、およびマルチプラットフォーム化の対応方法を記述している。

2. PATRAS の概要

2.1 PATRAS の実時間可視化機能

日本原子力研究所（以下、原研）では、以下の要求に答えるべく、実時間可視化システムである PATRAS (PARallel TRacking and Steering system) を開発している。

- デバッグやデモ等の目的で計算結果を計算途中でモニタリングする機能 (トラッキング機能)
- 解析計算実行の途中で、計算パラメータが変更できる機能 (ステアリング機能)

PATRAS の実時間可視化システム構成は ([1]) Fig. 2.1 のとおりであり、ハードウェアとしては、計算サーバーが一 PE のベクトル計算機、共有メモリ型のベクトル計算機および分散メモリ型列計算機の 3 階層のプロットフォームを、クライアント側では、Web ブラウがインストールされていることを前提としている。さらに、データ中継の機能を果たす Web サーバーの存在も仮定しており、計算サーバーと高速なネットワークで接続されているものとしている。ソフトウェアは、計算サーバー、Web サーバおよびクライアントのハードウェア上で動作する可視化モジュールデータ中継モジュールおよび GUI モジュールから構成されている。

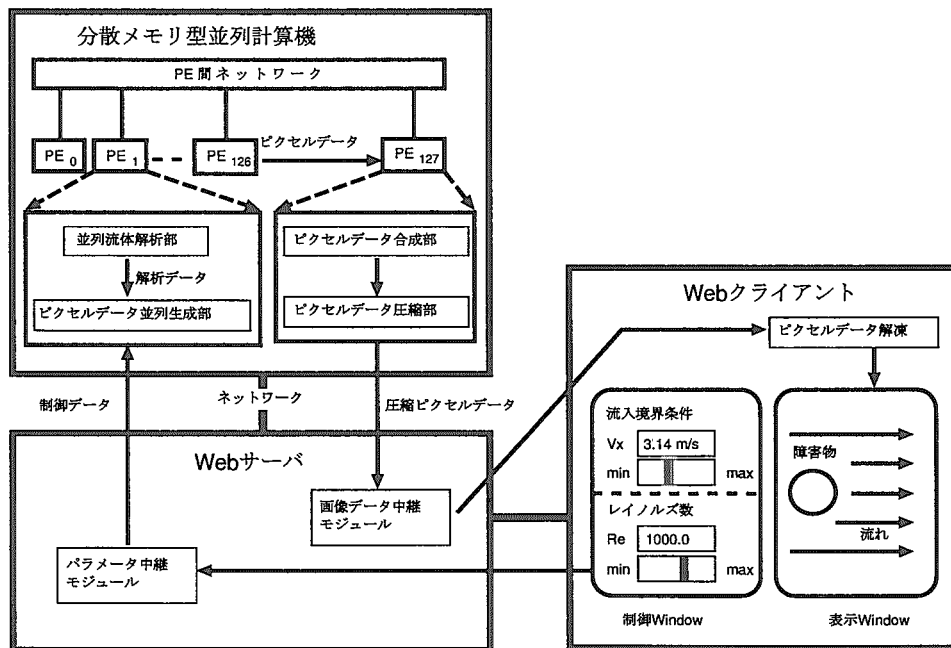


Fig. 2.1 System configuration of PATRAS

PATRAS は、Fig 2.1 にあるとおり、解析から可視化までの一連の処理（解析、マッピング処理およびレンダリング処理）を、以下の方式で行っている。

- 解析・マッピング処理・レンダリング処理の全てをサーバー側で実行し、ピクセル表示のみクライアント側で実行する。
- サーバーからクライアントに転送されるデータは、ピクセルデータのみであり、転送されるデータ量は、計算体系に依存せず一定である。

PATRAS が採用している並列化手法は、Owner Computational rule と呼ばれる手法を採用している。即ち、PATRAS での分散メモリ並列化は、解析計算の並列化を領域分割法に基づくものと想定し、各部分領域の可視化は、該当する解析計算を担当するプロセッサが行なうようにしている。この方式により、領域分割法で並列化された解析計算の可視化においては、プロセッサ間のデータ転送量を少なく抑えることができる。各部分領域の可視化画像は、最終的に 1 つのプロセッサに集めて合成し、1 枚の可視化画像を生成している。

現在、以下のような図種を可視化することが可能である。

- オブジェクト表示
- 等高線表示
- ベクトル図表示
- 等値面表示
- パーティクルトレース

また、可視化対象の格子体系は、以下の 3 種類である。

- BFC 格子
- 非構造四面体格子
- 非構造六面体格子

2.2 PATRAS のポスト可視化機能

PATRAS 自体のポストプロセッシング機能は、以下の 2 種類のファイル形式のデータを可視化の対象としている。

- NetCDF (Network Common Data Forum) 形式のファイル
NetCDF は、Unidata Program Center の Glenn Davis, Russ Rew および Steve Emmersion によって開発された、配列指向のデータアクセスのためのインターフェースおよびそのインターフェースの実装を与える C, C++ および Fortran ソフトウェアライブラリである。

- GrADS 形式のファイル

GrADS は、コロンビア大学海洋陸面大気研究所で開発されたフリーの気象データプロット用ソフトウェアである。GrADS 形式のデータには、gridded data と station data があるが、PATRAS では、gridded data のみである。

基本的なシステム構成は、実時間可視化システムと同じである。追加された機能としては、シナリオ機能がある。

ここでは、標準的な可視化ソフトウェアである AVS の UCD (Unstructured Cell Data : 非構造格子データ) 形式のファイルを対象とし、CHIKAKU DYNAMIC 専用に開発したものをポスト可視化機能と呼ぶことにする。

3. 新機能及び追加機能の概要

3.1 2次元テンソル表示対応

PATRAS で表示できる物理量は、スカラー量とベクトル量のみであり、地震の発生過程の解析などをする際に必要となる応力や歪などのテンソル量を可視化することができなかった。現状では、3次元のテンソル量の可視化方法が確立されている訳でなく、今後の研究課題と考えている。当面は、2次元のテンソル量を対象とし、応力や歪の Compression/Extension の方向を表示することとし（2次元表示の例は、[10] の図 8 等を参考のこと）、CHIKAKU DYNAMIC 等で必要となるテンソル量を2次元化して可視化する機能を新たに開発した。

3.2 分離要素対応

これまで、PATRAS は、マルチブロックに対応していなかった。

プレートを含んだ地殻および上部マントル構造を解析する場合、Fig. 3.1 のように、沈み込む海洋プレートを狭んだサンドイッチ構造をしている。

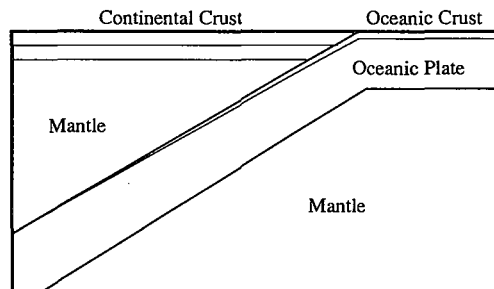


Fig. 3.1 Schematic Structure of Crust and Upper Mantle

CHIKAKU DYNAMIC では、プレートに対応するブロックとそれを狭んだ左右のブロックとは分離した構造を解析する場合、2つの物体の接触を扱うことになる。CHIKAKU DYNAMIC では、ペナルティ法を用いて解析している（詳細は、[8] 等を参照のこと）。この結果、ブロック間は摩擦力で保持されているので、複数のブロックが独立して存在することになる。そこで、PATRAS をマルチブロックに対応することにした。ただし、個々のブロックを独立に可視化できるのではなく、全ブロックを一体として、可視化するのもであり、分離要素対応と呼ぶことにする。

3.3 マルチプラットフォーム対応

これまで、PATRAS は、NEC SX-4 上で開発してきたが、地球シミュレータ向けではなく、ITBL (IT-Based Laboratory) 対応のために、原研内の各種並列計算機で実行できるようにマルチプラットフォーム化に対応した。実行可能な並列計算機は、以下のとおりである。ただし、新機能は、今のところ、NEC SX-4 以外に IBM RS/6000 SP と Hitachi SR8000 にしか対応していない。

- NEC SX-4
- IBM RS/6000 SP
- Hitachi SR8000F1
- Fujitsu VPP5000
- SGI Origin 3800
- Compaq SC/ES40

3.4 非構造六面体要素可視化の高速化

現行の PATRAS では、非構造六面体要素の格子を可視化する場合、一旦四面体格子に変換し、四面体格子を可視化する機能を使用して可視化を行なっている。このような方法で可視化を行なうと、以下の問題が生ずる。

- 大規模計算では、変換に時間がかかる。
- 分離要素に対応できない。

そこで、非構造六面体要素を直接可視化する方式に変更し、上記の問題点を改善することにした。ただし、対象とする図種は、ベクトル図および等高線図である。

4. PATRAS の CHIKAKU DYNAMIC への適用

4.1 CHIKAKU SYSTEM の概要

CHIKAKU SYSTEM は、以下の 6 種類のサブシステムから構成されている ([6] や [7]) .

- CHIKAKU DB (地殻データ)
本システムは、日本列島付近の地殻データの入力、編集および表示の操作を行なうための機能である。このデータの中からシミュレーションを行なう際に必要な領域のデータだけを切り取り、それを地層境界標高グリッドデータ、地層境界面ポイントデータ等、地殻 CAD システムに必要なデータに変換する。
- CHIKAKU CAD (ソリッドモデルの作成)
本システムは、変換されたデータを入力とし、地殻構造の 3 次元ソリッドモデルデータを作成する機能である。この 3 次元ソリッドモデルは IGES と呼ばれる CAD/CAM では一般的な共通フォーマットで出力され、メッシュジェネレータに渡される。
- CHIKAKU MESH (メッシュデータの作成)
本システムは、IGES データを基にメッシュデータを作成し、静的解析、動的解析を行なうプログラムにユニバーサルファイルの形式で引き渡す機能である。
- CHIKAKU STATIC (静的解析)
本システムは、自重解析やプレート境界での応力蓄積過程を計算し、得られた変位、応力、歪などの値を動的解析に渡す機能である。
- CHIKAKU DYNAMIC (動的解析)
本システムは、地震発生過程や地震波伝播の解析を行なう機能である。
- PATRAS (可視化処理)
本システムは、解析結果を実時間で可視化したり、計算終了後にポスト処理で可視化を行なう機能である。

なお、CHIKAKU MESH は、原研で開発しているメッシュ・ジェネレータである JAERI MESH のうち非構造格子生成システムの機能 (六面体格子) を使用している。非構造格子生成システムについては、[3] を参照のこと。

4.2 実時間可視化機能の適用

オリジナル版 CHIKAKU DYNAMIC の可視化に関連する部分のツリー構造は, Fig. 4.1 のとおりである.

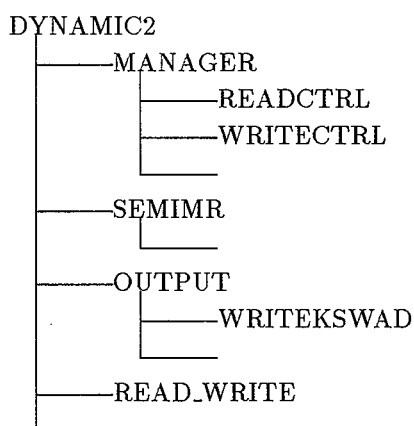


Fig. 4.1 Tree structure of Original CHIKAKU DYNAMIC

PATRAS の実時間可視化機能を組み込んだ CHIKAKU DYNAMIC の主要部分のツリー構造は, Fig. 4.2 である. 四角で囲んだ PATRAS, WRITEINP, NODESUM および NODEAVE ルーチンを追加し, 下線を付けたルーチンを変更した.

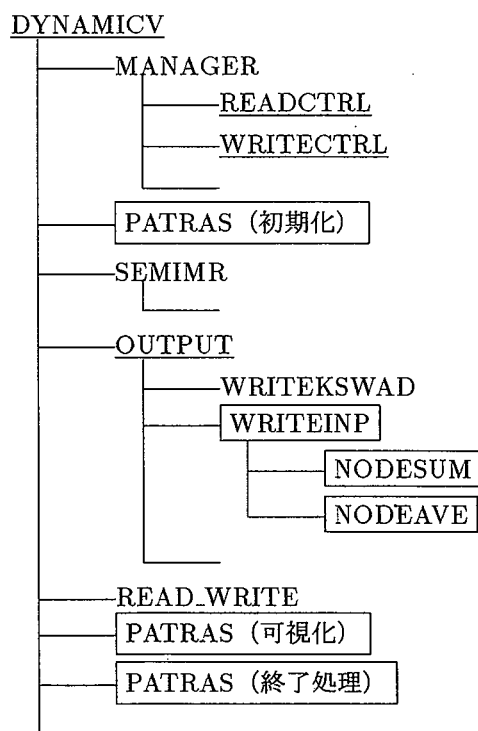


Fig. 4.2 Tree structure of PATRAS version CHIKAKU DYNAMIC

新規および修正したルーチンの内容は、以下のとおりである。

- DYNAMIC2(DYNAMICV)
解析するループの中に、PATRAS の制御（初期化処理、可視化処理および終了処理）を追加した。あわせて、必要な変数や配列の設定等をおこなっている。
- PATRAS
本ルーチンは、PATRAS の 3 種類の処理（初期化処理、可視化処理および終了処理）を制御し、個々の処理に必要な設定処理を実行するように新規に追加したものである。
- WRITEINP
本ルーチンは、ポスト可視化処理で使用する可視化データを作成するために、KSWAD 用の WRITEKSWAD を基に AVS の INP 形式で出力できるように改造したものである。
- NODESUM
NODESUM は、要素で定義された値を節点で定義された値に変換するために、節点上での合計値を計算するルーチンである。
- NODEAVE
NODEAVE は、要素で定義された値を節点で定義された値に変換するために、節点上での合計値から平均値を計算するルーチンである。
- READCTRL
既存の制御用のデータを入力するルーチンに、PATRAS で使用する制御用データを追加したものである。追加したデータは、以下のとおりである。
 - PATRAS による可視化の適否、PATRAS のモード (BATCH か INTERACTIVE) を指定するフラグ
 - 可視化あるいは可視化用データを出力するサイクル数
 - 可視化用データの出力の有無、出力フォーマット (AVS5 か AVS/Express) を指定するフラグ。
- WRITECTRL
既存の制御用のデータを出力するルーチンに、PATRAS で使用する制御用データを出力するように修正した。
- OUTPUT
既存の出力ルーチンに可視化用データを出力する WRITEINP ルーチンを追加した。

4.3 ポスト可視化機能の適用

PATRAS のポスト可視化機能用のプログラムのツリー構造は、Fig. 4.3 のとおりである。なお、下線を付けたルーチンは、実時間可視化と同じ入力データが使用できるようにオリジナル版 CHIKAKU DYNAMIC から必要な箇所を抜き出している。

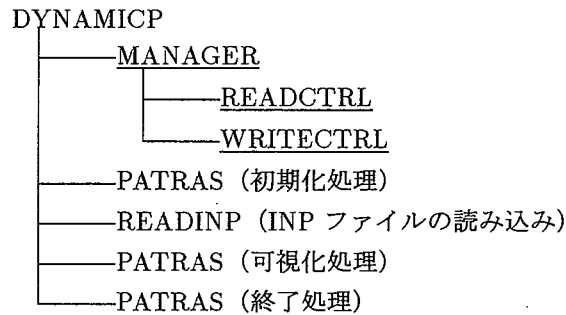


Fig. 4.3 Tree structure of Post Visualization System

新規および修正したルーチンの内容は、以下のとおりである。

- DYNAMICIP
PATRAS の制御（初期化处理，可視化处理および終了処理）および可視化用データを読み込むルーチン呼び出すような main プログラムである。
- READINP
AVS の inp フォーマットで出力された可視化用データを読み込むルーチンである。使用するプロセッサ台数の応じて、逐次用のデータと並列用のデータを選択して、データを読み込むようにしている。
- READCTRL
実時間可視化用に修正した READCTRL ルーチンを可視化に関連した部分のみに修正した。ただし、既存の入力データを変更しないで利用できるように配慮している。
- WRITECTRL
実時間可視化用に修正した WRITECTRL ルーチンを可視化に関連した部分のみに修正した。

4.4 可視化例

断層運動を模擬する岩石すべり破壊実験 [9] をシミュレーションした結果について可視化した例を Fig. 4.4 に示す。これは、ポスト可視化を INTERACTIVE モードで実行し、WEB 上で可視化結果を表示した例である。可視化の内容は、解析対象である岩石の下半分を構成する 5 面について全変位の等高線と変位の 3 成分のベクトルの矢印を用いて、可視化した。手前の斜面は断層を模擬したすべり面に相当し、等高線の赤い部分は変位が大きく、青い部分は変位が小さいことを示している。

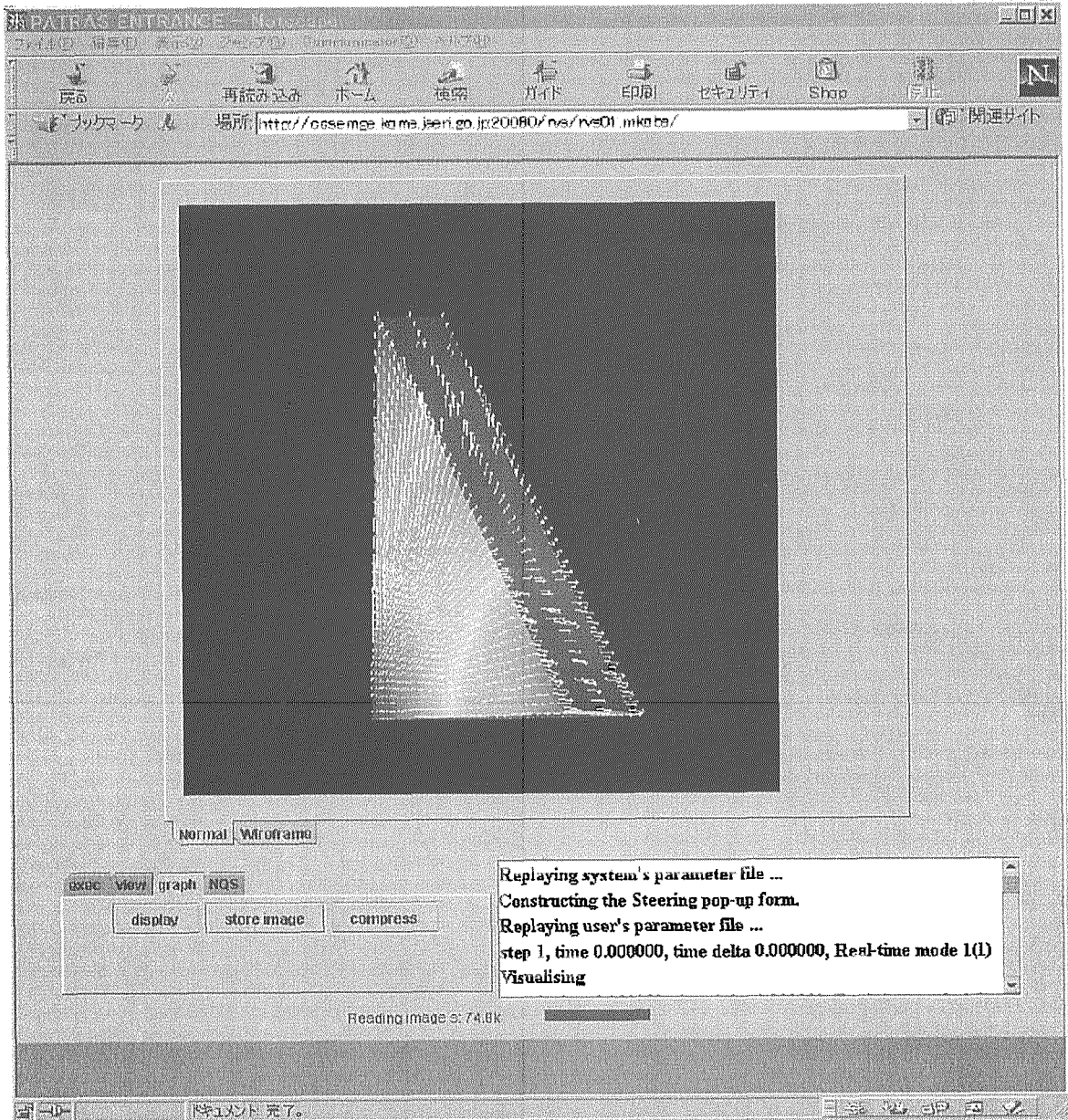


Fig. 4.4 Sample Visualization for simulation of stick-slip shear failure of Rock masses (Vector and Contour)

5. PATRAS の可視化性能

5.1 可視化速度の設定条件

断層運動を模擬する岩石すべり破壊実験のシミュレーション結果を以下の条件の基に可視化し、各種可視化速度を測定した。

- 測定対象の並列計算機
 - 原研内にある同程度の演算性能を持つ以下の並列計算機を測定対象とした。
 - NEC SX-4 (2.0 GFLOPS/CPU)
 - IBM RS/6000 SP (1.5 GFLOPS/CPU)
 - Hitachi SR8000/F1 (1.5 GFLOPS/CPU)
- 可視化の性能測定に採用した解析例
 - 岩石すべり破壊実験のシミュレーション (2 ブロック構造)
 - 2 ブロックのテストデータ
 - 3 ブロック版東海モデルによるシミュレーション
- 測定対象の図種
 - ベクトル図
 - 等高線図
 - 等値面図 (flat shading と smooth shading)
- 測定条件
 - rvs_hex1 の測定時間には可視化画像のファイル出力時間も含まれている。ファイル出力時間を除くために、Store_image オプションを false とした。
 - 可視化画面のサイズ : 512x512

使用したモデルの体系等は、Table 5.1 のとおりである。なお、岩石すべり破壊実験の欄中の括弧の中の数値は、1つのブロックのみの数値である（等値面図は分離要素に対応していないため、1ブロックで測定している）。

Table 5.1 Calculation size of various models

モデル	計算体系	節点数	要素数
岩石すべり破壊実験	280x280x50	6336 (3168)	4000 (2000)
2 ブロックのテストデータ	100x100x200	262701	250000
3 ブロック版東海モデル	95x675x90	89187	80380

5.2 図種毎の性能

5.2.1 ベクトル図の可視化速度

ベクトル図を各種計算機で可視化時間を測定した結果が、Table 5.2 である。表中の `rvs_hex1` と `rvs_main` は、測定したルーチン名で、それぞれ、非構造六面体格子可視化ルーチンおよび可視化・表示用制御ルーチンである。

Table 5.2 Performace comparison of vector mapping on various computers

使用計算機	岩石すべり破壊実験		2ブロックのテストデータ	
	<code>rvs_hex1</code>	<code>rvs_main</code>	<code>rvs_hex1</code>	<code>rvs_main</code>
NEC SX-4	8.295D-05 sec	5.184D-01 sec	8.325D-05 sec	1.144D+00 sec
IBM RS/6000 SP	3.564D-05 sec	8.798D-02 sec	6.456D-05 sec	1.537D-01 sec
Hitachi SR8000	2.110D-05 sec	1.360D-01 sec	2.654D-05 sec	4.604D-01 sec

上記の表は、`rvs_main` ルーチンの性能が RS/6000 SP, SR8000, SX-4 の順に遅くなっていることを示している。SX-4 はベクトル化があまり効いていないためか、他の計算機と比較して倍以上時間がかかっている。また、計算体系が大きくなっても、計算機間の可視化性能は、ほとんど同じである。これに対して、`rvs_hex1` ルーチンでは、SR8000 と RS/6000 SP との性能が逆転している。RS/6000 SP を除くと計算体系に依存せずに、ほぼ同じ性能となっている。

5.2.2 等高線図の可視化速度

等高線図を各種計算機で可視化時間を測定した結果が、Table 5.3 である。

Table 5.3 Performace comparison of contour mapping on various computers

使用計算機	岩石すべり破壊実験		2ブロックのテストデータ	
	<code>rvs_hex1</code>	<code>rvs_main</code>	<code>rvs_hex1</code>	<code>rvs_main</code>
NEC SX-4	9.821D-05 sec	2.286D-01 sec	8.325D-05 sec	2.142D-01 sec
IBM RS/6000 SP	3.469D-05 sec	5.368D-01 sec	6.163D-05 sec	1.774D-01 sec
Hitachi SR8000	2.511D-05 sec	4.794D-01 sec	2.474D-05 sec	1.341D-01 sec

等高線図では、ベクトル図と異なり、`rvs_main` ルーチンでは、SX-4 と他の計算機との差が小さく、岩石すべり破壊実験では、一番速くなっている。`rvs_hex1` ルーチンは同じ計算体系を使用しているため、ベクトル図と同等の性能となっている。

5.2.3 等値面図の可視化速度

等値面図を各種計算機で可視化時間を測定した結果が、Table 5.4 である。なお、表中には、2種類の shading 手法 (flat と smooth) で測定した結果を示している。

Table 5.4 Performace comparison of iso-surface mapping on various computers

使用計算機	岩石すべり破壊実験 (flat)		岩石すべり破壊実験 (smooth)	
	rvs_hex1	rvs_main	rvs_hex1	rvs_main
NEC SX-4	7.830D-05 sec	4.878D-01 sec	7.827D-05 sec	2.158D+00 sec
IBM RS/6000 SP	3.140D-05 sec	2.257D-01 sec	2.781D-05 sec	1.349D+00 sec
Hitachi SR8000	2.331D-05 sec	2.264D-01 sec	2.383D-05 sec	1.089D+00 sec

等値面図では、SR8000 と RS/6000 SP と同等か同等の性能であり、SX-4 は倍ぐらい遅くなっている。shading の手法による可視化性能の比較では、smooth shading は、flat shading の 5 倍ぐらい時間がかかっているため、アルゴリズムの見直しを実施することにより、性能改善を図る必要がある。

5.2.4 可視化方法別の性能比較

今回、分離要素への対応として、六面体格子から四面体格子への変換した後に可視化する方法から、六面体格子を直接可視化する方法に変更した。

NEC SX-4 上で、ベクトル図と等高線図の可視化速度を比較した。岩石すべり破壊実験で下半分のブロックのみ可視化することで、新旧の手法における可視化速度を比較した。結果は、Table 5.5 と 5.6 のとおりである。

Table 5.5 Performace comparison of vector mapping on various method

可視化手法	岩石すべり破壊実験		2 ブロックモデル	
	rvs_hex1	rvs_main	rvs_hex1	rvs_main
四面体に変換した可視化	7.989D-05 sec	1.866D-01 sec	7.745D-05 sec	2.502 sec
六面体を直接可視化	8.138D-05 sec	2.109D-01 sec	8.079D-05 sec	1.204 sec

Table 5.6 Performace comparison of contour mapping on various method

可視化手法	岩石すべり破壊実験		2 ブロックモデル	
	rvs_hex1	rvs_main	rvs_hex1	rvs_main
四面体に変換した可視化	7.857D-05 sec	1.765D-01 sec	7.801D-05 sec	1.840D-01 sec
六面体を直接可視化	8.193D-05 sec	2.284D-01 sec	8.133D-05 sec	2.182D-01 sec

上記の表から以下のことが明らかとなった。

- モデルの体系が小さい場合は、一旦四面体格子に変換した方が、六面体格子を直接可視化するより速い。
- モデルの体系が大きくなると、六面体格子を直接可視化した方が速くなる傾向がある。

5.3 並列処理の性能

並列版 PATRAS のテストをするために、以下の手順で 2 プロセッサ、4 プロセッサおよび 8 プロセッサ用のデータを作成した。なお、使用したデータは、岩石すべり破壊実験シミュレーションと 3 ブロック版東海モデルの 2 種類である。

- (1) CHIKAKU DYNAMIC のソリッドデータと境界条件を基に、変換ツール geo2chikaku を使用して GeoFEM のパーティショナーに渡すデータを作成する。
- (2) GeoFEM のパーティショナー (partGeoFEM) を使用して、METIS 入力情報を作成する。
- (3) METIS (kmetis) を使用して、必要なプロセッサ台数に分割する領域分割情報を作成する。
- (4) METIS (kmetis) で作成した領域分割情報のデータと逐次版で作成した AVS の inp ファイルを基に、並列版用に AVS の inp ファイルを作成する。

NEC SX-4 上で 3 ブロック版東海モデルの YZ 平面の変位をベクトル図で可視化させた場合の並列性能は、Table 5.7 のとおりである。

Table 5.7 Parallel efficiency of vector mapping for Tokai model on SX-4

	NEC SX-4	
	rvs_hex1	rvs_main
1 CPU	8.311D-05 sec	5.871D-01 sec
2 CPU	1.427D-04 sec	8.005D-01 sec
4 CPU	2.164D-04 sec	4.726D-01 sec

3 ブロック版東海モデルと同様に、NEC SX-4 上で、岩石すべり破壊実験シミュレーションにおける ZX 平面の変位をベクトル図で可視化させた場合の並列性能は、Table 5.8 のとおりである。

Table 5.8 Parallel efficiency of vector mapping for stick-slip simulation on SX-4

	NEC SX-4	
	rvs_hex1	rvs_main
1 CPU	8.295D-05 sec	5.184D-01 sec
2 CPU	1.445D-04 sec	4.632D-01 sec
4 CPU	2.166D-04 sec	4.333D-01 sec

SX-4 における両者の測定結果とも、並列した効果が全然発揮されていない。rvs_hex1 ルーチンの性能は遅くなっている。今後、問題点を明確にして、改善を図っていく必要がある。

NEC SX-4 と同様に IBM RS/6000 SP と Hitachi SR8000 上で、岩石すべり破壊実験シミュレーションにおける XZ 平面の変位をベクトル図で可視化させた場合の並列性能は、Table 5.9 のとおりである。

Table 5.9 Parallel efficiency of vector mapping for stick-slip simulation

	IBM RS/6000 SP		Hitachi SR8000	
	rvs_hex1	rvs_main	rvs_hex1	rvs_main
1 CPU	3.585D-05 sec	8.801D-02 sec	2.110D-05 sec	1.360D-01 sec
2 CPU	6.370D-05 sec	1.062D-01 sec	6.677D-05 sec	1.137D-01 sec
4 CPU	1.038D-04 sec	1.525D-01 sec	8.331D-05 sec	1.149D-01 sec
8 CPU	3.932D-02 sec	6.459D-01 sec	1.170D-04 sec	1.546D-01 sec

SX-4 と同様に、RS/6000 SP と SR8000 についても両者の測定結果とも、並列した効果がほとんど無い。rvs_main ルーチンでは多少速くなっているが、rvs_hex1 ルーチンの性能は逆に遅くなっている。今後、問題点を明確にして、改善を図っていく必要がある。

6. PATRAS の評価

6.1 分離要素の可視化

これまで、PATRAS が分離要素（マルチブロック）に対応していなかったために、Fig. 4.4 のように一つのブロック（図では、下半分ブロックのみ）しか可視化することができなかった。

これに対して、今回分離要素に対応したことにより、複数のブロックを一括して可視化することが可能となった。この結果、岩石すべり破壊実験のシミュレーションの可視化結果（Fig. 6.1）で、すべり面（図中の左上から右下への対角線）を狭んで、垂直方向の変位量の違いが比較できるようになった。

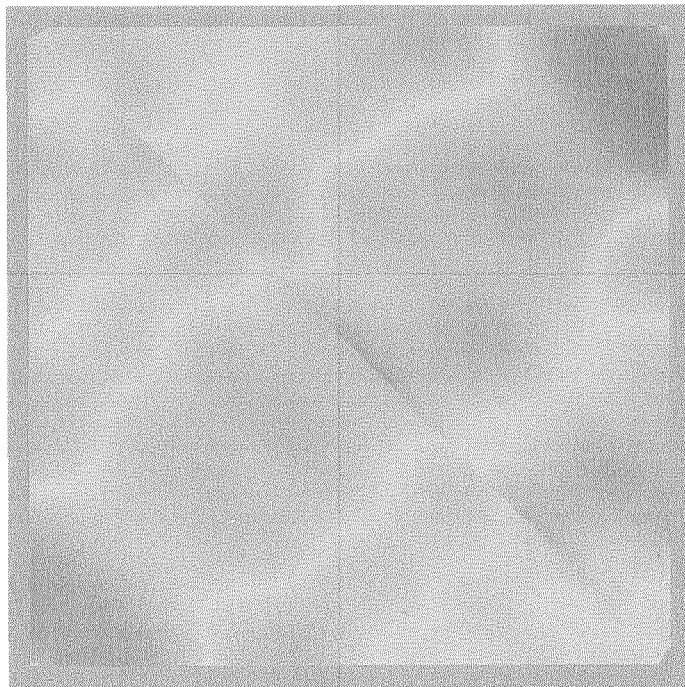


Fig. 6.1 Sample Visualization for simulation of stick-slip shear failure of Rock masses using Separated Element Version

さらに、プレートを狭めた3つのブロックからなる東海モデルに分離要素に対応したPATRASを適用することで、本機能が有効であることを確認した。

6.2 2次元テンソル量の可視化

2次元テンソル量の可視化について、以下に述べる。2次元テンソル量を可視化した岩石すべり破壊実験のシミュレーションの可視化結果を Fig. 6.2 に示す。この図は、応力を ZX 平面 ($Y = 275 \text{ mm}$) を上側から表示したものである。図の左端付近に断層を模擬したスリップ面 (図では線) があり、スリップ面付近で、応力の方向が変化している様子が可視化できている。

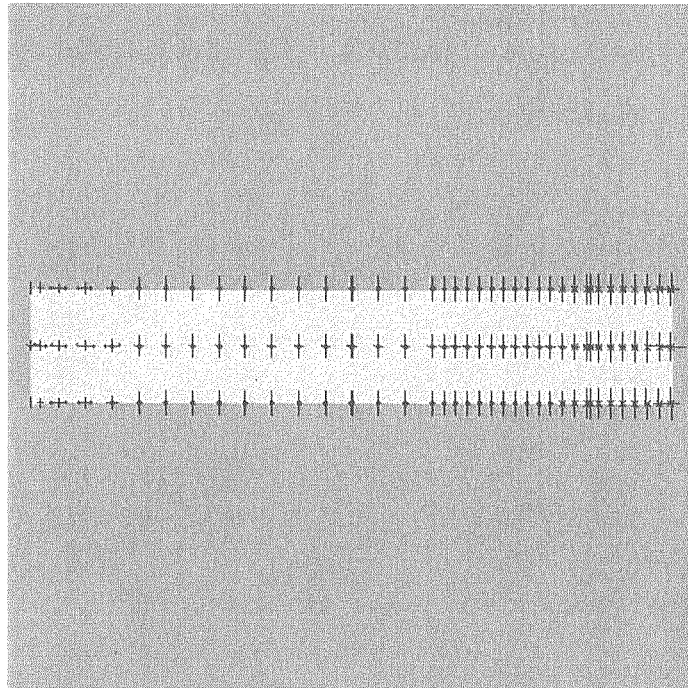


Fig. 6.2 2-D Tensor Visualization for simulation of stic-slip shear failure of Rock masses using Separated Element Version

各種並列計算機上でのテンソル図の可視化速度は、Table 6.1 のとおりである。

Table 6.1 Visualization speed of tensor mapping on various computers

使用計算機	岩石すべり破壊実験 (XY 面)		岩石すべり破壊実験 (ZX 面)	
	rvs_hex1	rvs_main	rvs_hex1	rvs_main
NEC SX-4	8.352D-05 sec	1.439D+00 sec	9.770D-05 sec	2.098D-01 sec
IBM RS/6000 SP	3.335D-05 sec	1.898D-01 sec	3.858D-05 sec	4.599D-02 sec
Hitachi SR8000	2.254D-05 sec	3.693D-01 sec	2.836D-05 sec	4.968D-02 sec

なお、基本的な計算機間の性能比は、他のベクトル図や等高線図と同等である。

6.3 PATRAS の機能

当初, PATRAS は流体解析の結果を可視化する目的で開発し [1], 構造格子の可視化が中心であった. このため, 非構造格子を扱おう構造解析の結果を可視化する上で不足している機能がある. 今回, 2次元テンソル量を表示する機能を追加したが, 以下のような機能が不足しており, 今後使用頻度に応じて, 機能追加する必要がある.

- 面コンター図

現在, 任意断面 (非構造格子) か格子面 (構造格子) を可視化する機能しかない. 構造解析では物体表面の物理量 (応力やひずみ等) を可視化することがあり, 面コンター図が必要となる.

- 変形図

構造解析では物体が変形する様子を可視化することがあり, 変形図が必要となる.

実時間可視化とポスト可視化の役割分担の観点から以下のことがある.

シナリオ機能以外は, PATRAS の実時間可視化とポスト可視化機能は基本的に同じである. 本来, ポスト可視化機能には, 実時間可視化で可視化速度を重視するために対応できなかった機能を補間する役割があると考えている. 例えば, ボリュームレンダリングのような時間がかかる処理を解析と同時に実行するのではなく, 可視化用データを基にポスト処理にて対応したり [4], データ階層化機能を用いてデータ転送量を削減するようにしていくことが考えられる. また, 業界標準の可視化ソフトウェアである AVS だけでなく, 地球シミュレータ上で提供されている気象や固体地球の分野で広く使用されている可視化用ソフトウェアである GrADS や Vis5D との棲み分けをはかっていく必要がある.

7. 議論

今回、各種計算機で可視化速度の性能を測定した。しかしながら、以下のことが原因で、十分な測定が実施できなかった。

- 実際に使用している計算体系を採用しているため、任意に体系を変更できなかった。この結果、サイズ効果の調査が不十分である。
- 実際の可視化で使用している格子点数を把握できていないので、格子 1 点あたりの可視化速度が個々の図種毎に比較できていない。
- (最初だけしか解析していないために) 可視化部分が一様でなく、計算体系を大きくしても、サイズの効果がほとんど確認できなかった (特に等高線図の場合)。

また、可視化の性能は、可視化画像を出力するフォーマット (JPEG, RMV 等) に依存するので、(実時間) 可視化の性能を向上させるためには、時間がかかっているフォーマットを、チューニングして高速化を図る必要がある。

並列性能の測定に関しては、可視化した断面と領域分割した領域との関係が最適でないために、負荷に不均衡が発生している可能性が大きい。この結果、並列化効率が悪くなっていることが考えられる。可視化する断面と領域分割とを最適にする工夫をすることで負荷分散を図り、並列性能を向上させる必要がある。

今回は、3 次元テンソル量を 2 次元にマッピングすることで可視化した。しかしながら、3 次元のテンソル量をそのまま可視化することが望ましい。

現状の 3 次元テンソル量の可視化法はあまり存在しないが、Scheuerman 等 [11] によると、以下のような手法が考えられている。

- Hedgehog (tripod) 法
- Hyperstreamline 法
- Hypersurface 法

また、医療分野ではテンソル量の可視化の研究が行われているので ([12])、今後ともテンソル量の可視化は重要であると考えている。

8. まとめ

日本原子力研究所計算科学技術推進センターが開発してきた PATRAS の非構造六面体格子の機能を理化学研究所が開発中の地球シミュレータ向けソフトウェアである CHIKAKU SYSTEM に適用した。PATRAS を CHIKAKU DYNAMIC に適用するにあたり、分離要素に対応するとともに、あらたに 2 次元テンソル量表示機能とカラーバー表示機能を追加した。新機能に対応した PATRAS を CHIKAKU DYNAMIC に組み込んだ実時間可視化版を開発した。実時間可視化版には、AVS の UCD 形式のデータを出力する機能も開発した。この可視化用データを基に、ポスト可視化用のコードもあわせて開発した。開発した実時間版およびポスト可視化版コードを使用し、分離要素対応と 2 次元テンソル量表示が CHIKAKU DYNAMIC の解析結果を可視化する上で有効であることろあわせて、原研内の 3 種類の並列計算機で可視化速度を測定し、性能を評価した。

謝辞

日本原子力研究所計算科学推進センターの平山 G L および村松課長代理には本開発にあたり、多大のご支援を賜りました。ここに感謝の意を表します。また、PATRAS を CHIKAKU SYSTEM に組み込むに際して、CHIKAKU SYSTEM を開発している理化学研究所の皆様にも多大な協力を得ましたことに感謝いたします。本研究は、理化学研究所との共同研究の成果であり、PATRAS は、日本電気株式会社殿との共同研究の成果です。

参考文献

- [1] 村松 ほか: 並列計算機上での流体解析のための実時間可視化システムの開発, *JAERI-Data/Code*, 98-014, 1998.
- [2] 村松 ほか: 並列計算機上での非構造格子生成システムの開発, 計算工学講演会論文集, *Vol.5*, pp.353-356, 1998.
- [3] K. Muramatsu et al: Parallel Real-time Visualization System for Large-scale Simulation - Application to WSPEEDI - (Abstract), *The Fourth International Conference on Supercomputing in Nuclear Applications*, pp.61, 2001.
- [4] 大谷孝之, 村松一弘: 実時間可視化システムのためのボリュームレンダリング・モジュールの開発, *JAERI-Data/Code*, 200-021, 2000.
- [5] 北端秀行, 茅野政道: 緊急時における放射能放出源推定システムの開発, *JAERI-Data/Code*, 99-012, 1999.
- [6] 本間高弘 ほか: 地震発生及び地震波伝播予測のための有限要素法並列ソフトウェアシステムの開発 (要旨), 日本地震学会 2000 年秋季大会, 2000
- [7] 理研シンポジウム: 地球シミュレータ用応用ソフトウェア CHIKAKU システムの開発ー地震発生過程のシミュレーションに向けてー, 2002 年 3 月
- [8] H. L. Xing and A. Makinouchi: ITAS/Contact:For Nonlinear Friction Contact Between Elasto-Plastic Body (Abstract), *Proceedings of the Conference on Computational Engineering and Science*, Vol. 6, pp.185-188, 2001
- [9] M. Ohnaka and Y. Kuwahara: Characteristic features of local breakdown near a crack-tip in the transition zone from nucleation to unstable rupture during stick-slip shear failure, *Tectonophysics*, 175, pp.197-220, 1990
- [10] 河内崇, 宮下芳: フィリピン海プレート北端部の沈み込み・衝突の 3 次元 FEM モデル, *地震*, 50, pp.229-240, 1997
- [11] G. Scheuermann, J. Frey, H. Hagen, B. Hamann, B. Jeremic and K. Joy: Visualization of Seismic Soils Structure Interaction Simulations, In *Proceedings of the Visualization*, Vol. 6, pp.185-188, 2001
- [12] G. Kindlmann and D. Weinstein: Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields, In *IEEE Visualization 99 Proceedings*, 1999

付録 A テンソルの計算式

A.1 テンソルの計算式

ここでは、3次元のテンソル量から2次元のテンソル量に射影する方法を述べる。なお、以下の式の導入方法は、理研殿から提供された資料から抜粋したものである。

ソルバから出力される全体座標系の基底ベクトルを、

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

とする。一方、切断面に沿った座標系を可視化座標系と呼び、その基底ベクトルを $\widetilde{\mathbf{e}}_i$ ($i = 1, 2, 3$) とし、 $\widetilde{\mathbf{e}}_3$ は、切断面の法線ベクトルに一致させる。

このような設定の基で、応力テンソルは、次のように定義することができる。

$$S_{ij} \mathbf{e}_i \otimes \mathbf{e}_j \equiv \widetilde{S}_{ij} \widetilde{\mathbf{e}}_i \otimes \widetilde{\mathbf{e}}_j \equiv S$$

ふたつの座標系に関する応力の座標変換は、次式のようになる。

$$\widetilde{S}_{lk} = \sum_{i=1}^3 \sum_{j=1}^3 S_{ij} (\mathbf{e}_j \cdot \widetilde{\mathbf{e}}_k) (\mathbf{e}_i \cdot \widetilde{\mathbf{e}}_l)$$

なお、基底ベクトルの内積は、全体座標系で計算する。

可視化座標系における応力テンソルの成分をマトリックス形式で表示すると以下のようになる（これは、対象マトリックスとなる）。

$$\begin{pmatrix} \widetilde{S}_{11} & \widetilde{S}_{12} & \widetilde{S}_{13} \\ \widetilde{S}_{21} & \widetilde{S}_{22} & \widetilde{S}_{23} \\ \widetilde{S}_{31} & \widetilde{S}_{32} & \widetilde{S}_{33} \end{pmatrix}$$

この中から法線方向の成分を取り除き、二次元化する。ただし、取り除いた5成分の中での独立なものは3成分である。

$$\begin{pmatrix} \widetilde{S}_{11} & \widetilde{S}_{12} \\ \widetilde{S}_{21} & \widetilde{S}_{22} \\ \widetilde{S}_{31} & \widetilde{S}_{32} \end{pmatrix} (\widetilde{S}_{12} = \widetilde{S}_{21})$$

上記から二次元の主応力を求めることになる。

$$\begin{aligned} \tan 2\theta &= -\frac{2\widetilde{S}_{12}}{\widetilde{S}_{11} - \widetilde{S}_{22}} \\ S_I &= \frac{1}{2}(\widetilde{S}_{11} + \widetilde{S}_{22}) + \sqrt{\frac{1}{4}(\widetilde{S}_{11} - \widetilde{S}_{22})^2 + \widetilde{S}_{12}^2} \\ S_{II} &= \frac{1}{2}(\widetilde{S}_{11} + \widetilde{S}_{22}) - \sqrt{\frac{1}{4}(\widetilde{S}_{11} - \widetilde{S}_{22})^2 + \widetilde{S}_{12}^2} \end{aligned}$$

求めた主応力を基に Fig. A.1 のような矢印を描くことで、2次元テンソルを表現する。

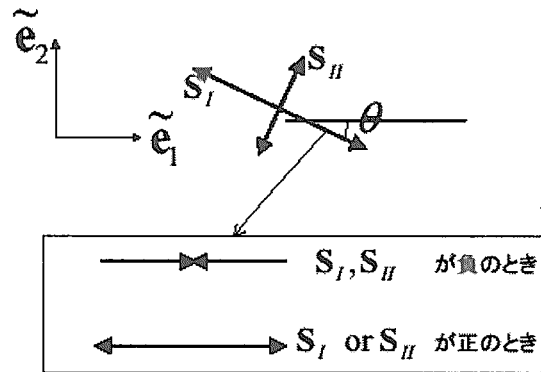


Fig. A.1 Definition of Tensor expression

付録B テンソル量表示のユーザズ・マニュアル

B.1 テンソル可視化機能の使用方法

これまでのスカラ量，ベクトル量の可視化に加え，テンソル量の可視化機能を追加した．テンソル可視化はひとつの図種として扱っている．可視化においては，テンソルから主応力を計算し，主応力が正の場合は発生点から外側に向かって矢印を描き，主応力が負の場合は内側に向かって矢印を描く仕様となっている．

B.1.1 ライブラリ呼び出し方法

既存のライブラリ呼び出しに，テンソル量の引数を追加し，スカラ量，ベクトル量と同じく，テンソル量の数，物理量ラベル，およびテンソル量の格納された配列を引数として渡す．呼び出し例は，以下のとおりである．

```
call rvs_init( ng, ns, nv, nt, label, imode, ierr )
call rvs_hex1( ig, mnod, melm, nnod, nelm, xinit, mcon,
1             ns, scal, nv, vect, nt, tens, ierr )
```

nt テンソル量の数
tens テンソル量の格納された配列

テンソル量は 6 次元で指定する．要素の並びは，テンソル量を以下のような 9 次元で表現したとき、

ライブラリには 1 データあたり S11, S22, S33, S12, S23, S13 の順で渡す．

B.1.2 パラメータ記述方法

パラメータ設定ファイル (user.para 等) の記述例を以下に示す．コンタ，ベクトル等の図種と同じく，セットアップと各面のパラメータを指定する．

【記述例】

```
Tensor_setup true stress -0.26 0.07 1 1 256 'no value' 'no value'
Tensor 1 255 128 0 250.0 1 ' ' true simple_color Arbitrary_plane \
IJ 1.0 0.0 0.0 1.0 0.0 0.0 50.0
```

パラメータフォーマットを以下に示す．

【記述例】

```
Tensor_setup true stress -0.26 0.07 1 1 256 'no value' 'no value'
```

【フォーマット】

【記述例】

型	内容	上の記述例での値
文字	表示のオン・オフ {true false}	true
文字	テンソル量ラベル	stress
実数	マッピング範囲 (min)	-0.26
実数	マッピング範囲 (max)	0.07
整数	カラーテーブル番号	1
整数	カラーテーブルインデックス (min)	1
整数	カラーテーブルインデックス (max)	256
	シミュレーション値	'no value' 'no value'

```
Tensor 1 255 128 0 250.0 1 ' ' true simple_color Arbitrary_plane \
IJ 1.0 0.0 0.0 1.0 0.0 0.0 50.0
```

【フォーマット】

型	内容	上の記述例での値
整数	面番号 1 - 8	1
整数	テンソルの色 RGB	255 128 0
実数	スケール	250.0
整数	間引き	1
update		
文字	ON/OFF {true false}	true
文字	表示タイプ {simple_color Use_color_mapping}	simple_color
文字	断面の種類 {Coordinate_plane Arbitrary_plane}	Arbitrary_plane
文字	格子断面の方向 {IJ JK KI}	IJ
実数	格子断面の位置	1.0
実数	任意断面の法線ベクトル	0.0 0.0 1.0
実数	任意断面上の点	0.0 0.0 50.0

【制限事項】

法線ベクトルはゼロベクトルでないこと。

B.1.3 GUI 操作方法

クライアント・サーバモードでの基本的な操作方は、既存のクライアントに準じる。ただし、テンソル表示に対応していない旧バージョンのクライアントとは互換がないので注意されたい。

接続が確立したら、

- (1) graph タブをクリックし、
- (2) display ボタンを押す。
- (3) 表示された図種メニューの Tensor タブをクリックすることにより、以下のようなテンソル表示パラメタ画面が表示される。

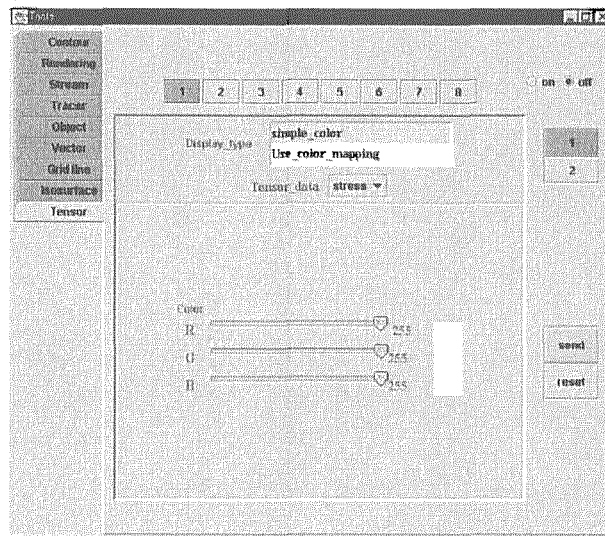


Fig. B.1 Screenshot for Tensor Mapping Parameter

単色表示 (simple_color) の場合、テンソル図の色の RGB 値を指定する。また、カラーマッピング表示の場合は、display_type のリストボックスを Use_color_mapping に指定すれば以下のような画面になる。マッピングはテンソルから計算した 2 つの主応力の大きさを元に行われる。

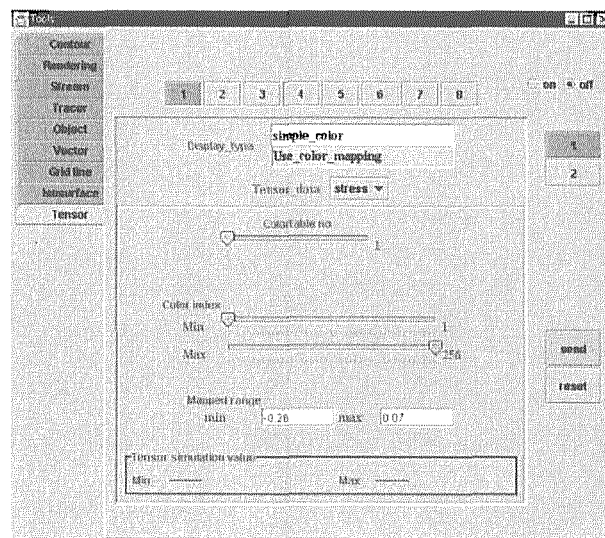


Fig. B.2 Screenshot for Tensor Color Mapping

画面右上のページボタンの「2」をクリックすると、以下のような画面に切り替わる。BFC 格子ソルバの場合はこの画面で、格子断面を指定する。

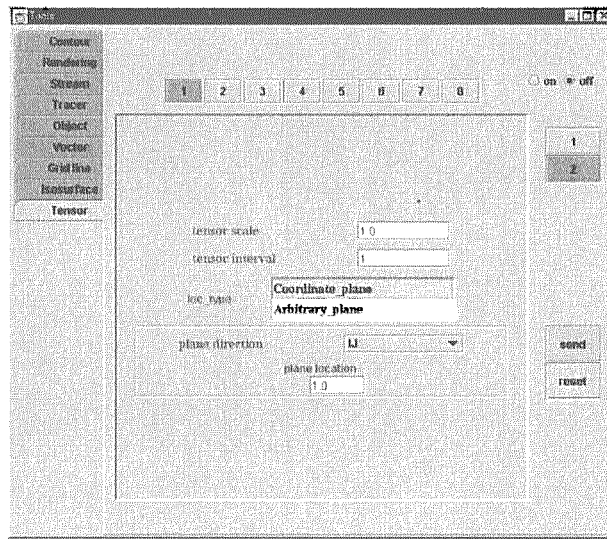


Fig. B.3 Screenshot for Tensor Mapping Parameter of BFC Lattice

また、任意断面の場合は loc_type の Arbitrary_plane をクリックすることにより、以下のような画面が表示されるので、断面の法線ベクトルと任意の 1 点を指定する。

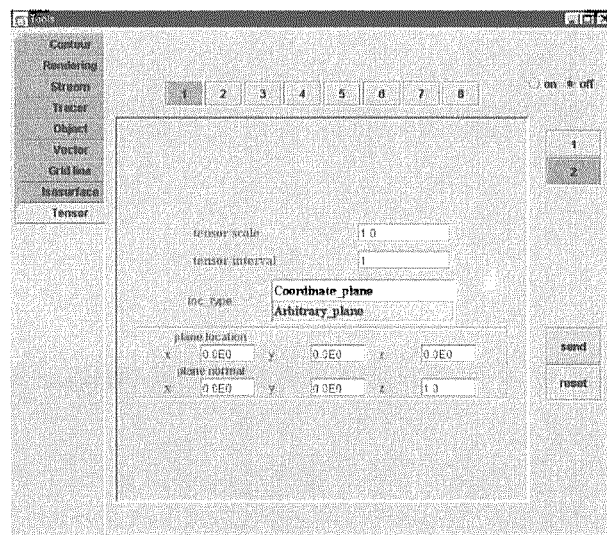


Fig. B.4 Screenshot for Tensor Mapping Parameter of Arbitrary Lattice

パラメタの指定が終了したら、右下の Send ボタンを押すことにより、パラメタがサーバに送信される。

B.2 カラーバー表示機能の使用方法

コンタ等の図種で表示された色とその数値との対応関係を示すためにカラーバーを表示する。カラーバーは縦表示および横表示が可能であり、スケールを変更することにより大きさを調整できる。また、カラーマップの最大値、最小値、中間値、図種名をカラーバー付近に文字表示する。文字については文字色およびフォントサイズを変更可能である。

B.2.1 パラメタ指定方法

パラメタ設定ファイル (user.para 等) の記述例を以下に示す。

【記述例】

```
Colorbar_setup true
```

```
Colorbar Contour Horizontal Small 255 255 0 0.2 0.2 0.7
```

パラメタフォーマットを以下に示す。

【記述例】

```
Colorbar_setup true
```

【フォーマット】

型	内容	上の記述例での値
文字	表示のオン・オフ {true false}	true

【記述例】

```
Colorbar Contour Horizontal Small 255 255 0 0.2 0.2 0.7
```

【フォーマット】

型	内容	上の記述例での値
文字	カラーテーブルを参照する図種名	Contour
文字	表示方法 {Horizontal Vertical}	Horizontal
文字	フォントサイズ {Small Medium Large}	Small
整数	最大値、最小値を表示する文字色 RGB	255 255 0
実数	表示位置	0.2 0.2
実数	スケール	0.7

【注釈】

図種名は、カラーマッピングに対応している図種名を 1 個指定する。以下のいずれかを指定する。

```
Contour, Vector, Stream_line, Tracer, Object, Tensor
```

カラーマッピングの最大値、最小値および中間値を表示する文字について、フォントの色とサイズを指定する。サイズは Small が 6x12, Medium が 10x20, Large が 16x32 である。表示位置は、図が表示されている画面の左上が (0.0,0.0) 右下が (1.0,1.0) としたときの 2 次元座標で指定する。

B.2.2 GUI 操作方法

クライアント・サーバモードでの基本的な操作方は、既存のクライアントに準じる。ただし、テンソル表示に対応していない旧バージョンのクライアントとは互換がないので注意されたい。

接続が確立したら、

- (1) view タブをクリックし、
- (2) Colorbar ボタンを押すと、以下のようなカラーバーパラメタ設定画面が表示される。

変更したいパラメタを変更した後、右下の Send ボタンを押すとパラメタがサーバに送信される。

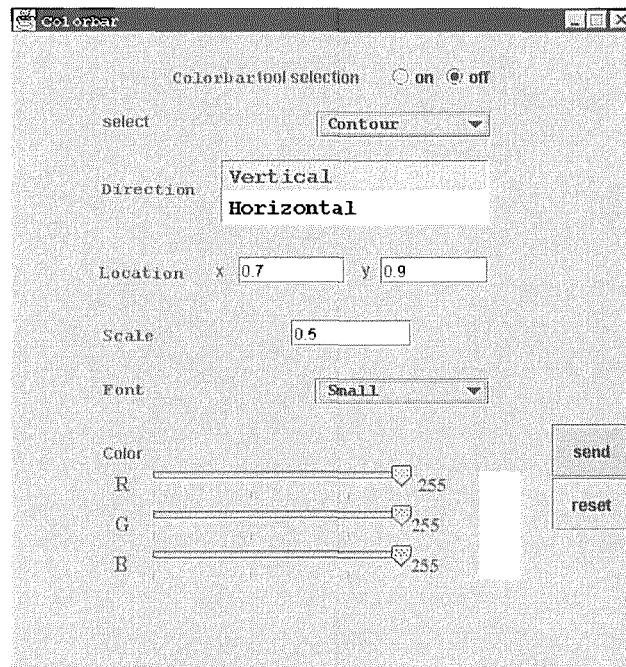


Fig. B.5 Screenshot of Colorbar Mapping Parameter

付録C テンソル量表示のプログラマーズ・マニュアル

C.1 RVS_INIT の仕様

C.1.1 機能

PATRAS サーバ機能を初期化する。

C.1.2 呼び出し形式

```
CALL RVS_INIT(NG,NS,NV,NT,LABEL,IMODE,IERR)
```

C.1.3 引数

引数名	型	サイズ	入出力	内容
NG	I4	1	入力	全ブロック数
NS	I4	1	入力	スカラデータ数
NV	I4	1	入力	ベクトルデータ数
NT	I4	1	入力	テンソルデータ数
LABEL	A*15	NS+NV+NT	入力	物理量種別名
IMODE	I4	1	入力	実行モード
IERR	I4	1	出力	エラーコード

C.1.4 補足

● NG

計算格子としてマルチブロック格子を用いる場合、NG にはブロック数を指定する。シングルブロック格子を用いる場合は、1 を指定する。本システムはマルチブロック格子をサポートしていないため、必ず 1 を指定する。

● NS, NV, NT

可視化する対象として解析プログラムから PATRAS サーバへ渡されるスカラ/ベクトル/テンソルデータの数。

● LABEL

LABEL には、各スカラデータおよびベクトルデータの意味内容を表すラベルを 15 文字以下の文字列で指定する。なお、ラベルは空白であってはならない。スカラデータ、ベクトルおよびテンソルデータに対するラベルを RVS_MAIN ヘデータを渡す順序で、(NS+NV+NT) 個分入力する。本システムの GUI 上でどのデータを可視化の対象とするかをここで指定したラベルの中から選択できるようになる。

- IMODE

IMODE が 0 の場合、バッチ処理モードで起動される。ユニバーサルシナリオファイルに従って可視化処理が行われ、画像ファイルが作成される。IMODE が 1 の場合、分散処理モードで起動される。クライアントサーバ型の処理が行われ、クライアント側から随時パラメータを変更することができる。

- IERR

エラーコード

C.2 RVS_HEX1 の仕様

C.2.1 機能

非構造六面体格子上に定義された数値データを可視化の対象とする場合の格子データおよび解析結果の配列のアドレスを引き渡す。

C.2.2 呼び出し形式

CALL RVS_HEX1(IG,MNOD,MELM,NNOD,NELM,CORD,IELM,NS,SCAL,NV,VECT,NT,TENS,IERR)

C.2.3 引数

引数名	型	サイズ	入出力	内容
IG	I4	1	入力	入力ブロック番号
MNOD	I4	1	入力	節点数に関する整合寸法
MELM	I4	1	入力	要素数に関する整合寸法
NNOD	I4	1	入力	全節点数
NELM	I4	1	入力	全要素数
CORD	R8	(3,MNOD)	入力	節点の X, Y, Z 座標値
IELM	I4	(8,MELM)	入力	要素内構成節点番号
NS	I4	1	入力	スカラデータ数
SCAL	R8	(MNOD,NS)	入力	スカラデータ
NV	I4	1	入力	ベクトルデータ数
VECT	R8	(MNOD,3,NV)	入力	ベクトルデータ
NT	I4	1	入力	テンソルデータ数
TENS	R8	(MNOD,6,NT)	入力	テンソルデータ
IERR	I4	1	出力	エラーコード

C.2.4 補足

- IG 各格子ブロックに対して、固有の番号付けを行う。この番号は、1 から全格子ブロック数 NG の範囲で指定する、本システムはマルチブロック格子をサポートしていないため IG として必ず 1 を指定する。

- MNOD, MELM 節点の物理座標値、要素内構成節点番号、および節点上での解析結果（スカラ／ベクトル／テンソルデータ）を格納するための配列（CORD, IELM, SCAL, VECT, TENS）の整合寸法を指定する。
- NNOD, NELM それぞれ、全節点数と全要素数を指定する。
- NS, NV, NT 可視化する対象として解析プログラムから PATRAS サーバへ渡されるスカラ／ベクトル／テンソルデータの数を指定する。
- CORD, IELM, SCAL, VECT, TENS 節点の物理座標値、要素内構成節点番号および節点上での解析結果（スカラ／ベクトル／テンソルデータ）を格納する。これらは整合配列である。CORD(1,i), CORD(2,i), CORD(3,i) には、i 番目の節点の物理座標値を格納する。IELM(1,i), IELM(2,i), IELM(3,i), ..., IELM(8,i) には、i 番目の六面体要素を構成する 8 つの節点の番号を格納する。SCAL(i,n) には、i 番目の節点上での n 番目のスカラデータのデータ値を格納する。VECT(i,1,n), VECT(i,2,n), VECT(i,3,n) には、i 番目の節点上での n 番目のベクトルデータの X 成分, Y 成分, Z 成分を格納する。TENS(i,1,n), TENS(i,2,n), TENS(i,3,n), TENS(i,4,n), TENS(i,5,n), TENS(i,6,n) には、i 番目の節点上での n 番目のテンソルデータの S11 成分, S22 成分, S33 成分, S12 成分, S23 成分および S13 成分を格納する。S11 ~ S13 成分はテンソルを 9 次元で表した際の右上の成分である。
- IERR エラーコードの詳細については、マニュアルを参照のこと。
- 解析ソルバの節点座標値、要素内構成節点番号、およびスカラ／ベクトル／テンソルデータ格納形式が本システムの形式と異なる場合、RVS_HEX1 を呼ぶ前に値の並び換えを行う必要がある。
- 本サブルーチンは、RVS_BFC, または RVS_TET1 と同時に呼び出すことはできない。

C.3 作業領域のサイズ

テンソル表示機能で使用する作業領域のサイズは、以下の通りである。ここで、Viewport_size はビューポートサイズである。

C.3.1 引数

作業領域	表示条件	格子条件	必要な大きさ
整数作業領域	座標面表示	BFC 格子	$\max(\text{NI}*\text{NJ}, \text{NJ}*\text{NK}, \text{NK}*\text{NI})$
	任意断面表示	BFC 格子	0
		非構造格子	$\text{Viewport_size}2+4*\text{Viewport_size}$
実数作業領域	座標面表示	BFC 格子	$6*\max(\text{NI}*\text{NJ}, \text{NJ}*\text{NK}, \text{NK}*\text{NI})$
	任意断面表示		0

C.4 rvs_colorbar の仕様

C.4.1 機能

カラーバー描画ルーチン(C言語)である。

C.4.2 呼び出し形式

```
call rvs_colorbar_(int *iflags_colorbar, int *itable_colorbar,
                  SolverReal *rtable_colorbar, int *irgb, int *ierr)
```

C.4.3 引数

引数名	型	サイズ	入出力	内容
iflags_colorbar	I4	18	入力	カラーバー表示用パラメータ
itable_colorbar	I4	264	入力	カラーバー表示用パラメータ
rtable_colorbar	R8	5	入力	カラーバー表示用パラメータ
IRGB	I4	*(*)	出力	出力画像の RGB 値
IERR	I4	1	出力	エラーコード

C.4.4 補足

- SolverReal 型は double に typedef されている。

付録D マルチプラットフォーム化

D.1 マルチプラットフォーム化の基本的考え方

ITBL (IT-Based Laboratory) の成果を基に、原研内の各種実行計算機 (SP3, SR8000F1, VPP5000, Origin3800, SC/ES40, SR2201) へ、SX-4 上で開発した新機能を移植している。基本的な考え方は以下のとおりである。

- ソースコードに対して、異種依存を吸収するようなコーディングに変更する。
- 各機種のコmpイル・リンクオプションについては、マスタとするシステムファイル (Makefile) にコメントを付けて残す。

なお、今回対応するのは、サーバー機能のみで、クライアント側は対象としていない。

D.2 ソースコードの修正

D.2.1 関数名 (サブルーチン名) に対する処理

PATRAS のソースコードは、Fortran と C が混在している。このため、移植時に Fortran から呼び出される C の関数名、または C から Fortran のサブルーチンを呼び出すサブルーチン名が各種計算機で異なることが問題となる (Table D.2.1)。

Table D.1 各機種ごとの C の関数名と Fortran のサブルーチン名

機種	C の関数名	Fortran のサブルーチン名
SP3	csub()	fsub()
SR8000F1, SR2201, T3E	CSUB()	FSUB()
VPP5000, SC/ES40, Origin3800	csub_()	fsub_()
その他	csub__()	fsub__()

これらを統一的に扱うために以下のように修正することにした。

まず、include/types.h で、各機種ごとのマクロ定義をする (Table D.2.1)。マクロ定義は、以下のとおりとする。

```
#if defined(_AIX)
#   define FNAME_NO_UNDERSCORE
#elif defined(_SX) || defined(__ALPHA) || defined(__VPP5000)
#   define FNAME_UNDERSCORE
#elif defined(__SR8000)
#   define FNAME_CAPS
#else
#   define FNAME_DOUBLE_UNDERSCORE
#endif
```

Table D.2 各機種ごとのマクロ定義名

機種	マクロ定義名
SP3	FNAME_NO_UNDERSCORE
SR80001F1, SR2201, T3E	FNAME_CAPS
VPP5000, SC/ES40, Origin3800	FNAME_UNDERSCORE
その他	FNAME_DOUBLE_UNDERSCORE

次に, src/visualiser/graph/rvs_tex1_isosurface.c を例として, 関数 (サブルーチン) 名に対する処理を説明する.

まず, include/types.h を組み込む.

関数 rvs_tex1_isosurface は, Fortran のサブルーチン rvs_grmain_fem から呼ばれているので, 関数名を以下の用に修正する.

```
#if defined(FNAME_CAPS)
    RVS_TEX1_ISOSURFACE
#elif defined(FNAME_NO_UNDERSCORE)
    rvs_tex1_isosurface
#elif defined(FNAME_UNDERSCORE)
    rvs_tex1_isosurface_
#elif defined(FNAME_DOUBLE_UNDERSCORE)
    rvs_tex1_isosurface__
#endif
```

また, rvs_tex1_isosurface は, 内部で Fortran のサブルーチン rvs_tex1_iso_alloc を呼び出している所以, 呼び出し部分を以下のように修正する.

```
#if defined(FNAME_CAPS)
    RVS_TEX1_ISO_ALLOC
#elif defined(FNAME_NO_UNDERSCORE)
    rvs_tex1_iso_alloc
#elif defined(FNAME_UNDERSCORE)
    rvs_tex1_iso_alloc_
#elif defined(FNAME_DOUBLE_UNDERSCORE)
    rvs_tex1_iso_alloc__
#endif
```

以下, 同様に Fortran から呼ばれる C の関数名と, C の関数内から呼ばれる Fortran のサブルーチン名を修正する.

```
src/ui/client/initialise_client_standalone.c
src/ui/server/alloc_memory.c
src/ui/server/alloc_memory_fem.c
src/ui/server/invoke_grmain.c
src/ui/server/library_style.c
src/ui/server/nasrvsui.c
src/ui/server/server_time.c
src/visualiser/draw/create_image_rmv.c
src/visualiser/draw/image2file.c
src/visualiser/draw/image2file.h
src/visualiser/draw/simufb.c
src/visualiser/draw/writetoclient.c
src/visualiser/draw/writetoclient.h
```

```

src/visualiser/graph/ISOSRF.c
src/visualiser/graph/action_mapper.c
src/visualiser/graph/grmain_mb.c
src/visualiser/graph/minmax_mb.c
src/visualiser/graph/polygon_c.c
src/visualiser/graph/rvs_text1_isosurface.c
src/visualiser/graph/rvs_user_object_client.c
src/visualiser/graph/rvs_user_object_client.h
src/visualiser/visio/transform_coord.c
src/flowsolver/flowsolver2.1/user_fem.c.c

```

D.2.2 MPI-2 への対応

SR8000F1, Origin3800 の MPI は、MPI-2 対応になっている。この MPI-2 対応では、コミュニケータ値の変数の型が C では構造体、Fortran では整数型と異なりおり、Fortran で取得したコミュニケータ値を C に受け渡して、そのまま使用しても動作しない（MPI では、コミュニケータ値の変数の型が C, Fortran とともに整数型で、値を受け渡して使用しても問題はない）。今回は、C と Fortran でのコミュニケータ値の受け渡しを行わず、MPI_COMM_WORLD をデフォルトのコミュニケータ値として使用するよう修正する。

```

src/visualiser/graph/ISOSRF.c
src/visualiser/graph/ISOSRF_EXEC.f
src/visualiser/graph/grmain_mb.c

```

D.2.3 インクルードファイルの有無

SR8000F1, SR2201 には、sys/select.h が存在しないので、SR8000F1, SR2201 のときは sys/select.h を外し以下のように修正する。

```

#ifndef __SR8000
#ifndef __SR2201
#include <sys/select.h>
#endif
#endif

```

該当するファイルは、以下のとおりである。

```

genlib/src/rpc/rpc/rpc.c
src/ui/server/server_connection_state.c

```

D.2.4 型宣言の二重定義

SR8000F1, SP3 では、二重に型宣言がされているとエラーになるので、二重に宣言しないように以下のように修正する。int8, int16, int32 は、SR8000F1, SP3 両方とも、u_int8, u_int16, u_int32 は、SP3 のみ、uint8, uint16, uint32 は、SR8000F1 のみである。

```

#ifndef _AIX
#ifndef __SR8000
typedef char          int8;
typedef short        int16;
typedef int          int32;
#endif
typedef unsigned char u_int8;
typedef unsigned short u_int16;
typedef unsigned int  u_int32;
#endif

```

該当するファイルは、以下のとおりである。

```
genlib/src/rpc/rpc/rpc.h
src/visualiser/draw/jfif.h
```

D.3 計算機個別の対応内容

ここでは、性能評価に関連した並列計算機である SR8000 と RS/6000 SP について記述する。なお、SR8000 は、クロスの環境でのライブラリ作成方法についても記述している。

D.3.1 SR8000F1

Makefile のコンパイルオプションの `-hansi` を `-ansi` に変更する。

```
genlib/src/rpc/amunix_lib/Makefile.super
src/visualiser/globals/Makefile
src/visualiser/draw/Makefile.server
src/visualiser/graph/Makefile
src/visualiser/misc/Makefile
src/visualiser/visio/Makefile
```

Fortran の最適化オプションとして、`-Oss -noparallel` を適宜追加した。

```
src/visualiser/draw/Makefile.server
src/visualiser/graph/Makefile
```

lex は、オプション `-l` を追加する。

```
genlib/src/log-n-replay/Makefile
src/usf/parse/server/Makefile
```

`src/usf/parse/server` にある `lex.yy.c` と `y.tab.c` は、それぞれ同じディレクトリにある `lex.yy.c.sr8000`、`y.tab.c.sr8000` と置き換える。

```
src/usf/parse/server/lex.yy.c
src/usf/parse/server/y.tab.c
```

PATRAS では、X ライブラリを使用しており、そのライブラリ名を XLIB でマクロ定義している。その XLIB に `-lm` を追加する（以下、各機種での X ライブラリ名のマクロ定義名を XLIB とする）。

```
src/ui/server/Makefile
```

以上の修正が、コンパイル等を SR8000 本体で実行する場合であり、フロントエンドの計算機で実行する場合、以下の点を変更する必要がある。

- コンパイラの名の変更：`xmpif90`、`xmpicc`、`xf90`、`xcc`
- ar の名の変更：`xar`
- lex 名の変更：`flex`

D.3.2 SP3

SP3 システムでは Fortran および, MPI 用 Fortran, MPI 用 C のコンパイラは, それぞれ xlf, mpixlf, mpcc とする.

```
src/flowsolver/flowsolver2.1/Makefile
src/visualiser/globals/Makefile
src/visualiser/draw/Makefile.server
src/visualiser/graph/Makefile
src/visualiser/visio/Makefile
src/ui/server/Makefile
```

Fortran のコンパイルオプションは, -cpp を削除して, 最適化のオプション (-qarch=pwr3 -qtune=pwr3 -O3 -qstrict を適宜追加している.

```
src/visualiser/draw/Makefile.server
src/visualiser/graph/Makefile
```

C のコンパイルオプションは, -qlanglvl=ansi -qarch=pwr3 を加えて, -hansi, -O, nooverlap を削除する. また, -hsignc は, -qchars=signed に変更する.

```
genlib/src/rpc/amunix_lib/Makefile.super
genlib/src/ui-all/Makefile
genlib/src/ui-gm/Makefile.super
src/visualiser/globals/Makefile
src/visualiser/draw/Makefile.server
src/visualiser/graph/Makefile
src/visualiser/misc/Makefile
src/visualiser/visio/Makefile
src/usf/misc/server/Makefile
src/usf/parse/server/Makefile
src/usf/replay/server/Makefile
src/usf/translate/server/Makefile
src/ui/server/Makefile
```

.F ファイルのコンパイル時に, .F をサフィックスとして認識しないので, Makefile でサフィックスを .SUFFIXES: .o .c .f .F と再定義し, オプションを-WF, -DLANGUAGE_FORTRAN にする.

```
src/visualiser/graph/Makefile
```

all clean fclean: のように Makefile に複数のターゲットが記述されているものは, 全てが実行されるので分割する.

```
src/usf/Makefile.server
```

XLIB は, -lSM -lICE を削除する.

```
src/ui/server/Makefile
```

src/usf/parse/server にある lex.yy.c と y.tab.c は, それぞれ同じディレクトリにあるか lex.yy.c.sp3, y.tab.c.sp3 か lex.yy.c.sr8000, y.tab.c.sr8000 と置き換える.

```
src/usf/parse/server/lex.yy.c
src/usf/parse/server/y.tab.c
```

src/visualiser/globals/global_table_defines.h は, Fortran ソースへのインクルード時に C のコメント文がエラーになるので, 同じディレクトリにある global_table_defines.h.sp と置き換える.

```
src/visualiser/globals/global_table_defines.h
```

D.4 ライブラリの作成方法

D.4.1 make 方法

- (1) 'NASRVS_WD' 環境変数をインストールディレクトリに設定する.

例えば, 以下のとおり.

```
> setenv NASRVS_WD /home/patras
```

- (2) インストールディレクトリ (/home/patras) 下で, 'make setup' を実行する.

```
> make setup
make -f Makefile_all setup
Enter the name of the "EWS" host system: <CR>
Enter the name of the "SGI" host system: <CR>
Enter the name of the SUPER host system: ne00011<CR>
Enter the name of the RS/6000 SP host system: <CR>
Enter the name of the SR8000 host system: <CR>
```

ここで, '>' はプロンプト, '<CR>' は Carrage Return を示している. 上記は, SX-4 の場合を示しているが, RS/6000 SP と SR8000 には, それぞれ imbsp10a や sr8fp1 を指定する.

- (3) サーバモジュールを make する.

```
> make server
```

make 結果は, LOG.server に出力される. 作成されるサーバモジュールは, 以下のとおりである.

```
bin/bin.super/nasrvsui-server
```

- (4) 可視化ライブラリ (アーカイブファイル) を作成する.

```
> cd src/ui/server
> make archive
```

これにより, lib/libRVS_D.a が作成される.

nasrvsui-server が, テストラン用の実行モジュールとなる.

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	t
電子ボルト	eV
原子質量単位	u

1 eV = 1.60218 × 10⁻¹⁹ J
1 u = 1.66054 × 10⁻²⁷ 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

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·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·m
工率, 放射束	ワット	W	J/s
電気量, 電荷	クーロン	C	A·s
電位, 電圧, 起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンズ	S	A/V
磁束	ウェーバ	Wb	V·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	ルーメン	lm	cd·sr
照射度	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4 SIと共に暫定的に維持される単位

名称	記号
オングストローム	Å
バ - ン	b
バ - ル	bar
ガ - ル	Gal
キュリ -	Ci
レントゲン	R
ラ - ド	rad
レ - ム	rem

1 Å = 0.1 nm = 10⁻¹⁰ m
1 b = 100 fm² = 10⁻²⁸ m²
1 bar = 0.1 MPa = 10⁵ Pa
1 Gal = 1 cm/s² = 10⁻² m/s²
1 Ci = 3.7 × 10¹⁰ Bq
1 R = 2.58 × 10⁻⁴ C/kg
1 rad = 1 cGy = 10⁻² Gy
1 rem = 1 cSv = 10⁻² Sv

(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ない。

換算表

力	N (=10 ⁵ 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/m²) = 10 P (ポアズ) (g/(cm·s))
動粘度 1 m²/s = 10⁶ St (ストークス) (cm²/s)

圧	MPa (=10 bar)	kgf/cm ²	atm	mmHg (Torr)	lbf/in ² (psi)
	1	10.1972	9.86923	7.50062 × 10 ³	145.038
力	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

エネルギー・仕事・熱量	J (=10 ⁷ 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 J (熱化学)
	9.80665	1	2.72407 × 10 ⁻⁶	2.34270	9.29487 × 10 ⁻³	7.23301	6.12082 × 10 ¹⁹	= 4.1855 J (15 °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 J (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 kgf·m/s
	1.35582	0.138255	3.76616 × 10 ⁻⁷	0.323890	1.28506 × 10 ⁻³	1	8.46233 × 10 ¹⁸	= 735.499 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
	1	2.70270 × 10 ⁻¹¹
	3.7 × 10 ¹⁰	1

吸収線量	Gy	rad
	1	100
	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

PATRASS 特殊インク用モニター用プリント用紙 CHIKAKU DYNAMIC の適用・評価



再生紙配合率100%
白色度70%再生紙を使用しています