JAERI-Tech
96-044

リアルタイムモニターシステムの開発

1996年10月

加藤克海* ・渡辺　正・蕪木英雄

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

本レポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-11 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（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　日立高速印刷株式会社

JAERI-Tech 96-044

リアルタイムモニターシステムの開発

日本原子力研究所計算科学技術推進センター
加藤　克海*・渡辺　　正・蕪木　英雄

（1996年10月1日受理）

計算科学技術推進センターでは，原子力分野における計算科学の研究の一環として大規模数値シミュレーション技術の研究を進めており，シミュレーション結果の効率的な理解のための可視化並びに動画像処理技術に関する研究・開発を行なっている。本報告書では，スーパーコンピュータや計算サーバ用ワークステーションで実行中のシミュレーションの途中経過を，逐次，可視化処理サーバ用ワークステーションへ転送し，実時間可視化とビデオ録画を行なうリアルタイムモニターシステムについて記述する。本システムの開発に用いた機器構成は，ネットワークに接続された可視化用ワークステーションとビデオ装置であり，制御用シェルプログラムは，スーパーコンピュータ上でのシミュレーションのためのジョブサブミット用シェル，可視化用出力ファイルの転送用シェル，可視化ツールを起動するための描画用シェルからなる。本システムは，特殊な画像処理技術や装置を必要とせず，汎用可視化ツールAVSと通常のUNIXコマンドにより構築されているため，様々な計算機環境に応じた使用，拡張が可能である。

東海研究所：〒319－11　茨城県那珂郡東海村白方白根2－4
＊高度情報科学技術研究機構

JAERI-Tech 96-044

Development of the Real Time Monitor System

Katsumi KATO *, Tadashi WATANABE, and Hideo KABURAKI

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

(Received October 1, 1996)

Large-scale simulation technique is studied at the Center for Promotion of Computational Science and Engineering (CCSE) for the computational science research in nuclear fields. Visualization and animation processing technique are studied and developed for efficient understanding of simulation results. The real time monitor system, in which on-going simulation results are transfered from a supercomputer or workstation to a graphic workstation and are visualized and recorded, is described in this report. This system is composed of the graphic workstation and the video equipment connected to the network. The controll shell programs are the job-execution shell for simulations on supercomputers, the file-transfer shell for output files for visualization, and the shell for starting visualization tools. Special image processing technique and hardware are not necessary in this system and the standard visualization tool AVS and the UNIX commands are used, so that this system can be implemented and applied in various computer environments.

Keywords: Simulation, Visualization, Real Time Monitor, AVS

---

* Research Organization for Information Science & Technology

目　　次

Contens

1. はじめに

日本原子力研究所・計算科学技術推進センターでは，原子力分野で見られる伝熱・流動現象，材料の物性変化などの複雑な現象を，基本的なモデル及び方程式を用いた計算機シミュレーションを通して解明する計算科学研究を進めている．計算科学研究では，しばしばスーパーコンピュータや計算サーバ用ワークステーションを用いて長時間のシミュレーションを行なうが，近年の並列計算機の発展により，シミュレーションはより大規模（大容量）なものになってきている．大規模シミュレーションでは，理想的な条件下であらゆる変数の数値データが得られる反面，その膨大なデータの処理，保存が問題となっている．例えば，等方性乱流の3次元計算では，一つのタイムステップで渦度の3方向成分が計算格子の点数だけ得られるが，256^3 の格子点を用いた計算では[1, 2]，これは $200MB$ に達する．また，直接シミュレーションモンテカルロ法による3次元対流計算において1000万個の粒子を使用すると[3]，粒子の位置と速度だけで $500MB$ になる．過渡変化が重要となる場合に時系列データを保存しようとすると，タイムステップ数だけこれらのデータが必要となる．並列計算機を用いることにより，計算速度の向上と同時に計算に使用できる変数の量，すなわちメモリーの容量も増加しており，計算結果を理解，判断するための効率的なデータ処理技術，あるいは大規模数値計算支援技術の研究・開発はますます重要なものとなってきている．

シミュレーションにもとづく計算科学研究において，データ処理技術として最も重要と考えられるものの一つは可視化技術である．ネットワーク技術の高度化とワークステーションやパーソナルコンピュータの高性能化により，シミュレーション結果の可視化は，高度の作画機能を持つワークステーション上において可視化ツールやユーティリティを用いて行なうということが普及してきた[4, 5]．可視化の結果は，ワークステーションの画面で直接見たり，あるいは論文等への使用のためにプリントするばかりでなく，時系列データを連続的に可視化表示したアニメーションとして，過渡現象の動的な理解や効果的なプレゼンテーションのためにも利用される[6, 7]．さらに，シミュレーション結果の保存方法としてアニメーションビデオを作成することも行なわれており，動画像処理技術も計算科学研究には欠くことのできないものである．

可視化技術や動画像処理技術は，これまではシミュレーション結果に対するデータ処理技術として研究・開発が行なわれてきた．また，実際の計算科学研究の手順としても，シミュレーションはセンターマシンとしてのスーパーコンピュータや高計算性能の計算サーバ用ワークステーションで実行し，計算終了後に結果をグラフィックワークステーションやパソコン等に転送し，可視化，アニメーション作成を行なうのが一般的である．これは，スーパーコンピュータがフロント／バックエンドとして構成されていたり，計算をバッチジョブとしてサブミットする形式を採用しているため，計算の開始や終了が任意ではなく，ファイル操作を含めてパソコンのように使い勝手が良くないためである．しかしながら，計算科学研究で行なわれるシミュレーションは長時間に及ぶことが多いため，可視化による結果の理解が効率良く行なわれるとしても，シミュレーション終了後にのみデータ処理や可視化を行なっていては，実質的なシミュレーション研究の効率を上げることはできない．シミュレーション研究の効率をあげ，計算機資源の有効利用を計るためには，パソコンのように計算を実行しながらその結果を描画し，任意に計算を中断し，パラ

メータを変更し再実行する，ということがスーパーコンピュータを用いた大規模シミュレーションにおいても必要となる [7].

本報告書では，スーパーコンピュータや計算サーバ用ワークステーションで実行中のシミュレーションの途中経過を，逐次，画像処理サーバ用ワークステーションへ転送し，実時間可視化を行なうリアルタイムモニターシステムについて記述する．本システムは，特殊な画像処理技術や装置を使用せず，汎用可視化ツール AVS と一般の UNIX コマンドにより構築されているため，様々な計算機環境に応じた使用，拡張が可能である．第2章では，本システムの機器構成について述べ，第3章では，それら機器と AVS を用いたスナップショットの描画方法について説明する．第4章では，本システムをスーパーコンピュータ上で実行中のシミュレーションジョブと連動させ，連続的な実時間可視化（リアルタイムモニター）を行なうためのシェルプログラムについて解説する．第5章では，シェルプログラムの実行と本システムでのビデオ録画について述べる．

2. システム機器構成

実時間可視化（リアルタイムモニター機能）を行なった処理システムの機器構成を Fig.2.1に示す.

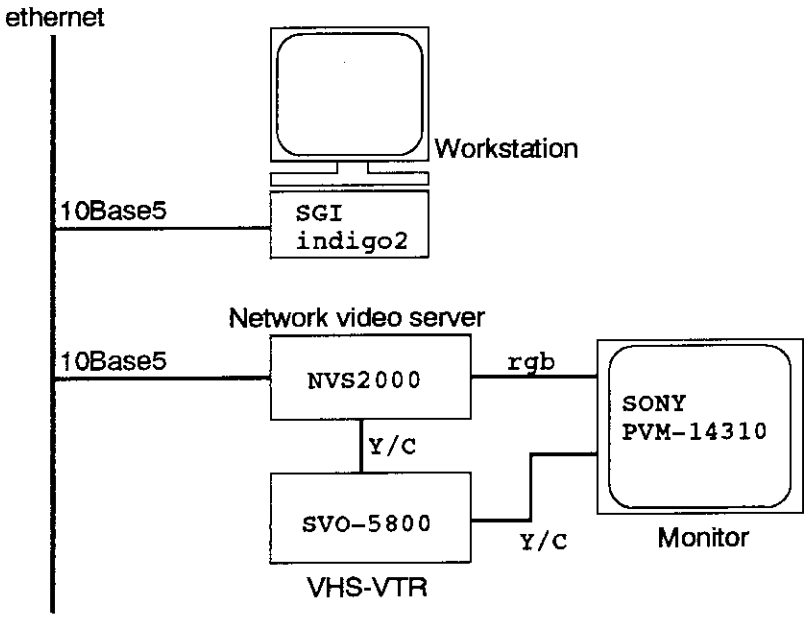

Fig. 2.1 System Hardware Construction

Fig.2.1に示されているように，可視化用のワークステーションは SGI Indigo^2 （主メモリ：384MB，ディスク容量：内蔵 4GB，外付 9GB）である．汎用可視化ツール AVS は，このワークステーション上で作動する.

ネットワークビデオサーバは，（株）計算流体力学研究所製 NVS2000 である．ワークステーションから ethernet 経由で送られてくるビデオ信号の解像度，ビーム走査（スキャン）スピードを変換する，またビデオ機器の制御も行なう.

VTR は，SONY 製 SVO-5800 である．モニターは，おなじく SONY 製カラービデオモニター PVM-14310 である.

このシステムにおいて，ワークステーション上で AVS を用いてシミュレーション結果を実時間で描画し，VTR に録画する．ワークステーション上の AVS には NVS2000 制御用のモジュールが含まれている.

リアルタイムモニター機能を利用するにあたっての機器の構成は，Fig.2.1に限定されるものではない．すなわち，可視化用ワークステーションは UNIX マシンであれば良く，また，AVS

から制御できるフレームスキャンコンバーターやビデオ装置であれば，実時間録画は問題なく行なうことができる[6, 7]．また，本システムを実時間可視化のみを行なうモニター装置とし，録画を想定しなければ，ビデオサーバーとVTRは必要がない．

Fig.2.1において，ethernetの先には，計算サーバ用ワークステーションや，大型計算機システムが接続されている．原研のスーパーコンピュータシステムとしては，東海研のベクトル並列機富士通VPP500（42プロセッサ，1.6GFlops × 42，256MB × 42），モンテカルロ計算のためのパイプラインを備えた高速モンテカルロ装置MONTE4（4プロセッサ，1.6GFlops × 4，512MB），那珂研のスカラー並列機Intel Paragon（256プロセッサ，75MFlops × 256，32MB × 256），等が使用可能である．いずれのスーパーコンピュータシステムもジョブをサブミットする形式を採用しており，第4章に記述するシェルプログラムによりリアルタイムモニター機能を利用することができる．

3. ワークステーション上での描画

本章ではAVSを用いたワークステーション上での描画についての概略を記述する．また，アニメーション作成例として，セルラオートマタ法により2相流の相分離をシミュレートする際の2次元密度分布データの描画について取り上げる．

3.1 可視化ツール AVS

可視化ツール AVS（Application Visualization System）は米国AVS（Advanced Visual Systems）社が開発した汎用可視化・コンピュータグラフィックス用ソフトウェアである[4, 5]. AVSはシミュレーション結果や実験データなどの数値データを，直接プログラミングすることなく容易に可視化することができる．ここでは，モジュールと呼ばれる処理単位を組み合わせて必要な画像処理の一連の作業を構成し，そのモジュールの組合せ（ネットワークと呼ばれる）を順次実行することによって，インタラクティブにデータの読み込み，処理，軸やラベルの設定，色指定，画像ファイルの作成などを行なう．また，連続的な描画や自動的な画像処理などを行なうために，AVSの実行をCLI（Command Language Interpreter）と呼ばれる機能により制御することができる．モジュール，ネットワークの詳細については，参考文献[8, 9, 10]を参照のこと．

3.2 AVSの起動

AVSの起動はコマンドavsにより行なうが，実時間での描画を実現するために，AVSを起動する際に`-server`オプションを用いる．このオプションを用いることにより，インタラクティブではなくAVSを起動とは別のウィンドウから制御することができる．第4章に示したシェルプログラムにはこの別ウィンドウから制御するためのコマンドを記述している．

なお，可視化用ワークステーション本対のコンソールではなく，X端末や他のワークステーションから可視化用ワークステーションにアクセスしAVSの起動を行なっている場合は，さらに`-nohw`オプションを用いる必要がある．これらのオプションを使用するにあたっては，オプションのコマンドが用意されているディレクトリを，コマンドサーチパスとして指定しておくと便利である[11]．ここでは，.cshrcに以下のようなパスを追加指定している．

```
set path=($path /usr/local/nvs/bin /usr/avs /usr/avs/examples)
```

/usr/local/nvs/bin はビデオ制御関係のコマンド，/usr/avs はAVS本体関係，
/usr/avs/examples はAVSのオプション関係のコマンドのディレクトリである．

3.2.1 AVS server オプション

AVSを別プログラムから制御するための方法で，AVSをサーバとして起動し，クライアントからCLIコマンドを送る方法である．以下にこれらの手順を述べる．

1. AVS サーバの起動

AVS をサーバとして起動する方法は，avs 起動時に，-server オプションを使用する．以下のように（> はキーボードからの入力を示す），サーバのプロセス ID <pid> と通信に使用するポート ID <port id> が表示される．またこの時に，/tmp/avs_server.<pid> というファイルも作成される．このファイルには，<port id> が書き込まれている．

-server オプション：

```
>  avs -server
    AVS server process is <pid>,port is:<port id>
```

なお，AVS サーバは，マルチ・クライアント・サーバではないため，複数のクライアントを同時には実行できない．

2. クライアント・プログラム avs_client の実行

サーバとクライアントが同一ホストか異なるホストかによって 2 通りある．

(a) サーバとクライアントが同一ホストの場合

```
>  avs_client <pid>
```

または，環境変数を使用して，

```
> setenv AVS_SERVER_PROCESS <pid>
> avs_client
```

(b) サーバとクライアント異なるホストの場合

```
> avs_client <port id> <avs サーバのホスト名>
```

3. CLI コマンドの送信

クライアント・プログラム avs_client では，標準入力から CLI コマンドを入力する．入力プロンプト client> が表示されたら，CLI コマンドが入力できる．

実行例：

```
client> parm_set ''file browser.user.24'':''File Browser'' c128test
client> quit
```

3.3 2次元データの描画

ここでは，セルラオートマタ法により2相流の相分離をシミュレートする例で用いた2次元密度分布データの描画用のAVSネットワークと描画例を示す．セルラオートマタ法は，連続体である流体の微小要素（流体粒子）を計算領域内の格子上で動かし，格子の粒子情報をサンプリング（粗視化）することによって巨視的な流れ場を求める手法であり，粒子法の一種である．ただし，ここで扱う粒子は，原子・分子そのものではなく，また，その運動は格子上に制限されている．ここで示すシミュレーション例は，はじめ均一に混ざっていた赤と青の2種類の粒子が，時間と共にそれぞれ凝集し，2相にわかれていくシミュレーションで，計算は東海研のVPP500で実行したものである．計算実行用シェル等は次章で解説する．

3.3.1 ネットワーク

2次元密度分布データの描画に使用したネットワークをFig.3.1に示す．

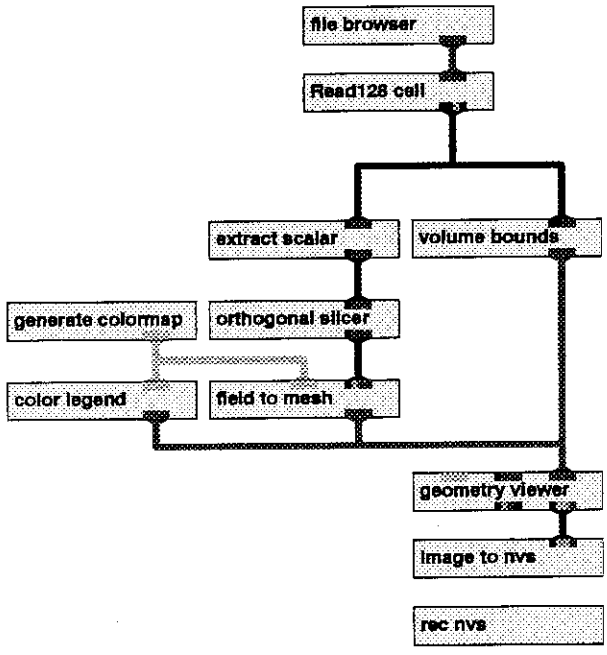

Fig. 3.1 Network for visualization of 2-D density distribution data

Fig.3.1のネットワークで使用しているモジュールの機能の概略は，以下の通りである．詳細は，文献[4, 5]または[8, 9, 10]を参照のこと．ビデオ制御関係のモジュール（image to nvs, rec nvs）に関する詳細は[12, 13, 14]を参照のこと．

- file browser （ファイルを選択する）

  描画するファイルを選択する.

- Read128 cell （描画データを読み込む）

  選択されたファイルから描画データを読み込む. このモジュールは2次元データ描画用に独自に作成した.

- extract scalar （ベクトル・フィールドからスカラー・フィールドを取り出す. ）

  本来は, ベクトルのフィールドの中から1つをスカラー値として取り出すモジュールであるが, 数種類のスカラーデータの中から1つを取り出すモジュールとして使用している.

- orthogonal slicer （座標軸に垂直な平面でボリュームデータを切断する）

  描画データは128×128の2次元であるが, 便宜上Read128 cellモジュールで128×128×3の3次元化している. この3次元の配列から2次元の断面を取り出している.

- field to mesh （2次元のスカラーフィールドを3次元の面に変換する. ）

  2次元のデータをgeometry viewerモジュールで描画可能な3次元の面に変換する.

- generate colormap （AVSカラーマップを出力する）

  カラー表示のためのカラーマップデータ構造を生成し出力する.

- color legend （描画ウィンドウ内に色とデータの対応関係を表示する）

  描画ウィンドウ内にカラーマップとデータ値を表示する. これにより, 色と数値の対応が一目で確認できる.

- geometry viewer （データを表示する）

  上段モジュールからのデータをもとにディスプレイ上に描画を行なう.

- image to nvs （画像データをNVS形式に変換する）

  上段モジュールからの画像データをNVS形式に変換し, 直接NVSHOST （NVS2000）に送出する.

- rec nvs （ビデオデッキを制御する）

  NVSコントロールコマンドを直接NVSHOSTに送出する. このコントロールコマンドにしたがってNVS2000はビデオデッキを制御する.

以下にネットワークファイルの内容を示す．このファイルは次章の可視化制御用シェルで読み込む /home/sigcs/watanabe/pepper/cellflow.net である．

```
#!/usr/bin/avs -network
version 5.02 (50.65 sg2)
module "geometry viewer.user.4" -xy 418,342
module "orthogonal slicer.user.5" -xy 258,192 -ex $Path/avs_library/mongo
module "field to mesh.user.6" -xy 258,262 -ex $Path/avs_library/mongo
module "volume bounds.user.8" -xy 418,142 -ex $Path/avs_library/mongo
module "extract scalar.user.9" -xy 258,142 -ex $Path/avs_library/mongo
module "generate colormap.user.11" -xy 98,192
module "color legend.user.12" -xy 98,262 -ex $Path/avs_library/mongo
module "file browser.user.24" -xy 258,42 -ex $Path/avs_library/mongo
module "Read128 cell.user.27" -xy 258,92
        -ex /home/sigcs/watanabe/avs/read_c128_f
port_vis "Read128 cell.user.27":"file name" -on
port_connect "orthogonal slicer.user.5":0 "field to mesh.user.6":0
port_connect "field to mesh.user.6":0 "geometry viewer.user.4":0
port_connect "volume bounds.user.8":0 "geometry viewer.user.4":0
port_connect "extract scalar.user.9":0 "orthogonal slicer.user.5":0
port_connect "generate colormap.user.11":0 "color legend.user.12":0
port_connect "generate colormap.user.11":0 "field to mesh.user.6":1
port_connect "color legend.user.12":0 "geometry viewer.user.4":0
port_connect "file browser.user.24":0 "Read128 cell.user.27":0
port_connect "Read128 cell.user.27":0 "volume bounds.user.8":0
port_connect "Read128 cell.user.27":0 "extract scalar.user.9":0
parm_set "volume bounds.user.8":"Colored Bounds" false
parm_set "extract scalar.user.9":Channel "channel 8" -range
         "channel 0;channel 1;channel 2;channel 3;channel 4;
          channel 5;channel 6;channel 7;channel 8" ;
parm_set "generate colormap.user.11":"hi value" 1
parm_set "color legend.user.12":Ticks true
parm_set "color legend.user.12":"Label Height" 0.08333333582
parm_set "color legend.user.12":"Label Gray Scale" 7
parm_set "color legend.user.12":"Label Font" 7
parm_set "color legend.user.12":"X Position" 0.6759259105
parm_set "color legend.user.12":"Y Position" 0.3518518209
parm_set "color legend.user.12":Thickness 0.04685185105
```

```
parm_set "color legend.user.12":Length 0.3877314925
parm_set "file browser.user.24":"File Browser"
          /home/sigcs/watanabe/vpp/c128test
parm_set "Read128 cell.user.27":
         "file name" /home/sigcs/watanabe/vpp/c128test
geom_set_scene -scene "geometry viewer.user.4"
geom_set_camera_name "Camera 1"
geom_resize_camera -view "Camera 1" 640 476
#
# State for view: Camera 1
#
geom_set_color -view "Camera 1" 0.908602 0.908602 0.908602
geom_set_matrix -view "Camera 1" -mat \
                  0.7    0    0    0 \
                    0    0.7    0    0 \
                    0    0    0.7    0 \
                    0    0    0    1
geom_set_position -view "Camera 1" 0 0 -8.4
geom_set_renderer "Software Renderer" -view "Camera 1"
geom_set_view_modes -depth_cue 0 -view "Camera 1"
geom_set_view_modes -polygonal_spheres 0 -view "Camera 1"
geom_set_view_modes -stereo 0 -view "Camera 1"
geom_set_view_modes -head_tracking 1 -view "Camera 1"
geom_set_view_modes -z_buffer 1 -view "Camera 1"
geom_set_camera_params -view "Camera 1" -front -88 -back 112
geom_set_depth_cue_params "Camera 1" -scale 0.1
#
# Light state
#
geom_set_light -light 1 -type directional -state 1
geom_set_light -type ambient -state 1
#
# State for object: top
#
geom_set_cur_cli_obj top
geom_set_matrix   -mat \
                  0.894775    0    0    0 \
                    0    0.894775    0    0 \
```

```
                    0     0     0.894775     0 \
                    -3.82963     -3.65068     0     1
geom_set_obj_window -0.8 9.35999 -1 9.15999 -0.08 0.08
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.3
geom_create_obj label -mod label.user.3
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.4
geom_create_obj label -mod label.user.4
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.5
geom_create_obj label -mod label.user.5
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.1
geom_create_obj label -mod label.user.1
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
```

```
geom_set_cur_cli_obj -push
geom_set_name_context label.user.2
geom_create_obj label -mod label.user.2
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: "color legend"
#
geom_set_cur_cli_obj -push
geom_set_name_context "color legend.user.12"
geom_create_obj "color legend" -mod "color legend.user.12"
geom_set_trans_mode locked
geom_set_render_mode no_light
geom_set_name_context
#
# State for object: "color boundary"
#
geom_set_cur_cli_obj -push
geom_set_name_context "color legend.user.12"
geom_create_obj "color boundary" -mod "color legend.user.12"
geom_set_select_mode parent
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: "legend labels"
#
geom_set_cur_cli_obj -push
geom_set_name_context "color legend.user.12"
geom_create_obj "legend labels" -mod "color legend.user.12"
geom_set_select_mode parent
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: "tick marks"
#
geom_set_cur_cli_obj -push
geom_set_name_context "color legend.user.12"
geom_create_obj "tick marks" -mod "color legend.user.12"
```

```
geom_set_select_mode parent
geom_set_name_context
geom_set_cur_cli_obj -pop
geom_set_cur_cli_obj -pop
#
# State for object: "field mesh"
#
geom_set_cur_cli_obj -push
geom_set_name_context "field to mesh.user.6"
geom_create_obj "field mesh" -mod "field to mesh.user.6"
geom_set_trans_mode parent
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: hedgehog
#
geom_set_cur_cli_obj -push
geom_set_name_context hedgehog.user.7
geom_create_obj hedgehog -mod hedgehog.user.7
geom_set_trans_mode redirect
geom_set_color 0.000 0.000 0.000
geom_set_properties -amb 0.300 -diff 0.700 -spec 0.000 -exp 50.000 \
 -trans 1.000 -spec_col 0.990 0.990 0.990
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: bounds
#
geom_set_cur_cli_obj -push
geom_set_name_context "volume bounds.user.8"
geom_create_obj bounds -mod "volume bounds.user.8"
geom_set_trans_mode parent
geom_set_obj_window -0.8 9.35999 -1 9.15999 -0.08 0.08
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
```

```
geom_set_cur_cli_obj -push
geom_set_name_context label.user.0
geom_create_obj label -mod label.user.0
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.14
geom_create_obj label -mod label.user.14
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.15
geom_create_obj label -mod label.user.15
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.16
geom_create_obj label -mod label.user.16
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.17
geom_create_obj label -mod label.user.17
geom_set_name_context
geom_set_cur_cli_obj -pop
#
```

```
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.18
geom_create_obj label -mod label.user.18
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.19
geom_create_obj label -mod label.user.19
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.20
geom_create_obj label -mod label.user.20
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.21
geom_create_obj label -mod label.user.21
geom_set_name_context
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.22
geom_create_obj label -mod label.user.22
geom_set_name_context
```

```
geom_set_cur_cli_obj -pop
#
# State for object: label
#
geom_set_cur_cli_obj -push
geom_set_name_context label.user.28
geom_create_obj label -mod label.user.28
geom_set_name_context
geom_set_cur_cli_obj -pop
shell "ui" shell
 panel Application -w app_panel -p ui -xy 0,0 -wh 260,984
  panel "Top Level Stack"
         -w master_stack -p Application -xy 2,100 -wh 256,731\
   -P columns integer 1
   panel "orthogonal slicer.user.5" -w panel -p "Top Level Stack" \
       -xy 0,193 -wh 224,150
    manipulator "orthogonal slicer.user.5:slice plane" -w idial -p
        "orthogonal slicer.user.5" \
        -xy 10,10 -wh 90,130
    manipulator "orthogonal slicer.user.5:axis" -w radio_buttons -p
        "orthogonal slicer.user.5" \
        -xy 100,10 -wh 118,66
   panel "field to mesh.user.6" -w panel -p "Top Level Stack" \
       -xy 0,193 -wh 138,42
    manipulator "field to mesh.user.6:Z scale" -w typein_real -p
        "field to mesh.user.6" \
        -xy 10,10 -wh 118,22
    manipulator "field to mesh.user.6:Normalize" -w toggle -p
        "field to mesh.user.6" \
        -xy 10,10 -wh 118,22
   panel "volume bounds.user.8" -w panel -p "Top Level Stack" \
       -xy 0,193 -wh 134,193
    manipulator "volume bounds.user.8:Hull" -w toggle -p
        "volume bounds.user.8" \
        -xy 10,10 -wh 118,22
    manipulator "volume bounds.user.8:Min I" -w toggle -p
        "volume bounds.user.8" \
        -xy 10,32 -wh 118,22
```

```
 manipulator "volume bounds.user.8:Max I" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,54 -wh 118,22
 manipulator "volume bounds.user.8:Min J" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,74 -wh 118,22
 manipulator "volume bounds.user.8:Max J" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,94 -wh 118,22
 manipulator "volume bounds.user.8:Min K" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,120 -wh 118,22
 manipulator "volume bounds.user.8:Max K" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,142 -wh 118,22
 manipulator "volume bounds.user.8:Colored Bounds" -w toggle -p
     "volume bounds.user.8" \
     -xy 10,164 -wh 118,22
panel "extract scalar.user.9" -w panel -p "Top Level Stack" \
    -xy 0,193 -wh 138,218
 manipulator "extract scalar.user.9:Channel" -w radio_buttons -p
     "extract scalar.user.9" \
     -xy 10,10 -wh 118,198
panel "generate colormap.user.11" -w panel -p "Top Level Stack" \
    -xy 0,193 -wh 254,460
 manipulator "generate colormap.user.11:colormap" -w color_editor -p
     "generate colormap.user.11" \
     -xy 10,10 -wh 236,416
 manipulator "generate colormap.user.11:lo value" -w typein_real -p
     "generate colormap.user.11" \
     -xy 10,428 -wh 118,22
 manipulator "generate colormap.user.11:hi value" -w typein_real -p
     "generate colormap.user.11" \
     -xy 100,428 -wh 118,22
panel "color legend.user.12" -w panel -p "Top Level Stack" \
    -xy 0,193 -wh 255,538
 manipulator "color legend.user.12:legend control" -w text -p
     "color legend.user.12" \
```

```
    -xy 10,10 -wh 236,22
manipulator "color legend.user.12:position" -w radio_buttons -p
    "color legend.user.12" \
    -xy 10,32 -wh 118,44
manipulator "color legend.user.12:Reverse Colors" -w toggle -p
    "color legend.user.12" \
    -xy 126,32 -wh 118,22
manipulator "color legend.user.12:Legend Outline" -w toggle -p
    "color legend.user.12" \
    -xy 126,54 -wh 118,22
manipulator "color legend.user.12:Outline Gray Scale" -w islider -p
    "color legend.user.12" \
    -xy 10,75 -wh 236,32
manipulator "color legend.user.12:Label Controls" -w text -p
    "color legend.user.12" \
    -xy 10,107 -wh 236,22
manipulator "color legend.user.12:Labels" -w toggle -p
    "color legend.user.12" \
    -xy 10,130 -wh 118,22
manipulator "color legend.user.12:Ticks" -w toggle -p
    "color legend.user.12" \
    -xy 126,130 -wh 118,22
manipulator "color legend.user.12:Number of Ticks" -w islider -p
    "color legend.user.12" \
    -xy 10,152 -wh 236,32
manipulator "color legend.user.12:Label Height" -w slider -p
    "color legend.user.12" \
    -xy 10,183 -wh 236,32
manipulator "color legend.user.12:Decimal Precision" -w islider -p
    "color legend.user.12" \
    -xy 10,215 -wh 236,32
manipulator "color legend.user.12:Label Gray Scale" -w islider -p
    "color legend.user.12" \
    -xy 10,257 -wh 236,32
manipulator "color legend.user.12:Label Font" -w islider -p
    "color legend.user.12" \
    -xy 10,290 -wh 236,32
manipulator "color legend.user.12:Legend Position" -w text -p
```

```
        "color legend.user.12" \
        -xy 10,322 -wh 236,22
    manipulator "color legend.user.12:X Position" -w slider -p
        "color legend.user.12" \
        -xy 10,343 -wh 236,32
    manipulator "color legend.user.12:Y Position" -w slider -p
        "color legend.user.12" \
        -xy 10,375 -wh 236,32
    manipulator "color legend.user.12:Z Position" -w slider -p
        "color legend.user.12" \
        -xy 10,408 -wh 236,32
    manipulator "color legend.user.12:Thickness" -w slider -p
        "color legend.user.12" \
        -xy 10,440 -wh 236,32
    manipulator "color legend.user.12:Length" -w slider -p
        "color legend.user.12" \
        -xy 10,472 -wh 236,32
  panel "file browser.user.24" -w panel -p "Top Level Stack" \
      -xy 0,193 -wh 256,214
    manipulator "file browser.user.24:File Browser" -w browser -p
        "file browser.user.24" \
        -xy 10,10 -wh 236,194
  panel "Read128 cell.user.27" -w panel -p "Top Level Stack" \
      -xy 0,193 -wh 256,474
    manipulator "Read128 cell.user.27:file name" -w browser -p
        "Read128 cell.user.27" \
        -xy 10,10 -wh 236,194
    manipulator "Read128 cell.user.27:base point x" -w dial -p
        "Read128 cell.user.27" \
        -xy 10,204 -wh 90,130
    manipulator "Read128 cell.user.27:base point y" -w dial -p
        "Read128 cell.user.27" \
        -xy 100,204 -wh 90,130
    manipulator "Read128 cell.user.27:base point z" -w dial -p
        "Read128 cell.user.27" \
        -xy 10,334 -wh 90,130
panel "geometry viewer.user.4!display" -w container -p ui
        -xy 1023,161 -wh 680,548\
```

```
   -P zoom_coords string "0 0 0 0 0 <$NULL> 0 0 0 0"
manipulator "geometry viewer.user.4":object -w none
manipulator "geometry viewer.user.4":"Update Always" -w none
manipulator "geometry viewer.user.4":"Update Image" -w none
# End of file
```

上記ネットワークファイルは，次章で示すモニター用シェルプログラムから自動的に読み込まれ，実行される．シェルプログラムでは，実行するネットワークと描画用データを読み込むためのモジュールを指定する必要があり，ネットワークを作成，保存後，次のパラメータの設定値を確認する．

```
parm_set ''file browser.user.24'':''File Browser''
         /home/sigcs/watanabe/vpp/c128test
```

作成済みのネットワークを変更，保存した場合， user.24 の値が変わる可能性があるため注意を要する．

3.3.2 リードモジュール

Fig.3.1で使用している描画データ読み込み用のモジュール， Read128 cell を作成する際の FORTRAN ソースプログラムを以下に示す．このソースプログラムをコンパイルし， AVS ライブラリとリンクすることにより，リードモジュールが作成される．

ここでは /home/sigcs/watanabe/avs/read_c128_f として作成される．作成したリードモジュールは，ネットワーク作成の際にモジュールとして登録することができる．モジュール作成方法の詳細は，文献 [4, 9] 参照のこと．

```
C ***************************************
C  Module Description
C ***************************************
      integer function Read_Vsc_desc()
      implicit none
      include '/usr/avs/include/avs.inc'

      integer in_port, out_port, param
      external Read_Vsc_compute
      integer Read_Vsc_compute

      call AVSset_module_name('Read128 cell', 'data')
      call AVSset_module_flags(single_arg_data)
```

```
C Output Port Specifications
      out_port = AVScreate_output_port('output field',
     $  'field 3D 3-space 3-vector irregular float')

C Parameter Specifications
      param = AVSadd_parameter('file name', 'string', ' ', ' ', '.dat')
      call AVSconnect_widget(param, 'browser')
      param = AVSadd_parameter('base point x','real', -0.800000,
     $  -1.00000, 1.00000)
      call AVSconnect_widget(param, 'dial')
      param = AVSadd_parameter('base point y','real', -1.00000,
     $  -1.00000, 1.00000)
      call AVSconnect_widget(param, 'dial')
      param = AVSadd_parameter('base point z','real', -0.0800000,
     $  -1.00000, 1.00000)
      call AVSconnect_widget(param, 'dial')

      call AVSset_compute_proc(Read_Vsc_compute)

      Read_Vsc_desc = 1
      return
      end

C ****************************************
C Module Compute Routine
C ****************************************
      integer function Read_Vsc_compute( output_field, file_name,
     $  base_point_x, base_point_y, base_point_z)
      implicit none
      include '/usr/avs/include/avs.inc'
      integer output_field
      character*(*) file_name
      real base_point_x
      real base_point_y
      real base_point_z

       integer dims0
       dimension dims0(3)
```

```
C Internal declarations
        external Read_Vsc_compute2
        integer iresult, ofield, ocoords, Read_Vsc_compute2
        real    field_data(1), coords(1)
        integer nx, ny, nz

C Field dimension size
       nx = 128
       ny = 128
       nz =  3

C Free old field data
       if (output_field .ne. 0) call AVSfield_free(output_field)

C Allocate space for new field output
       dims0(1) = nx
       dims0(2) = ny
       dims0(3) = nz
       output_field = AVSdata_alloc
     $               ('field 3D 3-space 9-vector irregular float',
     $                dims0)
       if (output_field .eq. 0) then
       call AVSerror('Allocation of output field failed.')
           Read_Vsc_compute = 0
           return
endif

       iresult = AVSfield_data_offset(output_field, field_data, ofield)
       iresult = AVSfield_points_offset(output_field, coords, ocoords)

       iresult = Read_Vsc_compute2(field_data(ofield+1),
     $           coords(ocoords+1), nx, ny, nz, file_name,
     $           base_point_x, base_point_y, base_point_z)

       Read_Vsc_compute = iresult
       return
       end
```

```
C ************************************************************************
C Initialization for modules contained in this file.
C ************************************************************************
      subroutine AVSinit_modules
      include '/usr/avs/include/avs.inc'

      external Read_Vsc_desc
      integer Read_Vsc_desc
      call AVSmodule_from_desc(Read_Vsc_desc)
      end

      integer function Read_Vsc_compute2(field_data, coords,
     $         nx, ny, nz, file_name, bpx, bpy, bpz)

      include '/usr/avs/include/avs.inc'
      integer   nx, ny, nz
      character*(*) file_name
      real      bpx, bpy, bpz
      real      field_data(9,nx,ny,nz), coords(nx,ny,nz,3)

C Internal declarations
      integer   i, j, k,nnn
      real      vx2, vy2, dl, dh, dw, xol, yol, zol, dx, dy, dz,ttt,xxx

C Vector set
       open(1, file=file_name)

        read(1,5)((nnn,field_data(1,i,j,1),field_data(2,i,j,1),
     *    field_data(4,i,j,1),field_data(5,i,j,1),field_data(6,i,j,1),
     *    field_data(8,i,j,1),field_data(9,i,j,1), j=1,nx),i=1,ny)
    5  format(1h ,i5,7f12.4)

      do  10  i = 1, nx
       do  10  j = 1, ny
         field_data(3,i,j,1) = 0.0
         field_data(7,i,j,1) = 0.0
 10   continue
```

```
      close(1)

C Coordinate set
      dl = 0.08
      dh = 0.08
      dw = 0.08

      xol = bpx
      yol = bpy
      zol = bpz

      dx = xol
      dy = yol
      dz = zol

      do 120 i = 1, nx
         do 110 j = 1, ny
            do 100 k = 1, nz
               coords(i,j,k,1) = dx
               coords(i,j,k,2) = dy
               coords(i,j,k,3) = dz
               dz = dz + dw
100         continue
            dy = dy + dh
            dz = zol
110      continue
         dx = dx + dl
         dy = yol
120   continue

      Read_Vsc_compute2 = 1

      return
      end
```

3.3.3 2次元密度分布図

Fig.3.1に示されている2次元密度分布描画用ネットワークを使用した描画例をFig.3.2に示す. これは，次章に示すシェルプログラムを使用することにより，連続的にVPP500でシミュレーションジョブを実行し，可視化用出力ファイルを自動的にワークステーションに転送し，描画を行なった際の表示画面の一例である．ここでは，赤粒子の割合を赤(1.0)から青(0.0)の色の変化で示している．シミュレーションは 1024×1024 の格子上で行ない，128×128 のサンプリングセルで粗視化したものである.

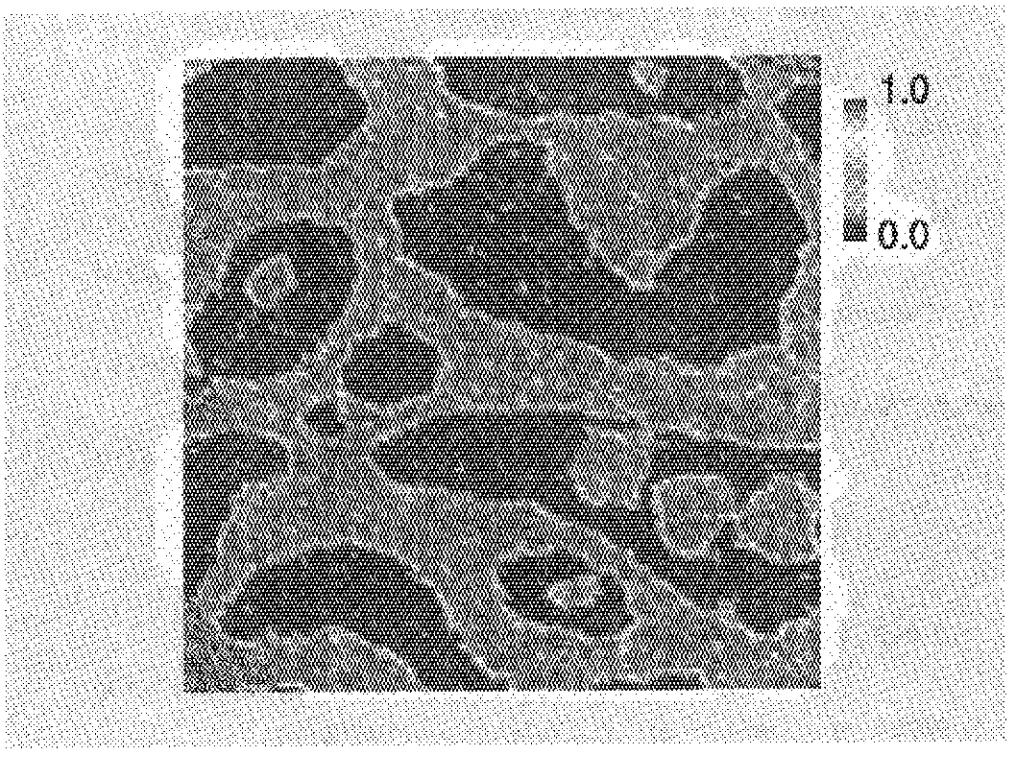

Fig. 3.2 2-D density distribution obtained by cellular automata method

Fig.3.2の描画に用いたデータファイルの一部を以下に示す．1画面，128 x 128 = 16384 サンプリングセルのデータとして，16384 行のデータからなっており，データの並びは，セルの通し番号，赤粒子に対する 2 次元の速度成分と密度，青粒子に対する 2 次元の速度成分と密度，セルの総粒子数に対する赤粒子の割合（密度比）である．この描画例では，最後の密度比のみを描画に用いているが，速度分布や密度そのものの描画にもこのデータファイルが使用できるように，このような並びで転送している．転送以前に画像ファイルに加工したり，圧縮操作を施してはいない．これは，スーパーコンピュータ上のシミュレーション用プログラムにはなるべく変更を加えることなくジョブを実行するためと，シミュレーション結果の生データをワークステーションに転送することにより，データの加工，保存，可視化等を，ワークステーション側で自由に行なうためである．また，目視でデータを確認するため，ここでは ASCII 形式のまま転送している．このため，一画面でおよそ 1.5MB の容量となっている．

```
    1  0.6818E-01 -0.3936E-01  0.8468E+01  0.0000E+00
                               0.0000E+00  0.0000E+00  0.1000E+01
    2  0.9524E-01 -0.8248E-01  0.8083E+01  0.0000E+00
                               0.0000E+00  0.0000E+00  0.1000E+01
    3  0.1087E+00 -0.3765E-01  0.8853E+01  0.0000E+00
                               0.0000E+00  0.0000E+00  0.1000E+01
    4 -0.6522E-01 -0.1130E+00  0.8853E+01  0.0000E+00
                               0.0000E+00  0.0000E+00  0.1000E+01

                               ～

16381  0.2381E-01  0.1237E+00  0.8083E+01 -0.1000E+00
                              -0.5196E+00  0.1925E+01  0.8077E+00
16382  0.2632E-01 -0.4558E-01  0.7313E+01 -0.1667E+00
                              -0.2887E+00  0.1155E+01  0.8636E+00
16383  0.2000E+00 -0.1155E+00  0.5774E+01 -0.2500E+00
                              -0.4330E+00  0.1540E+01  0.7895E+00
16384  0.5882E-01  0.0000E+00  0.6543E+01 -0.3333E+00
                              -0.5774E+00  0.1155E+01  0.8500E+00
```

4. シェルプログラム

4.1 可視化の制御

可視化を行なうワークステーション上で可視化制御用シェルプログラムを実行する．このシェルプログラムはAVSサーバの起動，AVSクライアントの起動，描画データの検索を行なう．このシェルプログラムは任意のUNIXマシン上で実行可能である．以下に可視化制御用シェルプログラム（monitor）とその重要なコマンドについて説明する．

```
001  #!/bin/csh
002  #
003  # AVS server & client  shell (monitor)
004  #
005  avs  -server &
006  sleep 5
007  set GG=`ls -1 -t /tmp/avs_server.* | sed -n '1,1p' |
     sed 's/\/tmp\/avs_server.//'`
008  echo $GG
009
010  avs_client $GG << eod
011  net_clear
012  net_read /home/sigcs/watanabe/pepper/cellflow.net
013  eod
014  sleep 5
015
016  while (1)
017  sleep 5
018
019  cd /home/sigcs/watanabe/vpp
020
021  set FL=`ls -1| grep c128test`
022
023  if ( $FL == 'c128test') then
024      sleep 10
025      date
026
027  avs_client $GG << eod
028  parm_set "file browser.user.24":
```

```
    "File Browser" /home/sigcs/watanabe/vpp/c128test
029 eod
030
031 sleep 10
032
033 /bin/rm -r /home/sigcs/watanabe/vpp/c128test
034 endif
035 continue
036 end
```

005 : AVS をサーバーオプションを付けて実行．コンソール以外の X 端末やワークステーションから使用する場合には，-nohw を追加する．

006 : AVS が確実に立ち上がるまで 5 秒間（機種による）スリープ．

007 : AVS サーバのプロセス ID を /tmp にできた avs_server.xxxx（xxxx はプロセス ID）から取得．

008 : 取得したプロセス ID を表示．

010-013 : avs_client xxxx（xxxx はプロセス ID）を実行し，AVS サーバに AVS ネットワークを読み込ませる．ネットワークは，/home/sigcs/watanabe/pepper のディレクトリ上の cellflow.net である．

014 : ネットワークが確実に読み込まれるまで 5 秒間（機種による）スリープ．

016 : 035 continue まで無限に実行する．終了するにはシェルプログラムを中断し，AVS サーバを exit する．

017 : 5 秒に一回ファイルを検索させるためスリープする．このスリープを入れないと，常に検索を行ないワークステーションの負荷が高くなってしまう．

019 : データが転送されてくるディレクトリ（/home/sigcs/watanabe/vpp）に移動する．

021 : ls -1| grep c128test の結果を（c128test は転送されてくる描画データ）FL にセットする．描画データが存在していれば，FL には c128test が入っている．

023-025 : $FL='c128test' ならば，１０秒（データ転送中であったときに転送終了まで待つ，ネットワークの状況による）スリープしてデータを見つけた日時を出力．データがなかった場合は，while (1) ループの先頭に戻る．

027-029 : avs_client xxxx（xxxx はプロセス ID）を実行し，AVS サーバに描画データ（c128test）を読み込ませる．（描画する）

031 : AVS サーバのネットワークが c128test を確実に読み込むまで１０秒（機種による）スリープする．

033 : c128test を削除する．

034 : if 文の終り．

035 : while 文の終り．

036 : シェルプログラムの終り．

4.2 シミュレーションジョブの実行

ここでは，シミュレーションを VPP500 で行なうことを想定する．一つのジョブは，前回のリスタートファイルを読み込み計算を開始し，中間データ（描画用データ）を一つ出力し，リスタートファイルを書き出して終了するものとする．一つのジョブが正常終了すると次のジョブが投入され，過渡変化を計算し続ける．これは，ある初期値から出発して，定常状態まで計算する時によく使用するシェルの例である．シミュレーションジョブ実行用のシェルは，シミュレーションにより作成された中間データ（描画用データ）を描画を行なうワークステーションに転送する部分（gc001）と，通常，ジョブをサブミットする時にコマンドラインから入力する qsub 命令の部分（gc1），ジョブのサブミットに使用する部分（gc1.sh）からなる．以下にシェルとその重要部分の説明を記す．これらのシェルは，この例では $HOME/cellx 上にあり，このディレクトリで実行している．

4.2.1 描画用データ転送用シェル：gc001

```
##################################################
#   gc001   submit & ftp shell
##################################################

001  gc1
002
003  until grep logout gc1.sh.o*
004
005  do
006  date
007  echo '** logout is not found in gc1 : sleep 30 **'
```

```
008  sleep  30
009  done
010  echo '** logout was found in gc1 **'
011
012  sleep 5
013
014  cat gc1.sh.o* >> gcshout
015
016  /bin/rm gc1.sh.*
017
018  ftp -n 133.53.10.152 << eod
019  user watanabe pasword
020  cd /home/sigcs/watanabe/vpp
021  prompt
022  bin
023  put /dg04/ufs02/j4042/wkvfl/cellx/flow.dat c128test
024  eod
025
026  nohup gc001 &
```

001 : ジョブをサブミットするコマンドが記述されたシェルプログラム gc1 を実行する. (ジョブがサブミットされる)

003 : 標準出力ファイル (gc1.sh.oXXXX) の中に logout の文字列が見つかるまで, 005 do ～ 009 done までを繰り返す.

006 : 日時を出力する.

007 : logout が見つからなかったことを表示.

008 : 計算機に負荷をかけないように３０秒間スリープする.

010 : logout が見つかったことを表示.

012 : 描画データが確実に出力されるまで5秒間スリープする.

014 : 標準出力ファイルを gcshout にセーブする.

016 : 標準出力ファイル, エラーファイルを削除.

018-024 : 可視化用ワークステーションに対して FTP を実行し, 描画データを転送する. ここでは, 可視化用ワークステーションの IP アドレスが 133.53.10.152, ユーザー

名 watanabe である．転送先のディレクトリが /home/sigcs/watanabe/vpp であり，転送するファイルは，VPP500 の /dg04/ufs02/j4042/wkvfl/cellx/flow.dat，転送先でのファイル名が c128test である．

026 : 自分自身を再度実行する．nohup はこのコマンドがセッション終了 (logout) 後も有効であることを示し，& はこのコマンドがバックグラウンドで機能することを表している [11].

4.2.2 ジョブサブミットコマンドシェル：gc1

```
##########################################################
#   gc1    shell of job submit
##########################################################

027  qsub -q vpps -lm 96mb -lt 0:30:00  gc1.sh
028  echo '** gc1 end **'
```

027 : ジョブをサブミットする．（vpps クラス，メモリ９６ＭＢ，ＣＰＵ時間３０分）

028 : このシェルプログラムが終ったことを表示.

4.2.3 ジョブサブミットシェル：gc1.sh

```
##########################################################
#   gc1.sh    true submit
##########################################################

031  #!/bin/csh -f
032
033  cd $HOME/wkvfl/cellx
034
035  /bin/cp restout.dat restin.dat
036  vfl_flush restin.dat
037
038  cd $HOME/cellx
```

```
039
040  setenv fu10 ~/wkvfl/cellx/restin.dat
041  setenv fu11 ~/wkvfl/cellx/debuggc1.dat
042  setenv fu20 ~/wkvfl/cellx/restout.dat
043  setenv fu31 ~/wkvfl/cellx/flow.dat
044
045  date
046
047  cell1
```

031 : Cシェルを高速モードで起動.

033 : $HOME/wkvfl/cellx に移動.

035 : 前回のジョブで出力した再スタート用データ（restout.dat）を restin.dat にコピーし，再スタートデータとする.

036 : VPP500 の SSU （システム記憶装置）上に残っている古い restin.dat ファイルを削除.

038 : $HOME/cellx に移動.

040-043 : 各機番にファイルを割り当てる.

045 : 日時を出力.

047 : cell1 を実行.

5. 実時間可視化とビデオ録画

VPP500と可視化用ワークステーションでそれぞれのシェルプログラムを実行することにより実時間での描画を実現できる．また，AVSネットワークにビデオ制御用モジュールを追加すると，描画結果をビデオ録画し，アニメーション化することも可能となる．Fig.3.1に示されるネットワーク例では，AVSによる描画（geometry viewerモジュール）の後にビデオ制御用モジュール（image to nvs, rec nvs）が追加してある．ビデオ制御機器，フレームスキャンコンバーター等が，Fig.2.1に示される本システムの機器構成と異なる場合は，ビデオ制御用モジュールも異なったものとなる．以下では，シェルプログラム及びAVSによる実時間描画，ビデオ録画の手順について述べる．

5.1 実時間可視化

実時間での描画を行なうには，VPPから描画データが転送されてくる可視化用ワークステーションのディレクトリで，可視化用シェルプログラム（monitor）を実行する．その後VPP上でジョブをサブミットするシェルプログラム（gc001）を実行する．

ここで示した例では，VPPのシミュレーションとして一つのジョブで描画ファイルを一つ生成しているため，ジョブ終了時にファイルを転送しているが，一つのジョブの中でいくつもの描画ファイルを生成し，逐次転送することも可能である．その場合は，シミュレーションプログラムの中で，描画用データを書き出した後，そのファイルをクローズする必要がある．これにより，ジョブ実行中でもファイルにアクセスすることが可能となり，計算が進んだ段階までのデータを転送することができ，実時間可視化を行なうことができる．

ファイル転送の速度はファイルの大きさとネットワークの状況に依存するため，これを考慮してシェル内のsleep時間を決める必要がある．また，あまり大量のデータを連続して転送するのは，ネットワークの負荷を高めるので，シミュレーションのうち描画に必要なデータ量と過渡変化を可視化する頻度の選択には注意を要する．

なお，那珂研のParagonを使用する場合でも，同様のシェルを組むことができる．これは，いずれのシステムも，フロントエンドとしてのファイルサーバーを持ち，ジョブをバッチジョブとして扱うためである．シミュレーションを行なう計算機が，ワークステーションなどの通常のUNIXマシンの場合，あるいは，可視化を行なうワークステーションと同一の場合，これらのシェルは，より簡単なものとなる．

5.2 ビデオ録画

ビデオ録画をリアルタイムモニター機能に付加する場合は，ビデオ制御用モジュールのパラメータを設定する必要がある．あらかじめタイムコードを記録したテープで，以下の手順で録画開始の位置を指定し，録画状態にしてネットワークをセーブしておく．

1. AVSを起動し，ネットワークを開く．

2. rec nvs モジュールのパラメータを設定する.

Clear Frame Buffer	: NVS 画面をクリアする.
Put a Color Bar	: カラーバーを表示する.
Tape Initialize	: テープを初期化する.
Tape Positioning	: テープ位置を Rec Position にする.
NVShost	: NVS2000 の IP アドレスを設定.
Rec Position	: 録画を開始するタイムコードを設定.
Current Rec. Time	: 現在の画面の録画時間を設定.
Each Frame Unit	: コマ撮りの画面の録画時間の設定.
View Only	: 表示のみで録画しない.
Rec. Current	: 現在の画面を Current Rec. Time 分録画する.
Rec. Standby	: コマ撮りモードで待機する.

3. image_to_nvs モジュールのパラメータを設定する.

NVShost	: NVS2000 の IP アドレスを設定.

4. ネットワークをセーブする.

5. 実時間描画を実行する.

本システムでは, AVS が起動されメニュー画面が現れた段階で, もういちど rec nvs の Rec. Standby のパラメータをオンにする必要がある. もちろん, 本システムの機器構成と異なるビデオ装置を使用する場合は, モジュールパラメーター等も違ったものとなる [6, 7].

6. おわりに

可視化技術は，シミュレーションを行なっている現場では，研究者個人やグループが計算結果を効率良く理解し，新たな計算やデータ処理をすみやかに行なうために利用される．このため，可視化，画像処理に関しては，ビジュアリゼーション技法や圧縮技術の研究ばかりではなく，ファイル操作やデータ処理を含め，利用技術の開発・研究が重要となる．可視化用ソフトウェアや可視化技術開発が細分化，専門化し，可視化ツールや機器のグラフィックス機能が豊富になり，また，次々に新しいソフトウェアが開発されるにともない，使用方法が簡単かつ不変的であることが利用上重要なこととなってきている．シミュレーション研究のかたわら可視化や画像処理技術の開発を行なうのは困難であり，また新しいソフトウェアや機器を導入する度に使用方法を習得するというのは効率的ではないためである．

本報告書では，実時間でシミュレーション結果を可視化用ワークステーション上で連続的に描画し，それを自動的にビデオ録画するリアルタイムモニターシステムの開発と使用方法について解説した．実時間可視化，あるいはスーパーコンピュータのリアルタイムモニターといった概念や手法は多数考えられるものであるが，本システムは，汎用可視化ツールであるグラフィックスソフトウェアAVSと通常のUNIXコマンドのみで構成されている．このため，本システムを異なる計算機環境へ応用，拡張することは容易であり，シミュレーションを研究手段として用いている計算機ユーザーは容易にリアルタイムモニター機能を使用することができる．また，AVS以外の可視化ツールを使用する場合でも，ジョブのサブミットとファイル転送に関するUNIXシェルは有効である．可視化用ワークステーションが手元に無い場合でも，計算科学技術推進センターで管理しているグラフィックサーバ用ワークステーションをネットワークから利用することにより[4, 6]，リアルタイムモニター機能をユーザーのシミュレーションに付加することが可能となる．

スーパーコンピュータにより詳細かつ大規模な計算が可能となり，計算結果が膨大なものになると，全ての数値結果を詳細に検討することは到底不可能である．膨大な数値の中から何をどのように表示すると何が理解できるかが，シミュレーション研究の現場における可視化の課題であるが，効率的で扱いの容易な可視化表示方法を含む大規模数値シミュレーション技術の開発・整備により，シミュレーション結果の正確かつ効率的な理解が可能となる．本報告書が効率的なシミュレーション研究の一助となることを期待する．

参考文献

[1] T. Atobe, H. Kaburaki and T. Watanabe, "Direct Numerical Simulation on Small-scale Vortex Dynamics for the Forced Homogeneous Isotropic Turbulence," 49th APS Division of Fluid Dynamics Annual Meeting, Syracuse, NY, 24-26 November(1996).

[2] 跡部 隆，上野 潤一郎，蕪木 英雄，渡辺 正，一様等方性を持つ強制乱流の渦構造，九州大学大学院総合理工学研究科報告，第18巻，第1号45-49（平成8年）

[3] T. Watanabe and H. Kaburaki, "Particle Simulation of Convection Pattern in 3-D Rayleigh-Bénard System", 8th Joint EPS-APS International Conf. Physics Computing, Kraków, Poland, 17-21 September(1996).

[4] 加藤 克海，渡辺 正，蕪木 英雄，可視化ツールＡＶＳの導入整備，JAERI-memo 07-058(1995).

[5] 増子 献児，五来 一夫，加藤 克海，山崎 和彦，ＡＶＳ (Application Visualization System) 利用手引書，JAERI-Tech 95-008(1996).

[6] 加藤 克海，渡辺 正，町田昌彦，蕪木 英雄，動画像処理システムの開発，JAERI-Tech 95-014(1995).

[7] 渡辺 正，計算科学研究における可視化技術の応用，第9回「原子力におけるソフトウェア開発」研究会報告集，JAERI-Conf 96-002(1996).

[8] クボタコンピュータ株式会社，AVS USER'S GUIDE, 1993.

[9] クボタコンピュータ株式会社，AVS MODULE REFERENCE, 1993.

[10] 株式会社ケイ・ジー・ティー，ＡＶＳハンドブックＶｏｌ３，（平成7年）

[11] Brian W. Kernighan, Rob Pike, UNIX プログラミング環境，アスキー出版局（1991）.

[12] 流体力学研究所，ネットワークビデオサーバＮＶＳ２０００ 取扱説明書，（平成8年）.

[13] 流体力学研究所，変換ユーティリティ説明書，（平成8年）.

[14] 流体力学研究所，ＡＶＳ ＮＶＳ Ｃｏｎｔｒｏｌ説明書，（平成8年）.