

統計数理研究所御中

# 計算途中可視化ツール rviz マニュアル

第 1.0 版 2006 年 6 月 2 日

日本 SGI 株式会社

## 改訂履歴

版	日付	記述
1.0	2006年6月2日	第1版

## 計算途中可視化ツール rviz マニュアル

## — 目次 —

1.	はじめに.....	3
2.	ファイル構成.....	4
3.	動作環境.....	5
3.1	ハードウェア環境.....	5
3.2	ソフトウェア環境.....	5
4.	計算途中可視化の仕組み.....	6
5.	使用方法.....	8
5.1	起動.....	8
5.2	終了.....	8
5.3	監視設定 (Setting タブ).....	8
5.4	可視化処理設定 (Visualization タブ).....	10
5.5	動作監視 (Monitoring タブ).....	11
5.6	メニュー.....	12
6.	スクリプト作成方法.....	13
6.1	スクリプト作成方法概要.....	13
6.2	AVS 用サンプルスクリプト.....	14
6.3	gnuplot 用サンプルスクリプト.....	19

## 1. はじめに

本書は、計算途中可視化ツール rviz の使用方法を記述したものです。rviz は計算機上での長時間にわたる計算の途中結果を可視化し、最終結果の予測や、初期値の誤り等による計算ミスの早期発見を可能にするものです。本ツールを利用することで、計算途中結果のデータ可視化をさまざまな可視化ソフトウェアを用いて、同一のインターフェースで制御することが可能になります。

## 2. ファイル構成

本ツールのファイル構成について記述します。

### 【起動コマンド】

/usr/local/bin/rviz : rviz 起動コマンド(起動スクリプト)

### 【関連ファイル】

/usr/local/bin/.rviz/ : rviz 関連ファイル(Java プログラム)

.rviz.properties : rviz 設定ファイル(実行時に実行ディレクトリ下に作成)

### 【スクリプト・データサンプル】

/usr/local/bin/.rviz/rviz\_sample/\*.v : AVS 用サンプルスクリプト

/usr/local/bin/.rviz/rviz\_sample/\*.gp : gnuplot 用サンプルスクリプト

/usr/local/bin/.rviz/rviz\_sample/input : サンプル入力データ(AVS 用)

/usr/local/bin/.rviz/rviz\_sample/input\_gp : サンプル入力データ(gnuplot 用)

※ サンプルスクリプトを使用することで、3 次元の棒グラフ、スキャッター、等高線、および 2 次元の折れ線グラフ、棒グラフが簡単に作成できます。詳細は最終章をご覧ください。

## 3. 動作環境

本ツールが稼動する想定環境(ハードウェア、ソフトウェア)は以下のとおりです。

### 3.1 ハードウェア環境

Prism(ismprsm.ism.ac.jp) ※

※ rviz は Java ベースで開発されているため、後述の可視化ソフトウェアが対応していれば Windows 等での実行も可能です。/usr/local/bin/.rviz の下に Windows 用の起動スクリプト サンプル rviz\_win\_sample.bat があります。

### 3.2 ソフトウェア環境

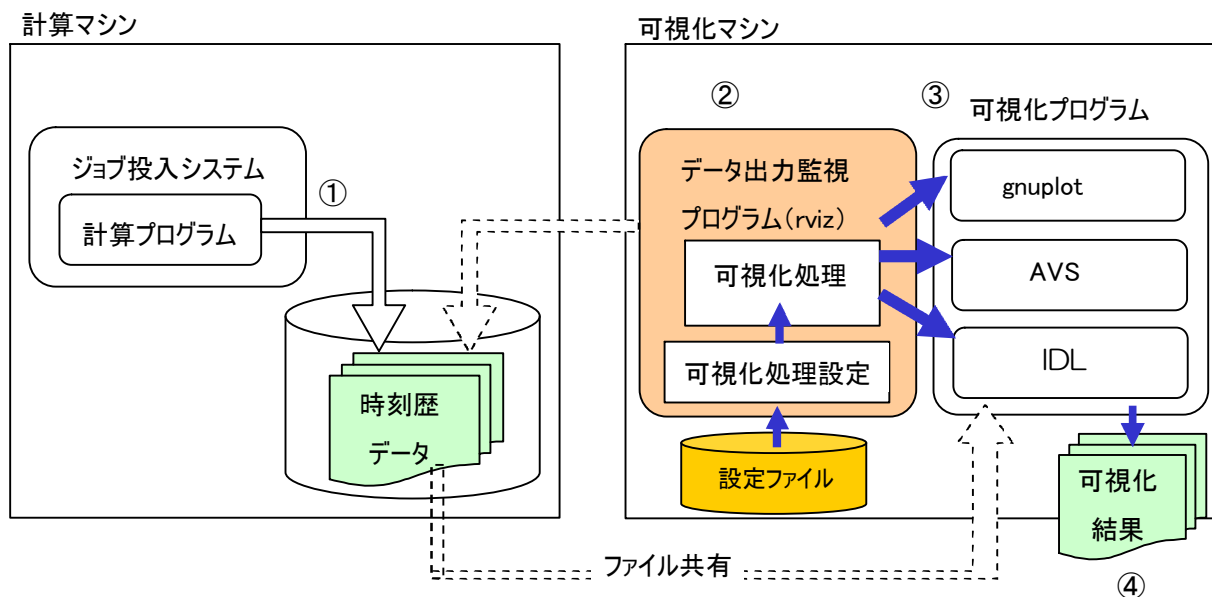
Java プラグイン(バージョン 1.4 以上)

可視化ソフトウェア ※

※AVS, gnuplot 等, スクリプトベースで実行できるソフトウェアを指します。

## 4. 計算途中可視化の仕組み

本ツールにおける計算途中可視化の仕組みについて説明します。



- ① 途中結果を監視したい計算プログラムから、途中結果を表す数値を逐次ファイル出力します。出力ディレクトリは可視化マシンから参照できる必要があります。各スーパーコンピュータと可視化サーバ (Prism) は下記の通りファイル共有されていますので、下記のスーパーコンピュータ側のディレクトリに途中結果を表す数値を逐次ファイル出力してください。

スーパーコンピュータ側ディレクトリ		Prism 側ディレクトリ
Altix システム (ismaltx)	/home0	/altx_home0
	/home1	/altx_home1
	/work	/altx_work
	/short	/altx_short
XC システム (ismxc)	/home	/home
	/work	/work

- ② 可視化サーバ(ismprsm)上に rviz と可視化プログラム、①の出力データフォーマットに対応した可視化プログラム用スクリプトを用意します。可視化サーバ(ismprsm)には rviz と、可視化プログラムとして AVS と gnuplot、およびそれぞれの可視化プログラム用のサンプルスクリプトがインストールされています (詳細後述)。

- ③ rviz を起動し、使用したい可視化プログラムとスクリプトを選択します。また監視先ディレクトリ(①の出力先ディレクトリ)や画像出力先ディレクトリ、監視インターバル、監視・出力ファイル名のフォーマット等を設定します。
- ④ rviz で監視を開始すると、監視ディレクトリ内にある計算プログラムの出力ファイルを入力として、可視化プログラムを起動します。可視化プログラムの出力画像は出力先ディレクトリに保存されます。監視は指定インターバルごとに定期的に行われ、前回可視化したものより新しいファイルを可視化対象として画像を生成します。



## 5. 使用方法

本ツールの使用方法について説明します。

rvizは3つのタブからなるGUIを提供します。それぞれ入力・出力設定, 可視化処理設定, 動作監視を行うことができます。またメニューからは設定ファイルの読み込み・保存等を行えます。

### 5.1 起動

起動コマンドをプロンプトから入力してください。

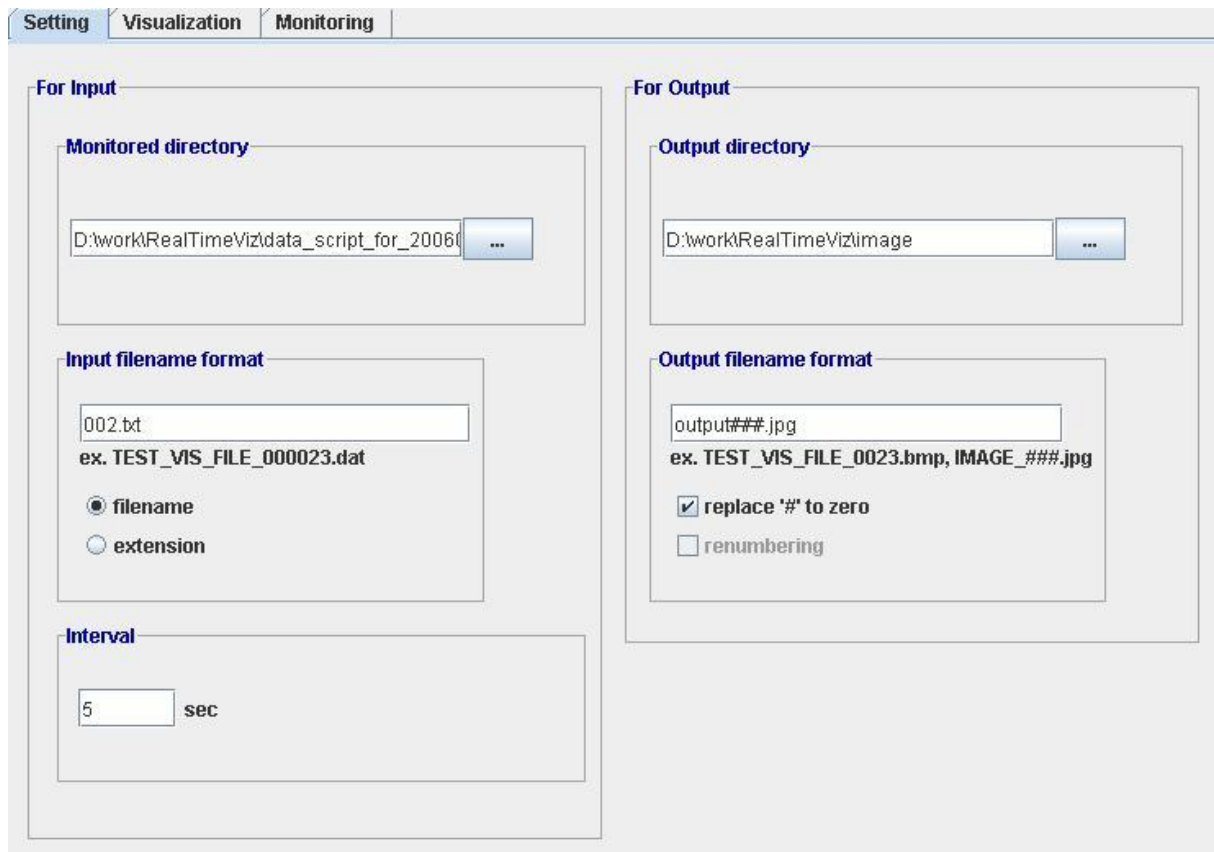
```
$ /usr/local/bin/rviz
```

### 5.2 終了

ウィンドウ右上の×ボタンを押すか、ファイルメニューのQuitを選択してください。

### 5.3 監視設定 (Setting タブ)

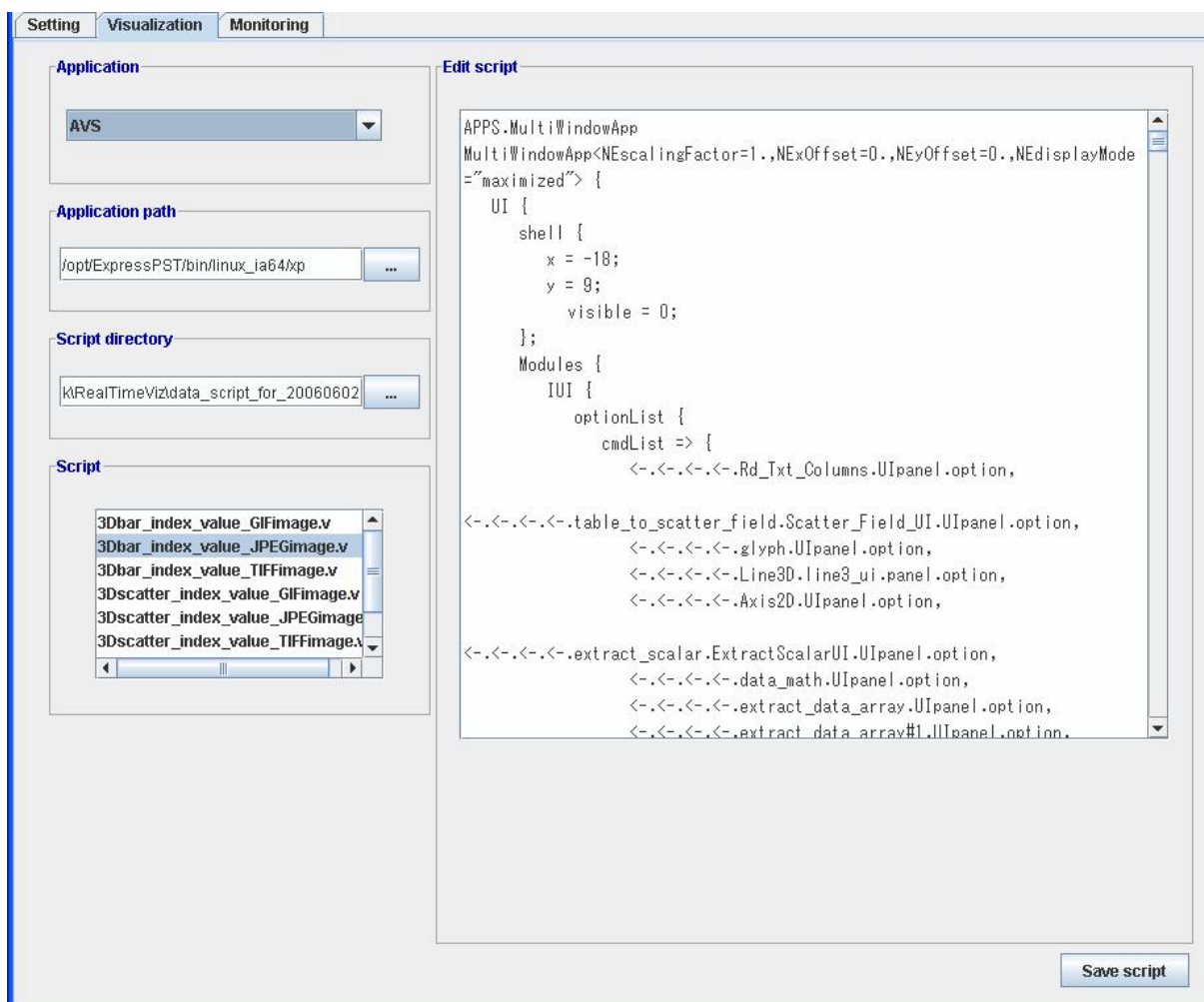
Settingタブでは入出力設定を行います。



- |                        |  |
|------------------------|--|
| Monitored directory    | 監視先ディレクトリ(計算プログラムの出力先ディレクトリ)を指定します。  |
| Input filename format  | 可視化したいファイル名の書式を指定します。<br>filename にチェックを入れた場合は指定した名前のファイルのみ監視します。特定のファイルにデータが追記される場合に利用できます。<br>extension にチェックを入れると、入力文字列を拡張子を持つファイルを監視します。   |
| Interval               | 監視インターバルです。5 であれば 5 秒ごとに監視・可視化処理を行います。   |
| Output directory       | 可視化結果の画像の出力先ディレクトリを指定します。  |
| Output filename format | 出力画像ファイル名のフォーマットを指定します。<br>このフォーマットに含まれる一連のシャープは連番に変換され、画像を出力するごとに 0 から順に番号が振られます。<br>replace “#” to zero にチェックを入れるとシャープの個数が連番の桁数として扱われ、現在の番号の桁数よりシャープの個数が多い場合、シャープの左から順に 0 詰めされます。<br>renumbering は監視を一度停止すると操作可能になります。これにチェックを入れると、監視を再開したときに連番を 0 から振り直します。 |

## 5.4 可視化処理設定 (Visualization タブ)

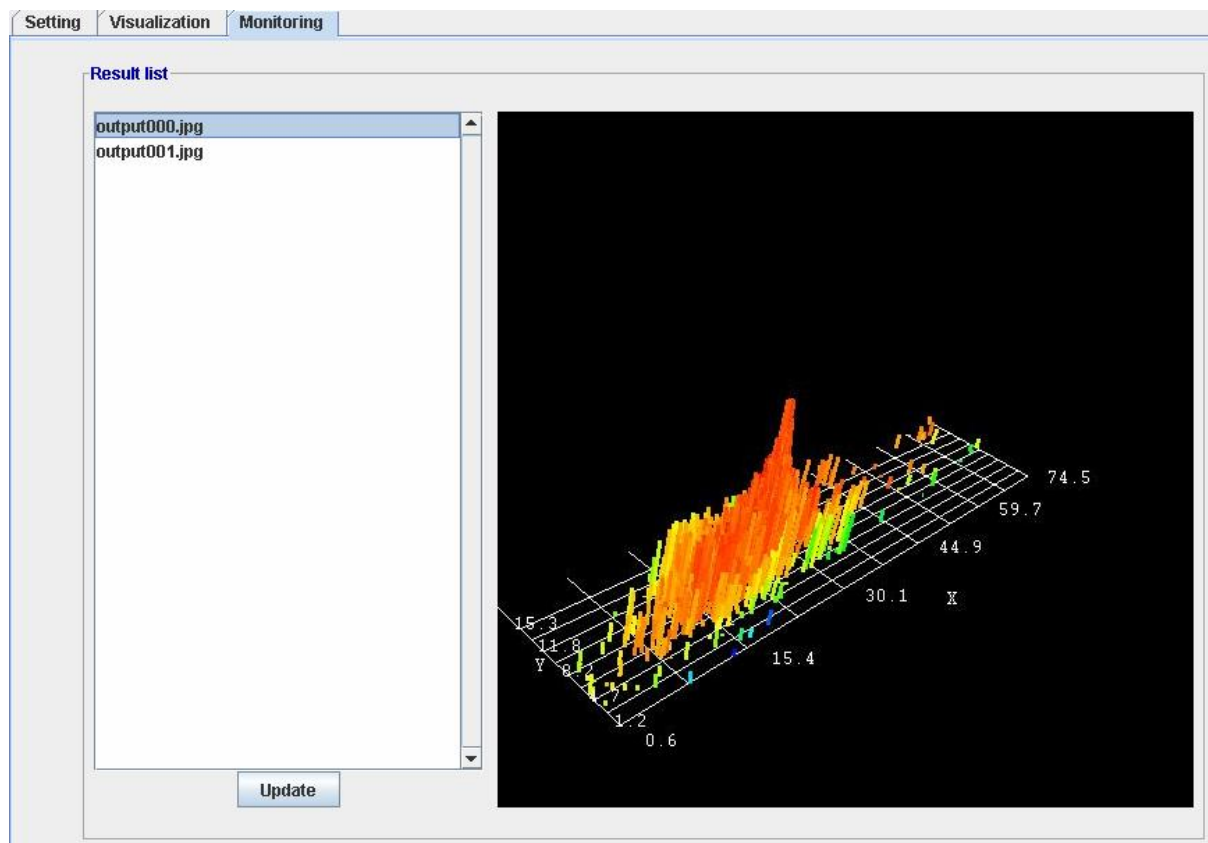
Visualization タブでは可視化処理に関する設定を行います。



Application	使用したいアプリケーションを選択します。初期設定では AVS と gnuplot が選択できます。アプリケーションを増やすには、設定ファイルをエディタで開き、application の行にカンマ区切りでアプリケーション名を追加して rviz を再起動してください。
Application path	アプリケーションのパスを指定します。
Script directory	可視化用スクリプトが置かれたディレクトリを指定します。
Script	Script directory の下にあるファイルの一覧が表示されます。この中から使用したいスクリプトを選択してください。
Edit script	Script で選択したスクリプトの内容が表示されます。ここでスクリプトの編集もできます。Save script ボタンで編集したスクリプトを保存することができます。編集内容はファイル保存後に有効になります。

## 5.5 動作監視 (Monitoring タブ)

Monitoringタブでは可視化結果の画像を表示することができます。



Result list

Setting タブで指定した出力先ディレクトリ内のファイル一覧が表示されます。監視中は順次画像ファイルが増えていくため、Update ボタンを押すことで一覧を最新の状態にできます。

※現バージョンの対応画像形式は BMP, JPEG, GIF, PNG です。

## 5.6 メニュー



Start monitoring

監視を開始します。

Stop monitoring

監視を停止します。可視化アプリケーションの起動・実行が長くなる場合、すぐには停止できないことがあります。

### 【File メニュー】

Save setting

タブで設定した内容を設定ファイルに保存します。

rviz 終了時には自動的に.rviz.properties に設定が保存されますが、このメニューでは別名で設定を保存できます。

Load setting

設定ファイルから設定を読み込みます。

rviz 起動時には自動的に.rviz.properties から設定が読み込まれますが、このメニューでは任意の設定ファイルから設定を読み込むことができます。

Quit

rviz を終了します。

### 【Help メニュー】

Manual

本ファイルを表示します。

Version

バージョン情報を表示します。

## 6. スクリプト作成方法

rviz で使用する可視化用スクリプトの作成方法について、AVS, gnuplot を例に説明します。

### 6.1 スクリプト作成方法概要

一般的に、可視化ソフトウェアの入力となるスクリプトには入力・出力データファイル名、作成したい画像の種類や座標軸名等の修飾情報、画像ファイルフォーマット等が記述されます。rviz では毎時名前の異なる複数のデータファイルを入力とし、連番画像ファイルを出力として想定しているため、スクリプトの入力・出力ファイル名の部分をパラメータ化する必要があります。具体的には、入力ファイル名を, 出力ファイル名をoutput-file に置き換えます。

例:

```
input: "/home/user1/data/input.txt"
output: "/home/user1/data/output.jpg"
      ↓
input: "$input-file"
output: "$output-file"
```

例えば AVS の場合、スクリプトは以下の手順で作成できます。

- ① AVS/Express でアプリケーションを作成します。アプリケーション作成方法については AVS のマニュアルをご覧ください。
- ② AVS のファイルメニューから「アプリケーションの保存」を選ぶと、AVS 用スクリプト(v ファイル)を出力できます。
- ③ 入力ファイル名、出力ファイル名を上記の例のように置き換えます。
- ④ スクリプトファイル末尾の行にquit を追記します。これにより監視中、画像を生成するごとに AVS が終了します。AVS は毎回終了しないと次の可視化処理が行われません。

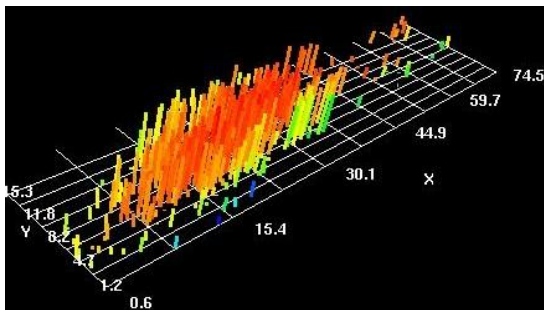
## 6.2 AVS 用サンプルスクリプト

ismprsm には rviz で使用できる AVS 用のサンプルスクリプトが 4 種類インストールされています(ファイルの置き場所については 2 章をご覧ください)。

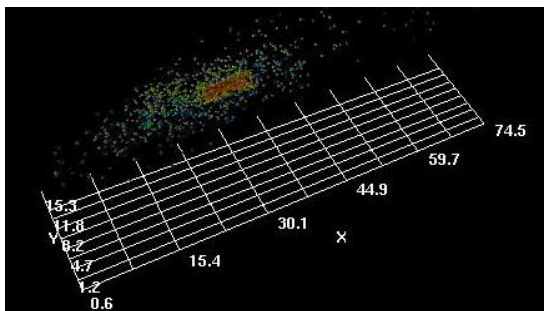
それぞれのスクリプトの用途は次の通りです。

3Dbar_index_value_GIF.image.v	: 3次元棒グラフを GIF 形式で出力します。
3Dbar_index_value_JPEG.image.v	: 3次元棒グラフを JPEG 形式で出力します。
3Dscatter_index_value_GIFimage.v	: 3次元スカッターを GIF 形式で出力します。
3Dscatter_index_value_JPEGimage.v	: 3次元スカッターを JPEG 形式で出力します。
3Disoline_index_value_GIFimage.v	: 3次元等高線を GIF 形式で出力します。
3Disoline_index_value_JPEGimage.v	: 3次元等高線を JPEG 形式で出力します。

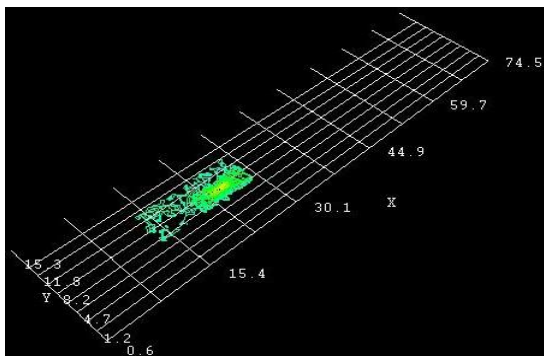
以下は出力画像サンプルです。



3次元棒グラフ



3次元スカッター



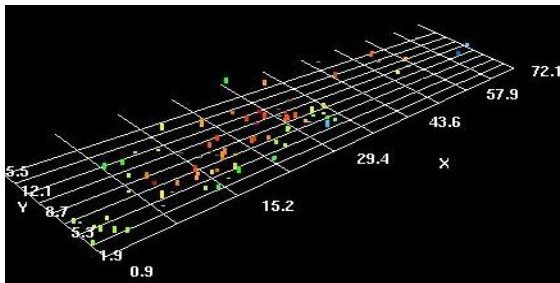
3次元等高線

これらのスクリプトは入力として空白区切りのデータを受け取ります。以下の例のように、各行がある時点での値の集合を表します。

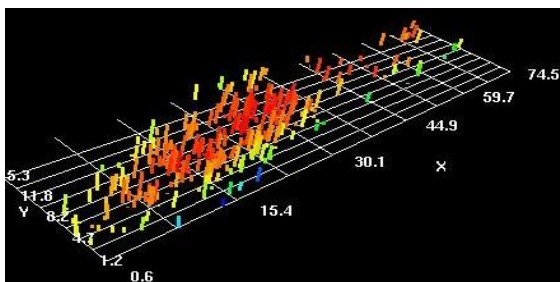
```
x1 y1 z1 w1 ...
x2 y2 z2 w2 ...
x3 y3 z3 w3 ...
...
```

サンプルの入力データファイルとして、000.txt~019.txt の 20 個のファイルをご用意しています(ファイルの置き場所については 2 章をご覧ください)。これらは時間の経過とともに、計算プログラムの途中結果を累積出力したものをイメージしています。

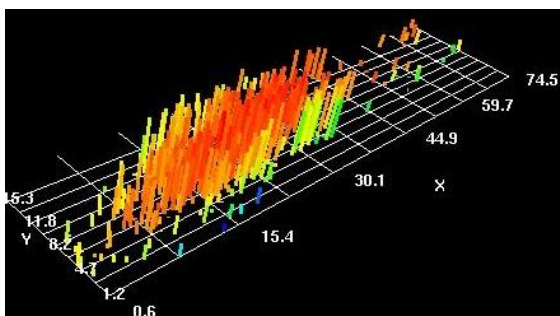
rviz を初期状態で起動(設定ファイル.rviz.properties を削除して起動)すると、上記の AVS サンプルを使える設定になっています。使用したいスクリプトを選択し、それに合わせて出力ファイル名フォーマットの拡張子を変更し監視を開始すると、時間経過に対応する画像が出力されます。例として、棒グラフ用スクリプトを使用すると次のような画像が出力されます。



監視直後



監視中盤



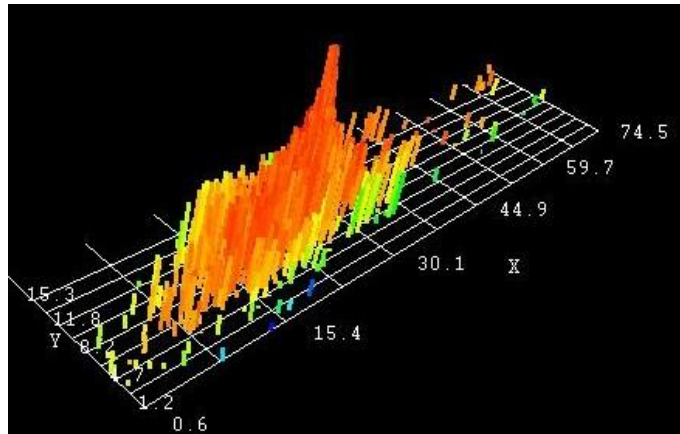
最終結果



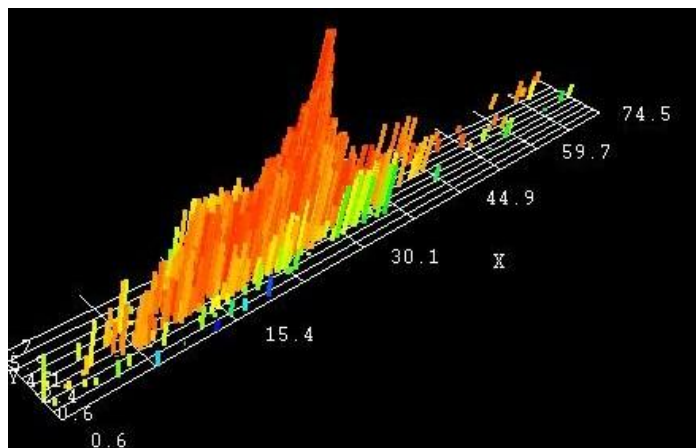
サンプルスクリプトを編集することで画像の出力方法を変更することもできます。簡単に変更できるものとして、xyz 各軸のスケールやオフセットの変更があります。各スクリプトの末尾近くには以下のようなコメント付きの部分があり、数値を変更することでスケールやオフセットを変更できます。

```
//  
// SGI Comment  
// operation_x : x-axis scale and offset  
// operation_y : y-axis scale and offset  
// operation_z : z-axis scale and offset  
// MODS.coordinate_math coordinate_math<NEx=440.,NEy=121.> {  
    in_field1 => <-.table_to_scatter_field.out fld;  
//    operation_x = "#1x + 100";  
//    operation_y = "#1y * 10";  
//    operation_z = "#1z * 0.01";  
};  
//  
//
```

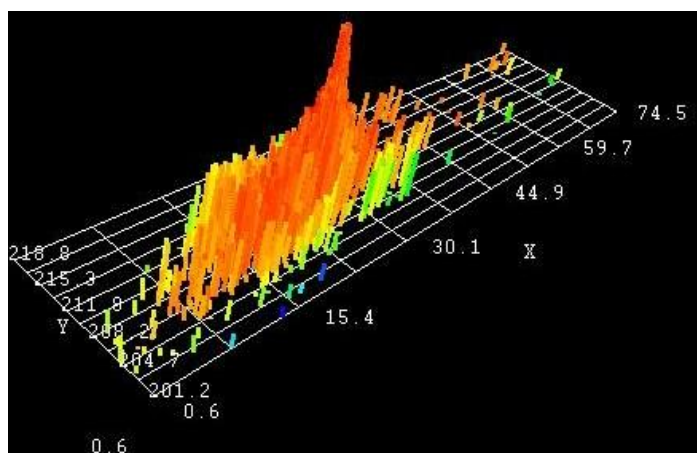
棒グラフのサンプルスクリプトで `operation_y` の値を操作した結果は次のようになります。



`operation_y = "#1y*10";`(初期状態のまま)



`operation_y = "#1y*5";`(y 軸のスケールを半分に変更)



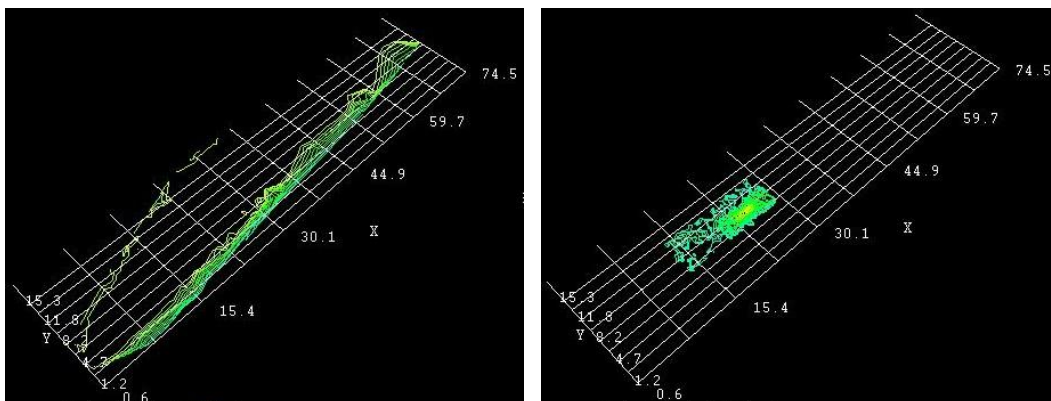
`operation_y = "#1y*10+200";`(y 軸の値にオフセットを追加)

さらに、等高線用スクリプトでは値域の最小・最大値の設定や、比較したい値の変更(データあるいはインデックスから選択)ができます。

```
MODS.isoline isoline<NEx=55.,NEy=363.,NExOffset=13.,NEyOffset=96.> {
  in_field => <-scat_to_tri.out_fld;
  IsoParam<NEx=242.,NEy=77.> {
    // SGI Comment
    // If you want to set min/max value directly, Please put the number on
    // level_min/level_max. Otherwise they are set by default value.
    //
    //level_min => 0.;
    //level_max => 1999.;

    // contour_comp selects value of the data or index.
    // contour_comp=0 means contour is drawn by the value.
    // contour_comp=1 means contour is drawn by the index
    //contour_comp = 1;
    contour_comp = 0;
  };
};
```

インデックスとは、入力データファイルにおける行番号です。途中結果をファイルに追記していくケースでは、インデックスが大きいほど最新に近いデータということになります。この場合、インデックスを比較対象として指定することで、等高線の密度から値の収束等を判断することができます。



左: 値を比較対象としたときの等高線

右: インデックスを比較対象としたときの等高線

## 6.3 gnuplot 用サンプルスクリプト

ismprsm には rviz で使用できる gnuplot 用のサンプルスクリプトが 1 つインストールされています(ファイルの置き場所については 2 章をご覧ください)。

line\_chart.gp : 折れ線グラフを png 形式で出力します。

このスクリプトは入力として空白区切りのデータを受け取ります。以下の例のように、各行がある時点での値の集合を表します。

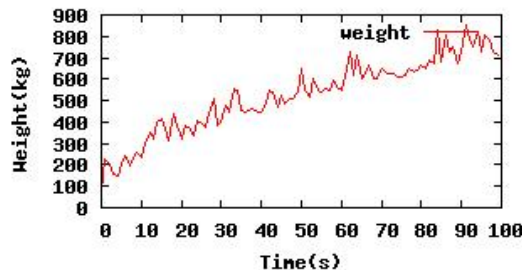
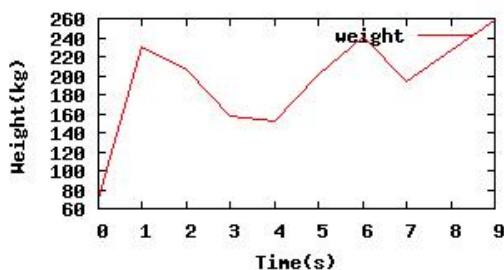
```
x1 y1 z1 w1 ...  
x2 y2 z2 w2 ...  
x3 y3 z3 w3 ...  
...
```

サンプルの入力データファイルとして、000.txt~009.txt の 10 個のファイルをご用意しています(ファイルの置き場所については 2 章をご覧ください)。これらは時間の経過とともに、計算プログラムの途中結果を累積出力したものをイメージしています。

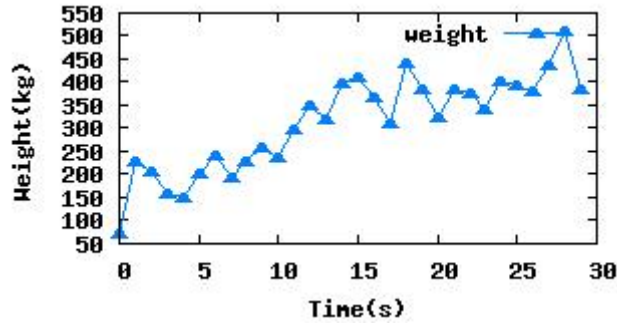
rviz を初期状態で起動(設定ファイル.rviz.properties を削除して起動)し、以下の設定を行うことで gnuplot サンプルを使用できます。

- ① 監視先ディレクトリを input から input\_gp に変更。
- ② 出力画像フォーマットの拡張子を png に変更。
- ③ 使用アプリケーションを gnuplot に変更。
- ④ スクリプト一覧から line\_chart.gp を選択。

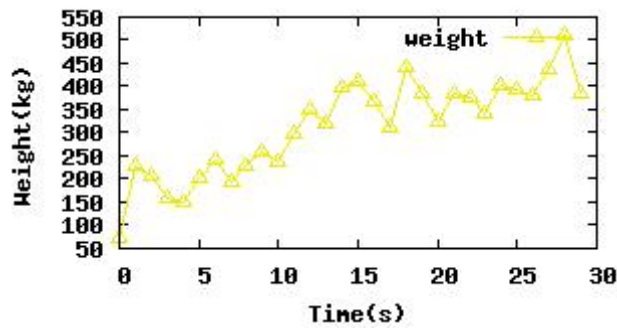
サンプルデータを可視化すると次のような画像が得られます。左が監視直後、右が時間経過後のデータの可視化結果です。



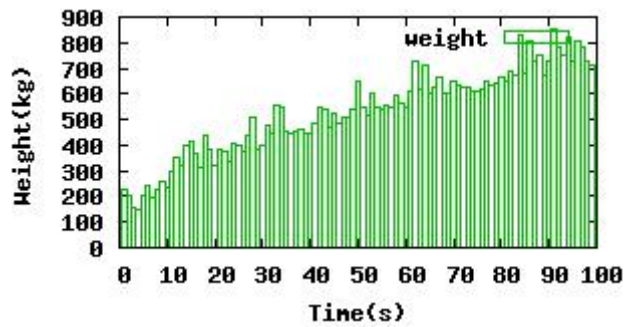
スクリプト内のコメントに従って書き換えることで、簡単にグラフの見栄えを変更することができます。グラフの色・点のスタイルや値域を変更した例を示します。



色を青色に、各点を塗りつぶし三角表示に変更



色を黄色に、各点を三角表示に変更



色を緑色に、形状を棒グラフに変更