

AVS/Express による
気象データの3次元可視化

2016年7月

サイバネットシステム株式会社
ビジュアライゼーション部
AVS サポートセンター

目次

第1章 概要	1
第2章 地図データの可視化	2
2.1 地図情報可視化オプション	2
2.2 外形線の表示	3
2.3 日本の標高地図データ	6
2.3.1 標高データの表示	6
2.3.2 ColormapEditor による色づけの変更	8
2.4 その他の標高地図データ	11
2.4.1 地図データの入手	11
2.4.2 GISDataExtractor の確認	13
2.4.3 GISDataExtractor を利用したデータ変換	14
2.4.4 AVS/Express による表示	17
2.5 標高地図データの間引き	20
2.6 可視化状態の保存	21
第3章 気象解析データの可視化	22
3.1 データの読み込み	22
3.2 スケール、境界表示、軸表示	35
3.3 等値面	37
3.4 ボリュームレンダリング	42
3.5 断面コンター図、等圧線など	49
3.6 ベクトル図	55
3.7 流線	58
3.8 各モジュールの組み合わせ	62
第4章 色、その他の属性の変更	63
4.1 カラーマップの色の調整	63
4.2 オブジェクトの属性の変更（色や反射）	70
4.3 その他の属性	72
第5章 時系列データの扱い	75
5.1 Multi Files モジュール	75
5.2 Loop モジュール	78
第6章 画像、動画ファイルへの出力	80
6.1 静止画の保存	80
6.2 動画の保存	81
6.3 連番画像の保存	84
6.4 連番画像の保存（image_capture の Disk モードの利用）	86

第1章 概要

豪雨、竜巻、台風などの自然災害を防ぐための取り組みの一環として、レーダーによる気象観測や、気象データの予測、現象把握のための数値解析などが行われています。

技術はもとより、コンピューターをはじめ、各種機器、装置の発達により、3次元データとして計測、解析が行われるようになってきました。それに伴い、その結果の検証を3次元で行うニーズも増えてきています。その他にも、情報公開や説明など、一般人にもわかりやすい情報の伝達方法としての可視化ニーズも増えていきます。

本書では、3次元気象データを AVS/Express を用いて可視化する方法について紹介しています。

なお、AVS/Express には、本書の執筆時の最新バージョン、V8.3 を利用しています。

なお、本書は、初めて AVS/Express を使われる方を対象に、その基本的な操作やヒントについて説明しているつもりです。この先は、各種 AVS/Express のマニュアルをはじめ、その他のドキュメントを参考にしてください。

第2章 地図データの可視化

気象データは、地形と密接に絡んでいるケースがほとんどです。そのため、日本地図や関東地図のように、地図データと一緒に表示することで、その位置関係が把握できるようになります。

そこで、まず、この章では、地図データに関する情報について説明します。例えば、デジタル標高地図データを AVS/Express に読み込み、凹凸をつけた鳥瞰図として表示します。

2.1 地図情報可視化オプション

地図の可視化に関するご要望も多いため、AVS/Express のオプション製品として、地図情報可視化オプション（別売）を 2015 年にリリースしました。

国土地理院発行の以下の地図を、可視化の背景画像として表示できます。

- 地図画像（標準）
- 地図画像（写真）
- 基盤地図情報（海岸線や市区町村境界、河川、建物など）
- 標高地図（広域から 10m/5m メッシュ）

以下に代表的な地図の表示例を示します。

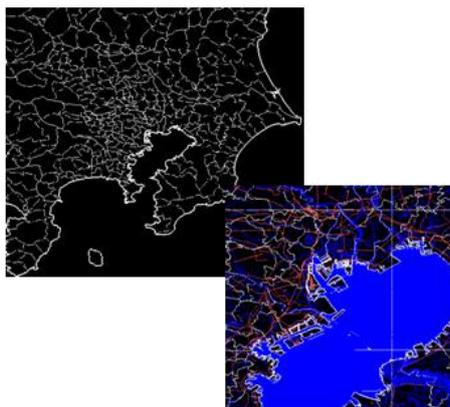


図 1 市区町村境界や河川域等の表示



図 2 地図画像と都道府県面表示

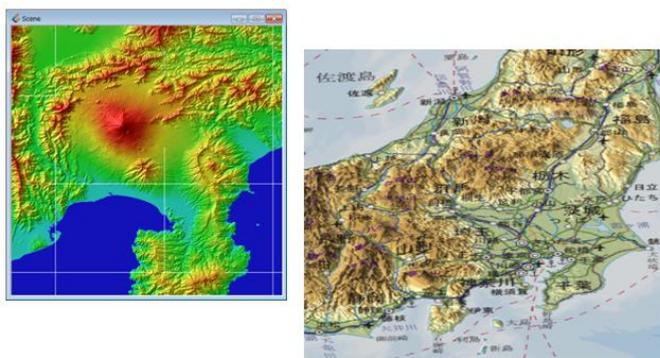


図 3 標高データ、ならびに、地図画像との合成

次節から、AVS/Express 標準で利用可能な、地図の外形線や標高データの表示について説明しますが、さらに詳細な地図を表示したい場合には、本オプションについて、ぜひ、ご検討ください。以下の URL もあわせてご参照ください。

<http://www.cybernet.co.jp/avs/products/avsexpress/option/map.html>

AVS/Express をご利用の場合、本オプションの評価も可能です。サイバネットシステム（株）のサポートセンターや営業窓口等にお問い合わせください。

2.2 外形線の表示

データとしては古いですが、AVS/Express の中に、CIA World Bank II の外形線データが含まれています。このデータを利用して、地図の外形線を作成できます。

このデータの読み込みと表示は、以下の方法で行います。

3番目のポートを DataObject モジュールに接続しています。

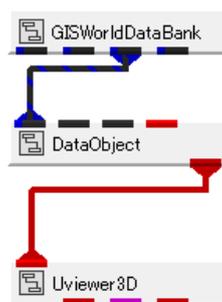


図 4 GISWorldDataBank モジュール

※ モジュールの在り処がわからない場合、オブジェクトメニューにある、すべてのライブラリを検索メニューで、検索してみてください。

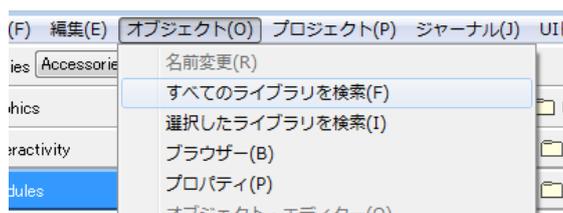


図 5 モジュールの検索

モジュールのパラメーターで、以下の設定を行います。

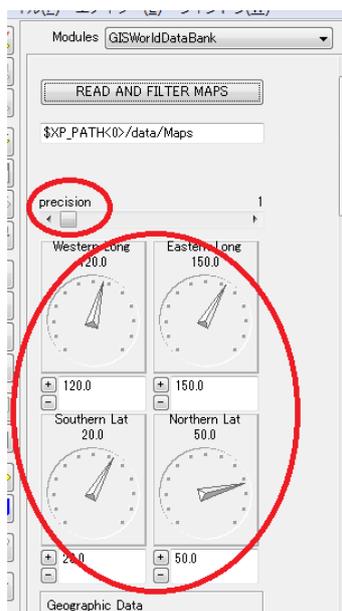


図 6 緯度経度範囲の指定

まず、precision を 1 に設定し、Long の範囲を 120 度から 150 度 に設定しています。また、Lat の範囲を 20 度から 50 度 に設定します。

次に、下図の 3 番目のブロックにある Coastal Data から Major を選びます。

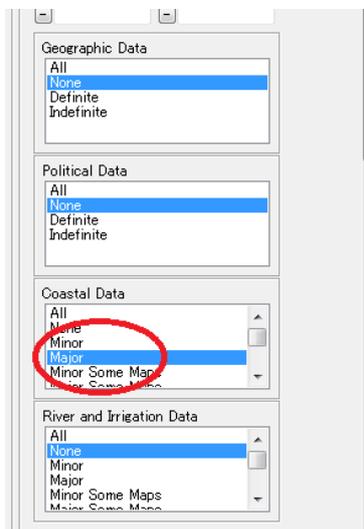


図 7 読み込み対象の選択

設定できたら、一番上にある（図 6 緯度経度範囲の指定）、READ AND FILTER MAPS ボタンをクリックします。日本近傍の外形線が表示されます。

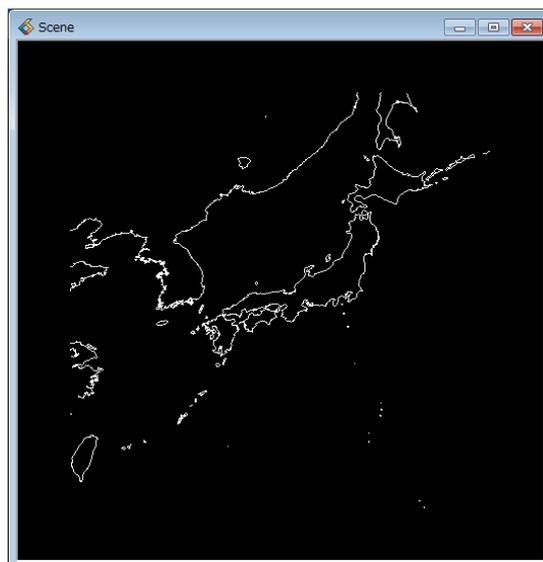


図 8 日本外形線の表示

図 7 読み込み対象の選択の 4 つのブロックは、GISWorldDataBank モジュールの 4 つの出力ポートにそれぞれ対応しています。河川などもありますが、残念ながら、日本の範囲には、その他のデータは含まれていません。

データの座標は緯度経度です。この地図と重ね合わせるには、ご利用のデータの座標値が緯度経度で定義されている必要があります。

注) 都道府県や市区町村境界などのデータは含まれていません。
必要な場合、地図オプションをご検討ください。

注) この各ブロック内の All や Major、Minor などの切り替えを試す場合、一度 None で読み込みを行った後、他の項目を選択、読み込みを行ってください。
(更新されない場合があります)

なお、ビューの中の操作は、マウス左ボタンで回転、マウス中ホイールで拡大縮小、Ctrl+マウス中ボタンで移動できます。

また、線の色や線幅は、「第 4 章 色、その他の属性の変更」に述べるオブジェクト・エディターで変更できます。その他、平行投影と透視投影の切り替えや双方向ライトのオンなど、適宜切り替えてみてください（「4.3 その他の属性」参照）。

※ 操作方法については、チュートリアルガイドやユーザズガイドもあわせて、ご参照ください。

2.3 日本の標高地図データ

標準のサンプルデータを利用して日本の標高地図を作成します。

2.3.1 標高データの表示

標高データ (DEM) の表示には、surf_plot モジュールを利用できます。

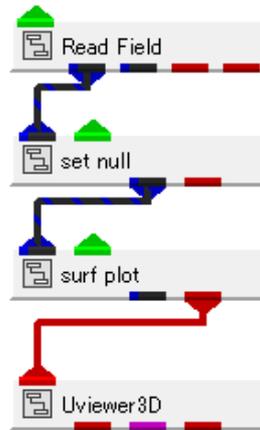


図 9 標高データの表示

各モジュールのパラメーターを設定します。

まず、Read_Field モジュールで、データの読み込みを行います。

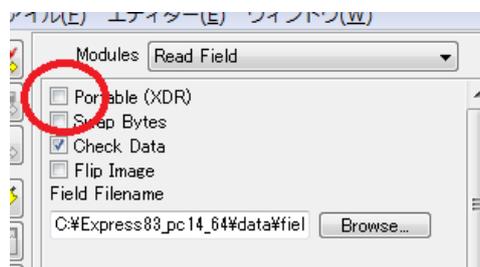


図 10 japan.fld の読み込み

Portable(XDR)をオフに設定してください。

次に、Browse ボタンをクリックし、以下のファイルを読み込みます。

`c:\Program Files\AVS Express\data\field\japan.fld`

次に、set_null モジュールです。

Null Value に、-9999 を設定 (入力し、リターンで確定) します。このデータは、海域に未定義値として -9999 の値が設定されています。set_null モジュールで、その値を null 値として扱い、非表示にします。

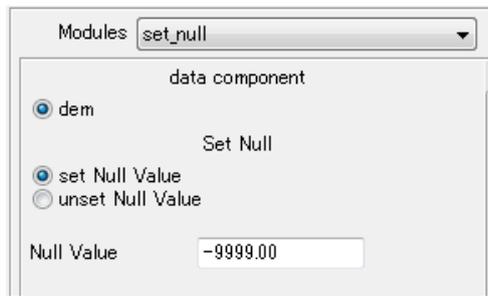


図 11 set_null モジュール

surf_plot モジュールは、メッシュデータに対して、その値を元に、形状の凹凸を作成するモジュールです。

以下の図にある方法で、scale パラメーターを設定してください。

scale パラメーターは、高さ方向の倍率を調整するパラメーターです。デフォルトではこのデータに対して桁数が足りません。スライダーの右にあるボタンをクリックし、スライダー値の数値指定パネルを開きます。桁数を増やした後、Value に値を設定（この例では 0.00001 を入力）してみてください。

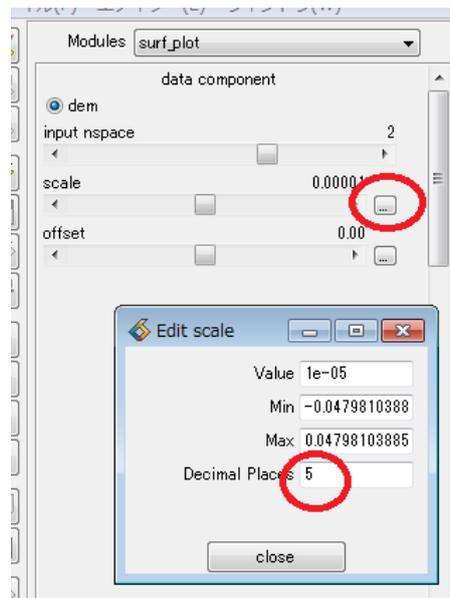


図 12 surf_plot モジュール

以上の設定で、下図に示すように、標高データの色づけ表示を行うことができます。

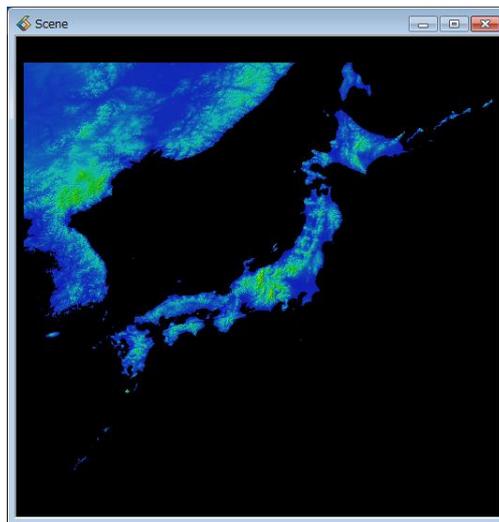


図 13 標高データの表示

2.3.2 ColormapEditor による色づけの変更

AVS/Express では、デフォルトでは青から赤のカラーマップによる色づけが行われます。

例えば、前項の標高データの表示を地形を表す茶系の色を使って色づけるには、ColormapEditor モジュールを利用します。図 9 標高データの表示のネットワークに、以下のように追加します。

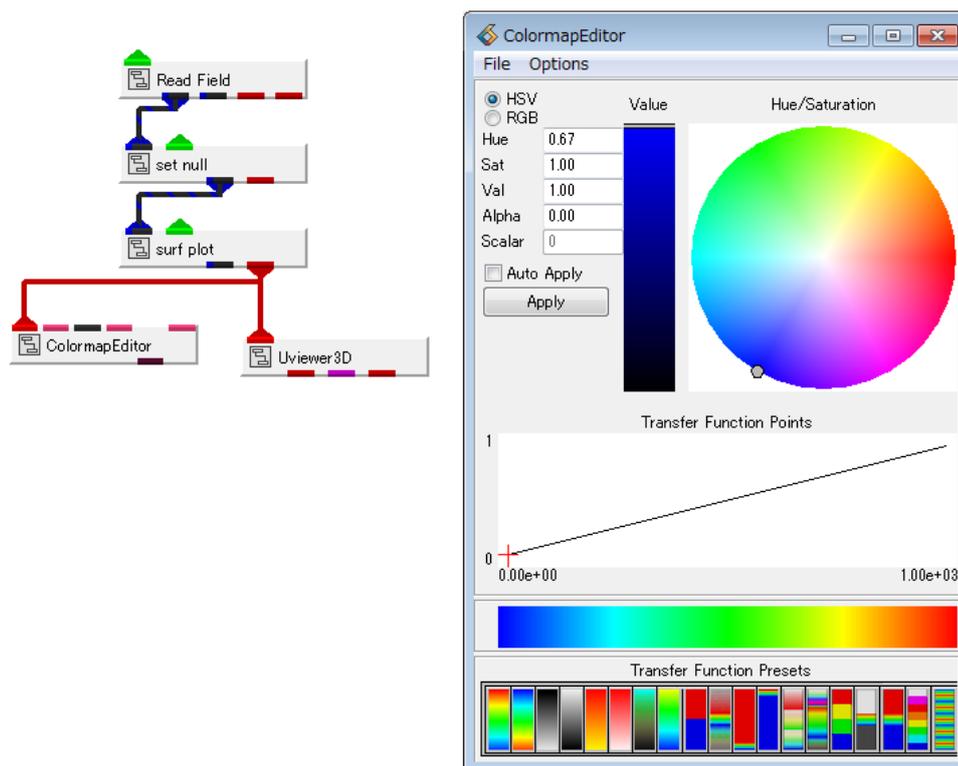


図 14 ColormapEditor モジュール

ColormapEditor は、その表示しているオブジェクトのカラーマップを変更できるモジュールです。

直接、このパネルの操作で色を変更することもできますが、微調整等は難しいため、カラーの対応を記したファイル (.cmp) を作って、その色を適用してみてください。

今回の例では、以下に示すファイルを使用します。

※ このファイルは、以下の URL で公開されている旧パッチファイルに含まれていますので、必要に応じて、ダウンロードしてみてください。

URL : <http://www.cybernet.co.jp/avs/support/avsexpress.html>
(GIS_20130514.zip (2013/5/14) をクリックして取得します)

ファイル `srtm3_earth.cmp` :

```
13 0 1000
0.000000 1.000000 0.100000 0.400000 0.700000
2.500000 1.000000 0.100000 0.400000 0.700000
2.610000 1.000000 0.000000 0.400000 0.500000
2.750000 1.000000 0.000000 0.600000 0.500000
5.000000 1.000000 0.008368 0.646400 0.051766
25.701859 1.000000 0.537239 0.620630 0.000000
32.868042 1.000000 0.727655 0.583903 0.000000
59.148582 1.000000 0.694475 0.516603 0.000000
90.660805 1.000000 0.665000 0.442149 0.007669
115.865097 1.000000 0.629259 0.387560 0.000000
115.967097 1.000000 0.581308 0.366887 0.000000
157.325516 1.000000 0.555326 0.254091 0.006227
255.000000 1.000000 0.433000 0.000000 0.000000
```

数値と色の対応を設定しています。

1 行目は、以下の設定です。13 個の制御点があり、0 ~ 1000 の数値に対する色の分布を定義しています。

制御点の数	数値の最小値	数値の最大値
-------	--------	--------

2 行目以降が色の定義です。13 行あります。

制御点の位置は、0 ~ 255 に正規化した値で定義します。この例では、255 が数値の最大値 1000 に対応します。0 ~ 1000 に対して、0 ~ 255 の範囲で位置を決めます。

次の α 値はボリュームレンダリングに利用する透明度で、面の表示では利用しません。

その後、RGB の各値を設定しています。 α 、RGB とも、0 ~ 1 の範囲で指定します。

制御点の位置	α 値	R 値	G 値	B 値
--------	------------	-----	-----	-----

拡張子を `.cmp` としてファイルを準備します。準備できたら、ColormapEditor の File メニューの Load メニューで指定します。

下側のカラーバーが定義した色に変わりますので、Apply ボタンをクリックしてください。

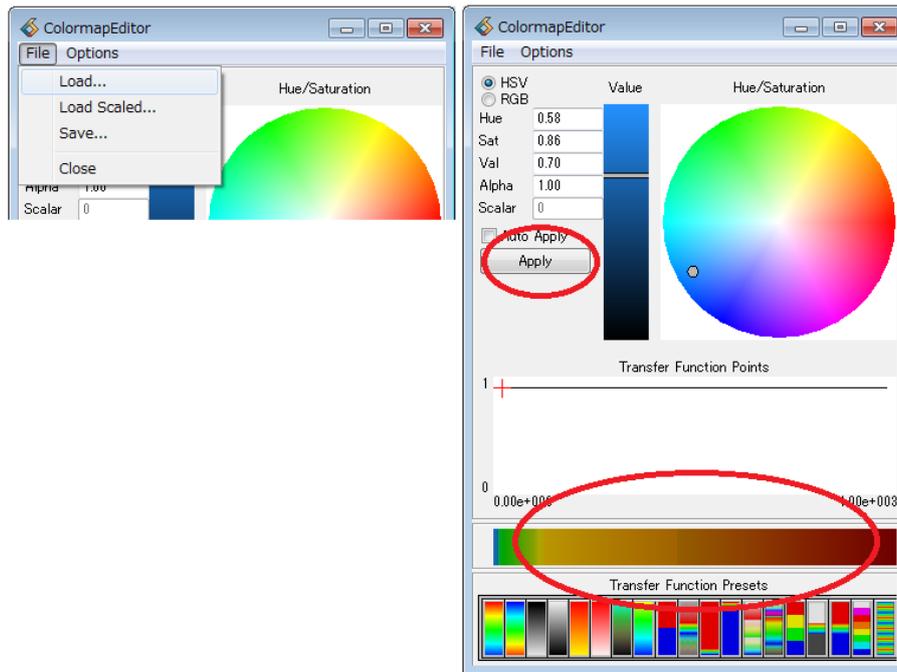


図 15 .cmp ファイルの読み込みと適用

地形の表示が指定したカラーマップの色に変わります。

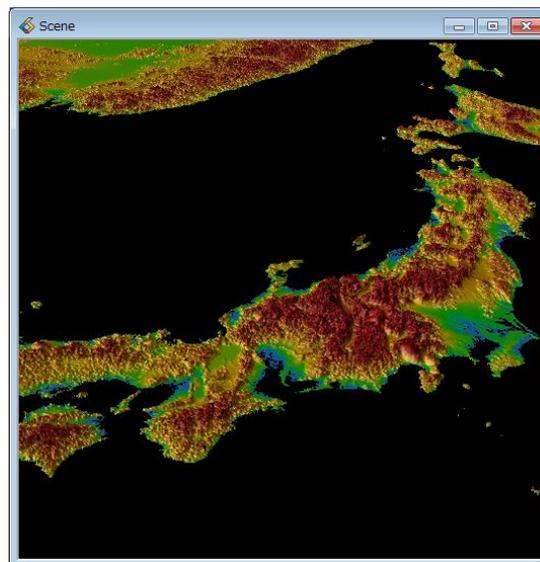


図 16 地形っぽくカラーマップを変更

※ この japan.fld は、次項に示す SRTM3 データを修正したデータです。

2.4 その他の標高地図データ

その他、標高地図データをダウンロードし、AVS/Express で読み込めるデータ形式に変換して利用する方法もあります。

2.4.1 地図データの入手

ここで説明する方法は、AVS/Express の外部ツール、GISDataExtractor（詳細は後述）を利用して、ダウンロードしたデータを AVS/Express で読み込める形式に変換します。

※ GISDataExtractor は Windows で利用できる外部ツールです。
AVS/Express に同梱されています。

以下のデータに対応しています。

- SRTM3 v2 (The Shuttle Radar Topography Mission)
- GTOPO30
- 国土地理院 標高数値地図データ (旧データ)
1km メッシュ(TEM)/250m メッシュ(SEM)/50m メッシュ(MEM)

注意：

各地図データの利用条件、使用許諾については、ご利用になる皆様にて、ご確認いただけますようお願いいたします。

例えば、国土地理院の地図利用には、利用用途に応じた規約があります。

また、以下の入手方法は、本ドキュメント執筆時点での情報です。

詳細は各 URL の他、Web 上で得られる情報などをご参照ください。

まず、利用したいデータを以下の情報を参考に、ダウンロードします。

1) SRTM3 データ

このデータは、アメリカ合衆国 NASA で測定されたデータです。

以下の URL をご参照ください。

参考：<http://www2.jpl.nasa.gov/srtm/>

GISDataExtractor では、以下の URL からダウンロードできる SRTM3(version2_1) に対応しています。

ダウンロード：http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/

このデータは、3 秒 (約 90 m) の解像度で取得されています。ただし、スペースシャトルによるレーダー観測のため、観測できなかった地点は、データが欠落しています。

ファイルをダウンロードできたら、以下のように配置します。ご利用になりたい緯度経度範

囲のデータを取得、zip 圧縮されていますので、展開してください。
(日本は Eurasia に含まれています)

ファイルの配置例) D:\¥地図データ¥SRTM3¥N20E121.hgt
N20E122.hgt
N21E120.hgt
....

2) GTOPO30 データ

GTOPO30 は、全球レベルの標高データで、30 秒 (約 1km) の解像度のデータです。アメリカ合衆国の地質調査所 (EROS) を中心に各関連機関により作成されました。

ダウンロード等の情報 : <https://lta.cr.usgs.gov/GTOPO30>

地域によっては、品質が悪い部分もあると言われています。1) のスペースシャトルで測定された SRTM30 がこの GTOPO30 に変わるデータとして知られています。また、船を使った水深測量で測定された海底標高データを含む SRTM30plus もあります。

※ 本書執筆時点では、SRTM30 や SRTM30plus には対応していません。

ファイルをダウンロードできたら、以下のように配置します。ご利用になりたい緯度経度範囲のデータを取得、tar 圧縮されていますので、展開してください。

ファイルの配置例) D:\¥地図データ¥GTOPO30¥E100N40.DEM
E100N40.DMW
....

3) 国土地理院 標高数値地図データ

Web からダウンロード販売、もしくはメディアで販売されていた国土地理院の標高数値データです。1km メッシュ(TEM)、250m メッシュ(SEM)、50m メッシュ(MEM) に対応しています。

現在ダウンロードできる基盤情報ではありませんので、ご注意ください。

オンライン等で入手している場合は、それらのデータをあるフォルダにコピー (配置) してください。ディスク (1 枚) の場合は、そのままマウントしている状態でも構いません。

ファイルの配置例) D:\¥地図データ¥国土地理院¥DATA¥250M¥3036.SEM
3622.SEM
....

※ 該当解像度のデータを配置します。

1) ~ 3) のデータが準備できたら、ツールを使って、AVS 用のデータを作成します。

2.4.2 GISDataExtractor の確認

GISDataExtractor は、前節で述べたデータを AVS/Express で表示できるフィールド形式のファイルに変換するツール（Windows 版のみ）です。

Windows 版の AVS/Express をご利用の場合は、AVS/Express のインストールフォルダに同梱されています。その他の機種で AVS/Express をご利用の場合は、適当な Windows マシンにて、AVS/Express のインストールメディア（DVD）にあるツールを実行します（適宜、適当なフォルダにコピーしてご利用ください）。

c:\Program Files\AVS/Express\runtime\tool\GISDataExtractor
[DVD]\tool\GISDataExtractor

実行バイナリは、GISDataExtractor\bin\GISDataExtractor.exe です。
まずは、この exe を実行（ダブルクリック）し、ツールが起動できるかをご確認ください。

※ ツールが起動できない場合、サポート窓口までお問い合わせください。

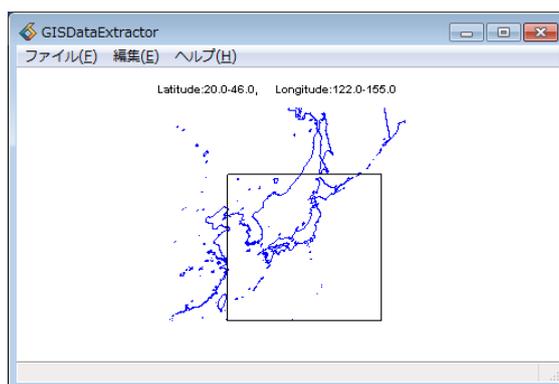


図 17 GISDataExtractor の起動

2.4.3 GISDataExtractor を利用したデータ変換

ツールの使い方について説明します。

1) 抽出範囲の指定

では、実際にデータの抽出、変換を行ってみましょう。

まずは、抽出したいデータの範囲を指定してみてください。

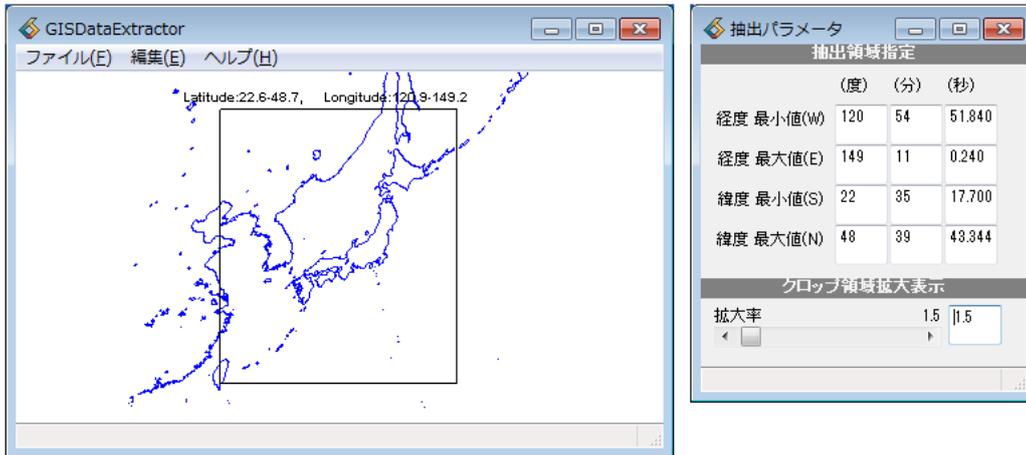


図 18 日本全体を抽出した例

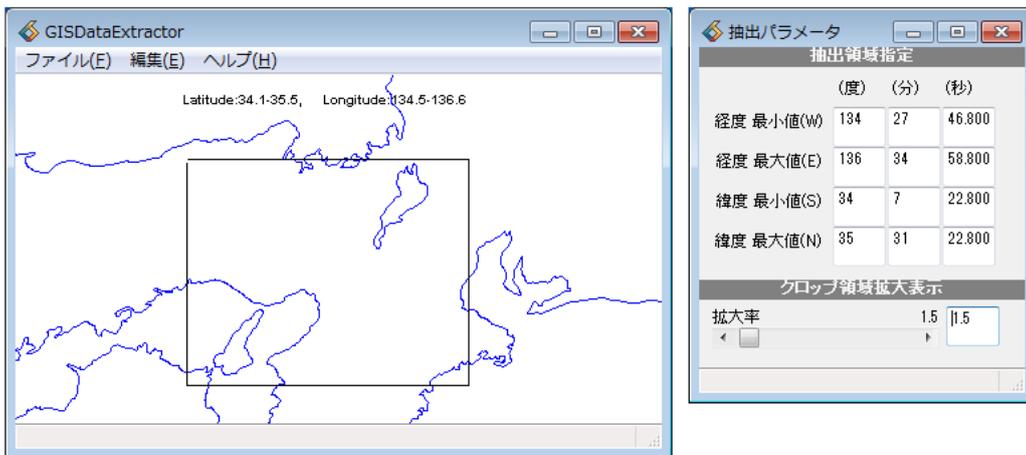


図 19 関西領域を抽出した例

編集メニューから抽出パラメータ設定を選びます。
度分秒で緯度経度の範囲を指定するようになっています。

※ デフォルトでは、選択対象を示すための地図は日本の周囲となっています。
抽出したい範囲が異なる場合、次の 2) で述べる地図の表示範囲の変更を
先に行ってください。

表示されている地図は、マウス左ボタンを押したまま、移動することができます。
また、マウスホイールを使って拡大縮小（もしくは、上記右パネルの拡大率で操作）する
こともできますので、適当な大きさに変更してみてください。

※ マウスを勢いよく放すとどこかに行ってしまうかもしれません。
その際は、上記の拡大率のスライダを一度触ってみてください。

なお、緯度経度の値を 10 進数でお持ちの場合は、Excel 等を使って、度分秒に変換してから入力してください。

※ Excel を使った 10 進数の度分秒変換を行うには？
例えば、以下の計算式を作成します。

A1 : 10 進表記の値
B1 = TRUNC(A1) : 度の計算
C1 = TRUNC((A1-B1)*60) : 分の計算
D1 = (A1-(B1+C1/60))*3600 : 秒の計算

2) 地図データのタイプとフォルダの指定

範囲が決まったら、次に、編集メニューにある抽出データ選択を選びます。

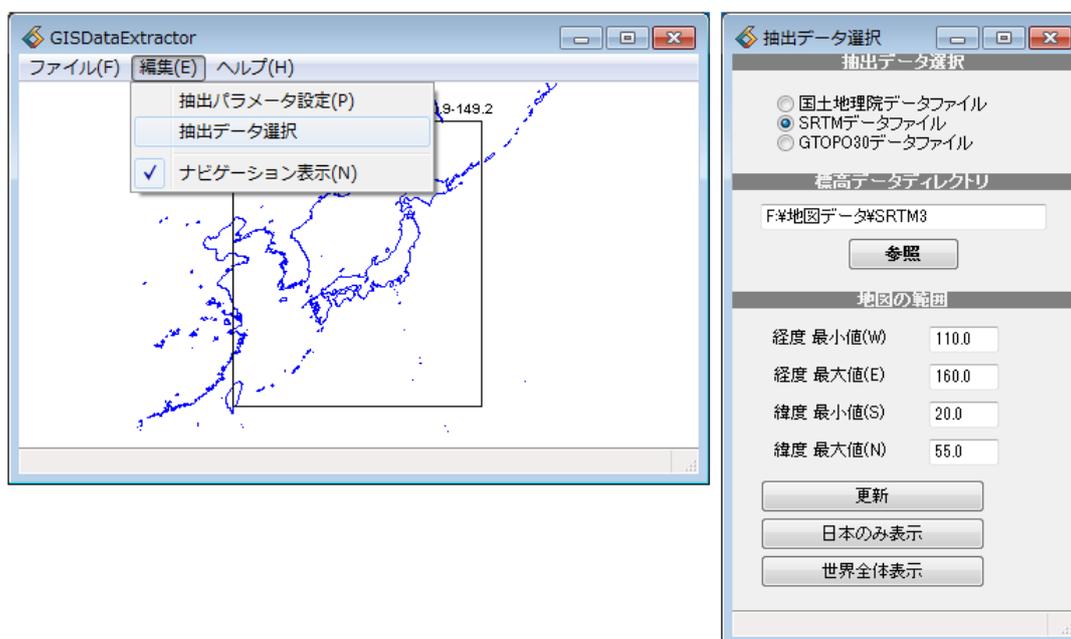


図 20 抽出データの選択

図の例では、SRTM3 のデータを選択（右図の選択）し、次に、参照ボタンを使って、その SRTM3 のデータをダウンロード、展開したフォルダを指定しています。
この 2 箇所の設定ができれば、次に進みます。

なお、この下にある地図の範囲は、ラインで表示している地図の表示範囲を示しています。
デフォルトでは、日本の周囲を表示しています。
より広範囲のデータの抽出や、地域の異なる領域での抽出を行う場合、この地図の範囲のパラメーターにある緯度経度を変更してみてください。

3) データ変換の実行

いよいよ変換処理を実行します。

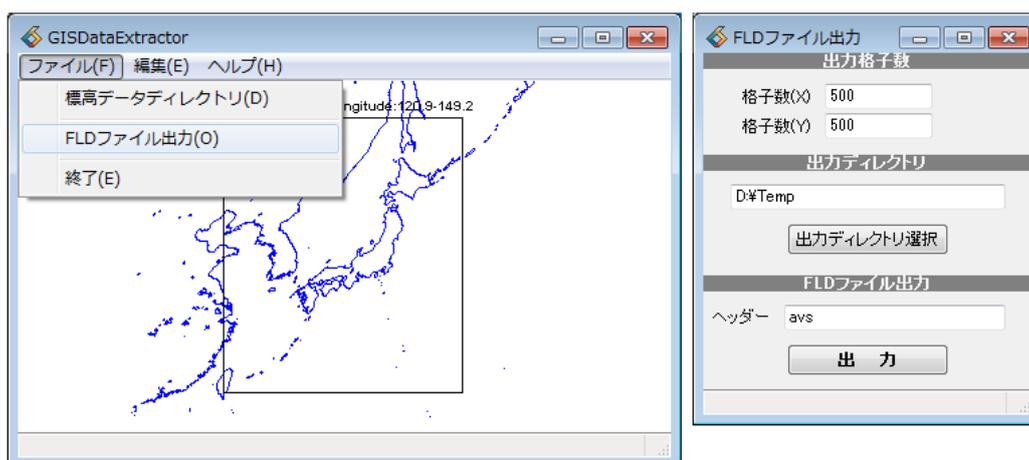


図 21 変換実行

ファイルメニューにある、**FLD** ファイル出力を選びます。

まず、格子数を指定します（図の例では 500 × 500）。

この格子数については後述します。

まずは、500 より小さい値くらいでテストしてみてください。

次に出力ディレクトリ選択ボタンをクリックし、適当なフォルダを指定します。

出力実行すると、その下にあるヘッダーに指定した名前+.fld と 名前+.dat、
名前+.cod の3種類のファイルが作成されます。

上記例では、以下のファイルが作成されます。

D:\Temp\avs.fld	: 1KB (アスキー)
D:\Temp\avs.dat	: 1,382KB (バイナリ)
D:\Temp\avs.cod	: 1KB (アスキー)

4) 変換時に指定する格子数について

このツールは、もともとのデジタル標高データから、線形補間を行い、新しくメッシュを再生成します。

この補間によって、元々の標高データの最大最小値は消えてしまいます。

解析結果や測定結果との重ね合わせによる表示効果を目的としているため、
その点についてはご了承の上、ご利用ください。

上記変換時に指定する格子数を大きくすると、大きなメッシュを作成できますが、
ファイルサイズやレンダリング時のリソースも多く消費してしまいます。

また、元々のデータの解像度に比べて、極端に大きな値を指定しても、あまり意味は
ありません。

再度、以下の解像度と、ご自身が抽出しようとしていらっしゃる範囲を考慮して、
適当な値の格子数を指定してみてください。

・ SRTM3	約 90 m メッシュ
・ GTOPO30	約 1km メッシュ
・ 国土地理院	50m メッシュ
	250m メッシュ
	1km メッシュ

例えば、先の例では、SRTM3 を利用しています。

日本全体では、2000km くらいですので、本来は、20000 × 20000 くらいの指定がいいかもしれません。

ただし、変換にも 4, 5 分以上、データサイズは、約 900 Mbyte となります。

表示上は、AVS/Express 上でも `downsize` モジュールを使って間引くこともできますが、2000 × 2000 くらいの設定でも十分な凹凸を見ることはできます。

2.4.4 AVS/Express による表示

変換したデータを AVS/Express で表示する方法は、先の「2.3 日本の標高地図データ」の表示と同じです。

`surf_plot` モジュールを利用し、メッシュ状の各頂点の高さに応じて、凹凸を付けて表示します。図 9 標高データの表示を再度、ご参照ください。

また、データの読み込みモジュール `Read_Field` の `Portable` も同様にオフに設定します。

例えば、SRTM3 を使った場合の表示例を以下に示します。

左図は 500×500 の格子数で、右図は 2000×2000 の格子数です。

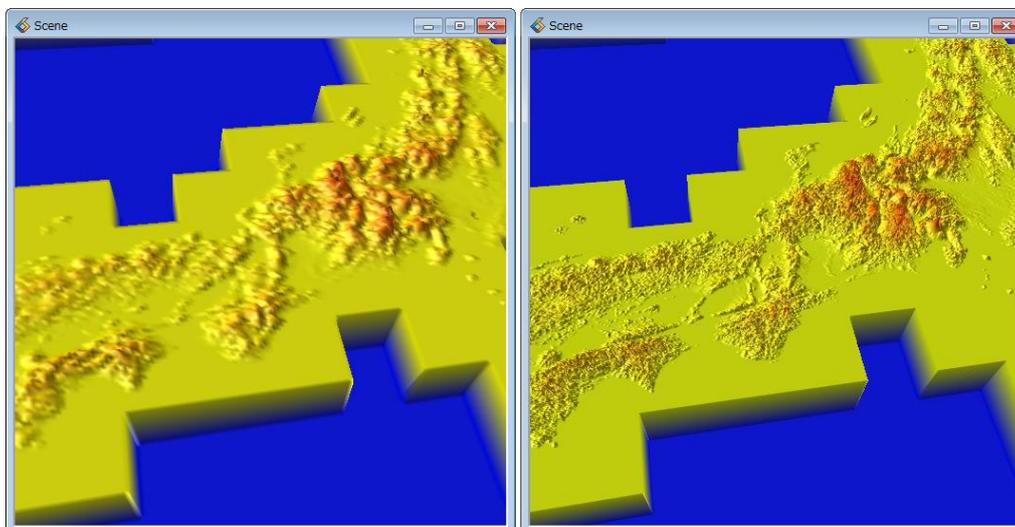


図 22 SRTM3 データの表示

また、下図は、GTOPO30 を使って変換した結果 (2000x2000) の表示例です。

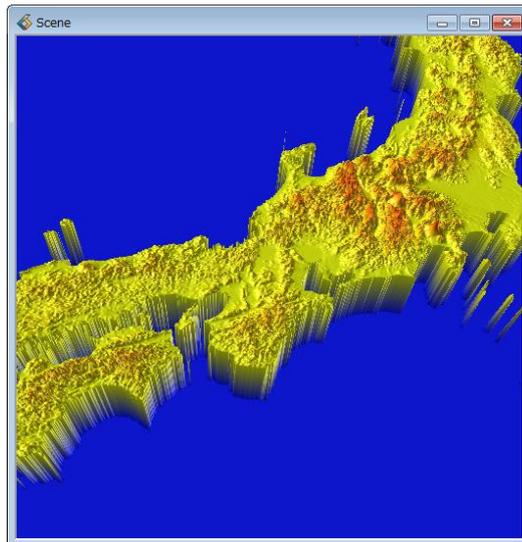


図 23 GTOPO30 データの表示

欠損データや未定義値の違いによって、表示が異なります。

「2.3.1 標高データの表示」に述べたように、欠損値は `set_null` モジュールを使って、非表示にすることができます。

先の例同様、-9999 を `null` 値として設定します。

下図のように、欠損データが非表示になります。

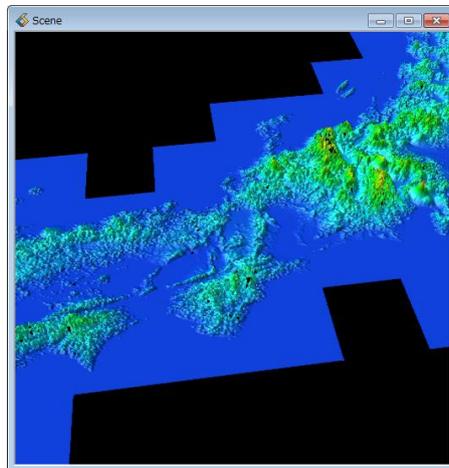


図 24 欠損値を非表示に変換

また、SRTM3 のデータでは、ところどころ欠損値が存在するため、陸地の中にも非表示部分があることがわかります。

SRTM3 のデータの場合には、以下のネットワークを使って、海面に相当する部分もすべて非表示にした方がいいかもしれません。

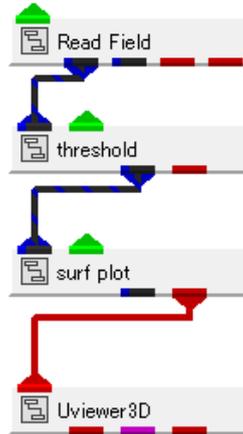


図 25 threshold モジュールの利用

• threshold モジュール

threshold モジュールのパラメーターで、Below min value にチェックし、min value にて、-1000 を指定しています。

このモジュールはしきい値処理をするモジュールで、-1000 より小さい値をすべて null 値に変換します。

また、この例では、null thresh value = 0 に設定されていますので、元データが 0 についても null 値として扱われます。

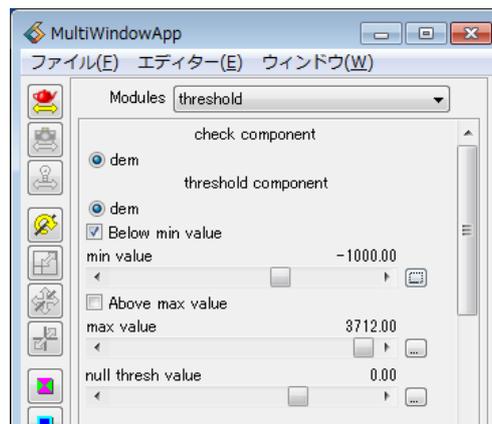


図 26 set_null モジュールのパラメーター

このネットワークを使った結果を以下に示します。

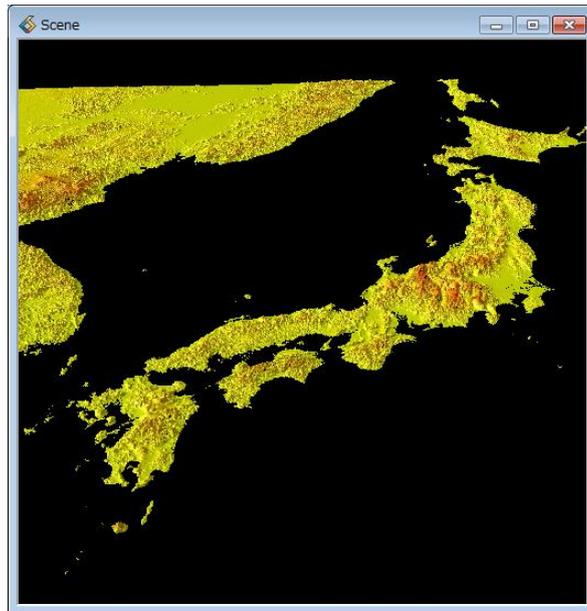


図 27 海面を非表示にした表示例

陸地と海面との境目で、0.0 が含まれる部分も非表示になるため、海岸線が正しく表示できていない部分もありますが、地形のみの表示に代わります。

また、「2.3.2 ColormapEditor による色づけの変更」に述べたカラーマップファイルを使って、色の調整等も行ってみてください。

2.5 標高地図データの間引き

標高地図データは、1km メッシュや 500m メッシュのように、格子状に並んだデータで構成されています。特にデータ変換を行った後のデータの表示が重い場合、AVS 上でデータを間引いて表示してみてください。

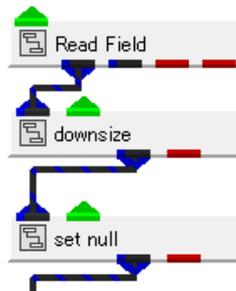


図 28 downsize モジュールの利用

downsize モジュールのパラメーターには、I / J downsize factor というスライダーがあります。例えば、それぞれ 2 に設定すると、元のデータに対して、1/2 に間引くことができます。適宜、絵を確認しながら、調整してみてください。

2.6 可視化状態の保存

地図データの表示ができれば、ひとまず、可視化の状態（ネットワークと呼びます）を保存しておきましょう。

この際、ColormapEditor モジュールは、その上の surf_plot モジュールに色の情報を設定し終えていますので、削除しても構いません。

ファイルメニューから適当な名前（拡張子を .v ）をつけて、保存しておいてください。

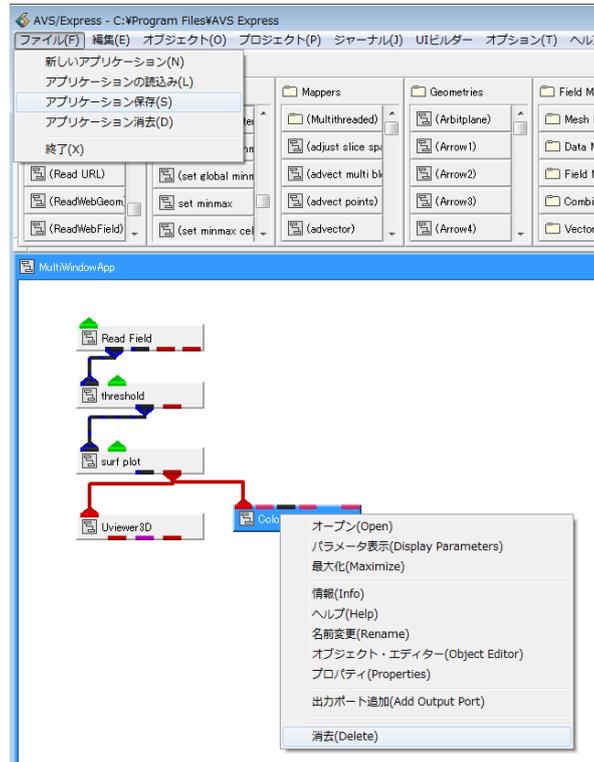


図 29 アプリケーション（ネットワーク）の保存

第3章 気象解析データの可視化

では、いよいよ気象解析データの可視化に移ります。本書では、3次元（もしくは2次元）の構造型メッシュで解析された結果を対象としています。

3.1 データの読み込み

まず、データの読み込み方法について説明します。ここでは、気象データとしてよく知られている、以下の形式のデータを対象としています。

- netCDF
- WRF
- GrADS
- Vis5D

1) netCDF データの読み込み

netCDF データを読み込むには、Rd_netCDF_Fld モジュールを利用します。

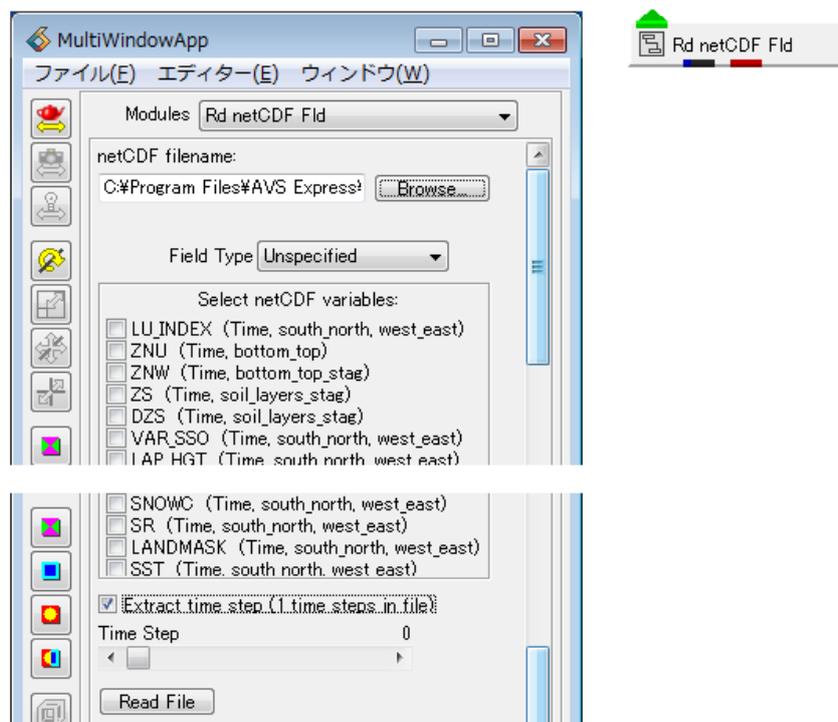


図 30 Rd_netCDF_Fld モジュール

Browse ボタンをクリックし、netCDF データファイルを指定します。

上図に示すように、データファイルに含まれる変数名がパラメーターの欄にリストされます。

AVS/Express の netCDF モジュールは、データを以下のように扱います。

- 対象となるデータ配列について
このモジュールは、2次元以上の配列データを扱います。
1次元で定義されたデータ配列は、上記のリストには表示されません。

また、char 型のデータも対象としていません。

netCDF ファイルの中に含まれている short/int/float/double の数値データで、2次元以上の配列を持つデータがリストされます。

- Time について (UNLIMITED 次元)

time = UNLIMITED の無制限次元が定義されている場合、その time を Time Step として扱います。

この Time Step が認識されている場合は、上図のパラメーターにある Extract time step のチェックボックスと Time Step スライダーが表示されます。

Extract time step にチェックをつけ、抜き出したい Time のステップをスライダーで指定することができます。

※ データに time が定義されていても、その次元が無制限次元でない場合は、配列のひとつとして扱われます。

また、変数名は time 以外でも構いません。

無制限次元の場合は、同様に Time Step のスライダーで抽出できます。

- 配列の扱い

データ配列の一番外側から順番に AVS/Express の Field データの次元、dims[0]/dims[1]/dims[2] の配列に割り当てます。

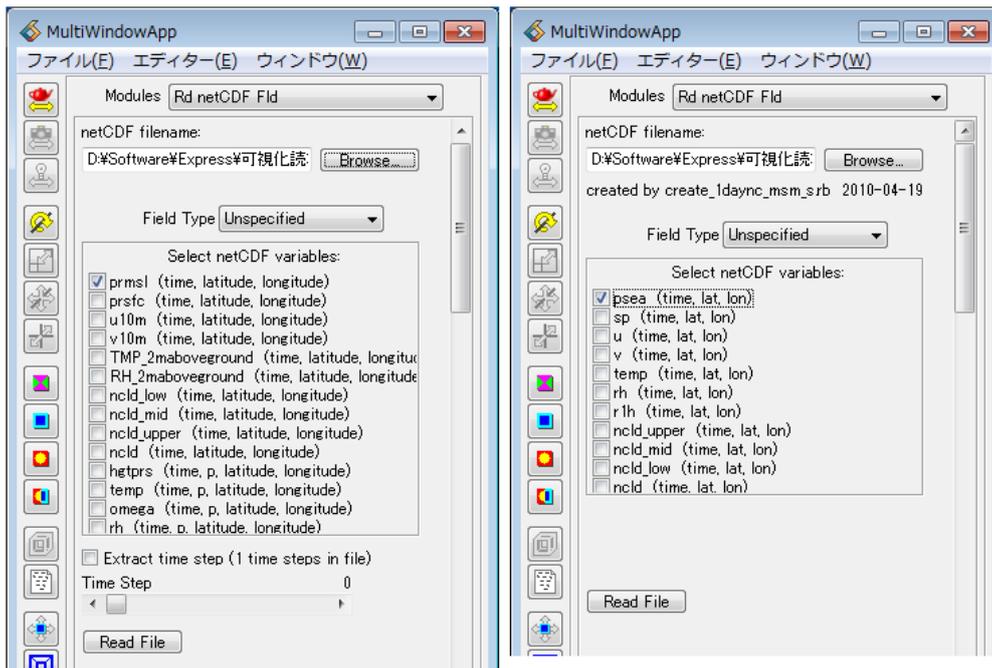


図 31 time の指定の違い

上図は、異なる2つの netCDF データによるパラメーターの表示例を示しています。どちらも、(time, lat, lon) の3次元データが定義されているのを見ることができます。

左図では、Extract time step と Time Step のスライダーが表示されています。

このデータでは、配列の大きさが、以下のように UNLIMITED で定義されています。

```
lon = 120
lat = 46
time = UNLIMITED // (12 currently)
```

このデータは、以下の2種類の方法で読み込むことができます。

- **Extract time** にチェックし、**Time Step** を指定し、2次元配列として読み込む。
`dims = {120, 46}` の2次元配列として読み込まれます。
(**Time Step** で指定した時間のデータとなります)
- **Extract time** にチェックせずに、3次元配列として読み込む。
この場合は、`dims = {120, 46, 12}` の3次元データとして、読み込まれます。

一方、右の図には、**Extract time step** のパラメーターが表示されていません。同じように (`time, lat, lon`) の配列データになっていますが、この `time` が **UNLIMITED** ではなく、定数値で指定されているためです。この場合は、3次元の配列データとして読み込みます。

※ `time` を3次元配列で読み込んだ場合、**orthoslice** モジュールを使って、その時刻を抽出することができます。
可視化方法については、以降の章で説明します。

4次元配列も同様です。

`data(time, p, lat, lon)` のようなデータを4次元で読み込むと、**AVS/Express** 上では、`dims[4] = {lon, lat, p, time}` の4次元配列として読み込まれます。

同様に **orthoslice** モジュールを使って、3次元にすることができます。

- 異なる配列サイズのデータの扱い

例えば、以下の図は、**AVS/Express** の `data\%netCDF` フォルダにある、`sst-t.nc` サンプルデータを読み込んだ例です。

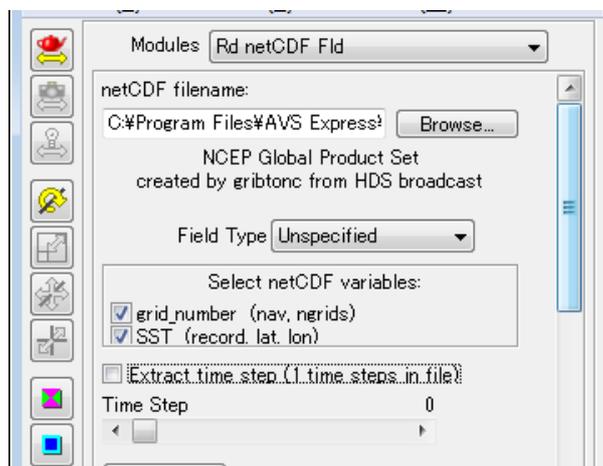


図 32 異なる配列で定義されたデータ

このデータには、以下の2つのデータが定義されています。

```
grid_number(nav, ngrids)
SST(record, lat, lon)
```

※ 実際のデータには `ncdump` コマンドで見ただけであればわかりますが、多数の1次元データや `char` 型のデータが含まれています。サポートしていないタイプのデータはリストされていません。

このように異なる配列データは、一緒に読み込むことはできません。例えば、上図のように両方にチェックして読み込むと、以下のエラーが発生します。

```
--- Warning from: module: netcdf read_field ---
Dimensions of variable SST do not match dimensions of variable grid_number
-----
```

同時に読み込むことができるデータは、同じ配列である必要があります。配列の異なるデータを同時に扱いたい場合は、もう1つ `Rd_netCDF_Fld` モジュールをインスタンスし、同じデータファイルに対して、異なる変数を指定してください。

・座標値の扱い

このモジュールは、1次元配列のデータで、且つ、データ配列の大きさと同じ変数名の配列が定義されている場合は、そのデータを座標値として扱います。例えば、以下の `netCDF` データの場合、

```
latitude = 360;
longitude = 720;
time = UNLIMITED ; // (1 currently)
p = 5;

double latitude(latitude);
double longitude(longitude);
float p(p);
float temp(time, p, latitude, longitude);
```

読み込み対象のパラメーターには、`temp` のみが表示されます。

この `temp` を `Extract time step` にチェックをつけて、`Time Step = 1` を指定して読み込みます。

3次元データとして読み込まれますが、`double longitude()` に出力されている値が `x` 座標に、`double latitude()` の値が `y` 座標に、`float p()` に出力されている値が `z` 座標に割り当てられます。

※ ただし、`AVS/Express` の座標値は、`float` 型のみがサポートされています。`double` の場合も `float` 型にキャストされます。

下図は、可視化例です。
 (可視化方法の詳細については、後述します)

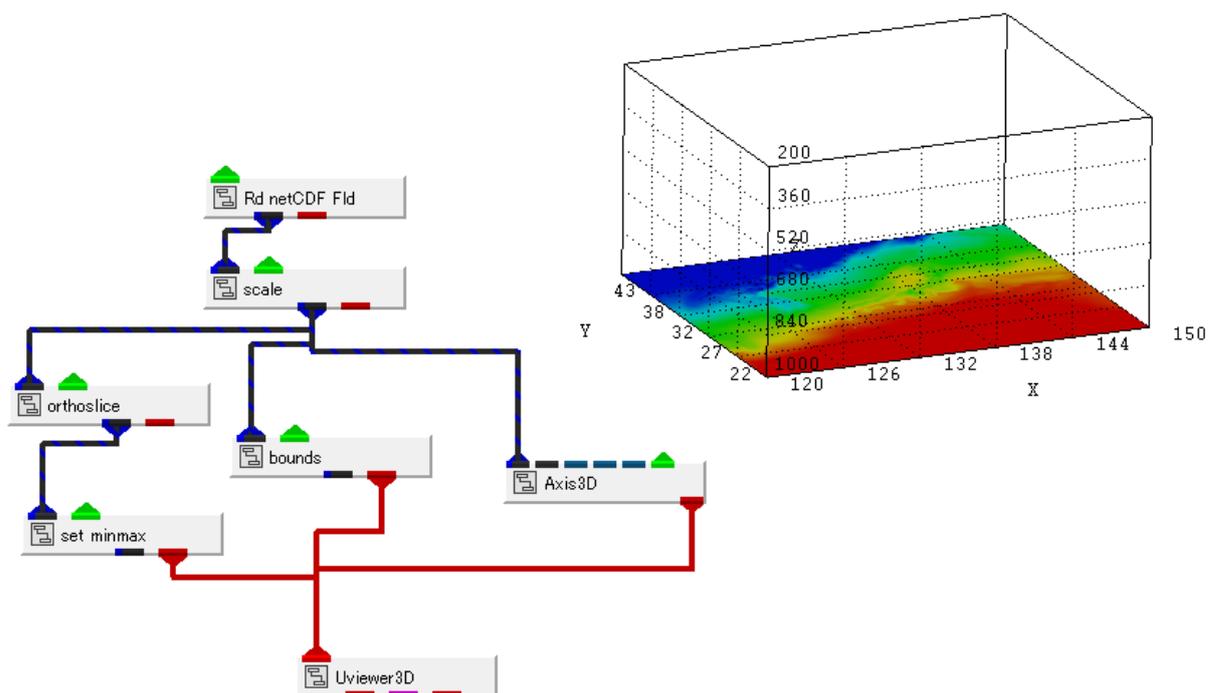


図 33 netCDF データの可視化例

図のように、経度 120 ~ 150 度の値と緯度 22 ~ 43 度の値が、XY 座標値に割り当てられます。
 Z 軸の p0 配列には、このデータでは、200 ~ 1000 (hPa) の値を持っています。
 可視化上、Z 軸方向を反転（且つ、縮小）して、表示しています。

逆に、このように配列の大きさを示す名前と同じ 1 次元データが定義されていない場合は、座標値は、直交等間隔として扱われます。
 その配列の大きさ、例えば、上記例では 360×720×5 の範囲である、{0, 359}, {0, 719}, {0, 4} が座標値に割り当てられます。

また、このモジュールには、オプションとして、このモジュールの出力（青ポート）のタイプをどのように扱うかを指定できるパラメーターがあります。



図 34 座標値の扱い

通常、Unspecified のまま、指定する必要はありません。
 座標値の 1 次元データが指定されている場合、Rectilinear 型で読み込まれます。

ここを Uniform 型に変更すると、座標の最大最小値のみを指定した直交等間隔のデータとして出力されます。

- Uniform : x/y/z 軸の範囲を定義
- Rectilinear : x/y/z 軸方向の軸座標を定義

※ 選択した後、データの再読み込みが必要です。

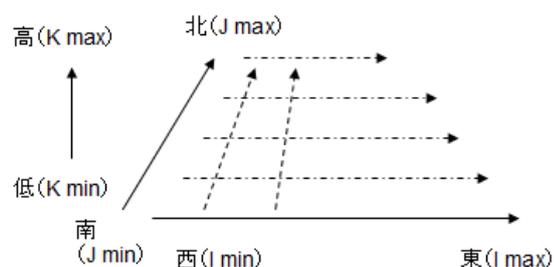
Structured はすべての座標値を指定するタイプですが、AVS/Express の Write_netCDF モジュールで作成された netCDF データの場合のみ、意味があります。

一般的な netCDF データでは、Rectilinear 型を Uniform 型として出力したい場合にのみ、利用します。

各データタイプの違いについては、AVS/Express のユーザズガイド (第 1 章) をご参照ください。

「データの並びについて」

AVS/Express は、以下の図に示すように、基本的には、経度方向 (西から東)、緯度方向 (南から北)、高さ方向 (地上から空)、に並んだデータに対応しています。



GRIB データのように、北から南に並んでいる場合、基本的な可視化は可能ですが、可視化機能によっては、動作しない機能があります。

(例えば、任意断面によるコンター図や流線表示など)

※ 次期バージョンでは、メッシュの並べ替えができる機能の追加も検討中です。

また、高さデータが気圧で出力されているケースもあります。

例えば、Z 座標値が 1000hpa, 800hpa, ... といった値になっている場合、読み込みは可能ですが、反転して表示されます。

基本的な可視化は可能ですが、表示上、scale で反転させてください。

また、この場合も任意断面など一部可視化機能は利用できません。

その他、単位の違いの自動変換などは行われません。例えば、速度成分と各軸方向の単位の違いに対する変換などは行われません。

2) WRF データの読み込み (netCDF)

WRF で計算された結果の netCDF データは、それを読み込む専用の Read_WRF モジュールを利用することができます。

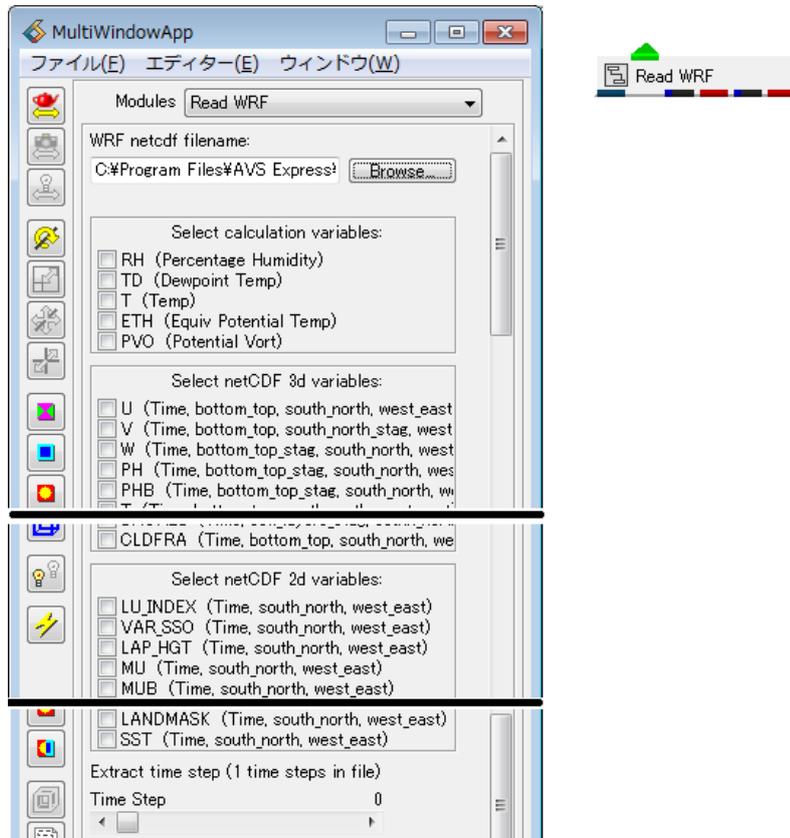


図 35 Read_WRF モジュール

Browse ボタンをクリックし、WRF で計算された結果の netCDF ファイルを指定します。先の Rd_netCDF_Fld モジュールと同じように、データファイルに含まれる変数名がパラメーターの欄にリストされます。

Rd_netCDF_Fld モジュールと異なり、以下の特徴があります。

- 3 種類のデータリスト

データを指定すると、上図に示すように、以下の 3 つの領域に分かれて変数のリストが表示されます。

Select calculation variable

Select netCDF 3d variable

Select netCDF 2d variable

まず、netCDF ファイルに含まれる変数を 2 次元データ、3 次元データに分類し、それぞれでリストします。

また、モジュール内で、解析結果を使って、以下の 5 つの変数の計算を行っています。これらの変数が一番上にリストされています。

RH (Percentage Humidity)
TD (Dewpoint Temp)
T (Temp)
ETH (Equiv Potential Temp)
PVO (Potential Vort)

- スタッガード成分の平均
風速の UVW のように、データがスタッガード格子で出力されているデータに対して、以下のように、その方向に対して、平均値を計算して出力しています。

$$U(i, j, k) = (U(i, j, k) + U(i+1, j, k)) / 2$$
$$V(i, j, k) = (V(i, j, k) + V(i, j+1, k)) / 2$$

- 座標値の扱い
X 座標と Y 座標は、XLONG と XLAT の値を割り当てています。
3次元データに対する Z 座標には、以下の計算結果を割り当てています。

$$Z = (PH + PHB) / 9.81$$

- 出力ポート
このモジュールは、Rd_netCDF_Fld モジュールと異なり、5つの出力ポートがあります。



図 36 Read_WRF モジュールの出力ポート

- 一番左は、時間データの文字列です。
Times データに出力されている文字列を読み込み、その文字列が出力されます。
左から 2 番目と 3 番目は、2D データの出力です。また、右の 2 つが 3D データです。

可視化したい変数をチェックしてください。
また、時系列データの場合は、Extract time step に、Time Step スライダーがあります。
このスライダーで時間ステップを指定してください。

3) GrADS データの読み込み

Fortran ダイレクトアクセス形式で出力された GrADS データを読み込みます。
例えば、以下のように出力されたデータがあるとします。

```
real*4 temp(100, 50, 10)
real*4 press(100, 50, 10)
open(77, file='out.dat', form='unformatted',
&      access='direct', recl=100*50*4)
irec=1
do k = 1, 10
  write(1,rec=irec) (( temp(i,j,k), i= 1, 100), j=1, 50)
  irec=irec+1
enddo
do k = 1, 10
  write(1,rec=irec) (( press(i,j,k), i= 1, 100), j=1, 50)
  irec=irec+1
enddo
```

経度方向に西から東へ、その次に緯度方向に南から北へ並んだメッシュデータです。
また、その後、高さ方向に並びます。

※ 上記例では、K 方向に別ループとしていますが、一括で出力していても
問題ありません。

このファイルを読み込むには、以下の Field アスキー・ヘッダー・ファイル (拡張子.fld) を
作成します。

```
----- この下から (ファイル名 out.fld) -----
# AVS
#
ndim = 3
dim1 = 100
dim2 = 50
dim3 = 10
nspace = 3
veclen = 2
label = temp  press
data = float
field = uniform

variable 1 file = ./out.dat  filetype = binary  skip = 0
variable 2 file = ./out.dat  filetype = binary  skip = 200000

coord 1 file = ./point.txt  filetype = ascii  skip = 0
coord 2 file = ./point.txt  filetype = ascii  skip = 1
```

```
coord 3 file = ./point.txt filetype = ascii skip = 2
----- この上まで -----
```

※ Field ヘッダーの詳細については、AVS/Express のユーザズガイド、第 1 章をあわせてご参照ください。

まず、半角 # (スペース) AVS (半角) で必ず始まる必要があります。
次に、`ndim = 3` (次元) であることと、`dim1/2/3` を使って、メッシュ数を指定します。
`veclen = 2` は、このファイルに含まれている変数の数です。
この例では、`temp` と `press` の 2 つがあります。
`label` はスペースで区切って、その変数名を指定しています。
(`label` はオプションです。ない場合は、内部で `data1,data2` が振られます。)
このデータが 4byte float で出力されているため、`data = float` です。

`variable` に、そのファイルへのパスとどのように読み込むかを指定します。
`veclen` の数分の `variable` 数値 の行を定義してください。
`binary` フォーマットであることと、`skip` で読み飛ばすバイト数を指定します。
`temp` データは、ファイルの先頭から入っているので、`skip = 0` です。
`press` データは、メッシュ数 $100 \times 50 \times 10$ に 4byte をかけた値 200000 byte を読み飛ばしたところから入っています。

※ エンディアンの違いについては、データの読み込み時に指定します。
`Read_Field` モジュールのパラメーターにあります (後述)。

次に `coord` 行です。
この `coord` 行には、XYZ 座標値の情報を指定します。
まず、`point.txt` というアスキーのテキストファイルを作成してください。
以下にサンプルを示します。

```
----- この下から (ファイル名 point.txt) -----
120.9144  149.1834
22.58825  48.66204
0.0      2.0
----- この上まで -----
```

このデータの経度の最小、最大値を指定します。
次に緯度の最小、最大値です。
3 行目は高さ方向の最小、最大値です。高さ方向は、可視化時にスケールをかけることもできます。まずは、適当な値でも結構です。

先の `coord` 行で、この経度 (X データ)、緯度 (Y データ)、高さ (Z データ) を読み込むように、指定しています。
直交等間隔 (`fld` ファイルのキーワード、`field = uniform`) のメッシュとして読み込みます。

なお、データによっては、高さ方向が等間隔のデータではないなど、不等間隔のメッシュで

扱いたい場合があるかもしれません。

その場合は、X 軸方向の格子数分 (dim1) の配列値、Y 軸方向の格子数分 (dim2) の配列値、Z 軸方向の格子数分 (dim3) の配列値を、point.txt 同様に、適当なアスキーファイルで作成してください。

例えば、以下のような 3 つのファイルを作成します。

```
-- この下から (ファイル名 x.txt)  --
120.9144
121.1999
121.4855
...
149.1834
-- この上まで (dim1 個分記述) ----

-- この下から (ファイル名 y.txt)  --
22.58825
23.12037
23.65249
...
48.66204
-- この上まで (dim2 個分記述) ----

-- この下から (ファイル名 z.txt)  --
1000
800
700
...
100
-- この上まで (dim3 個分記述) ----
```

fld ファイルの設定の以下の該当部分を書き換えます。

uniform ではなく rectilinear で定義します。

また、coord 1/2/3 で、各 XYZ 軸の値を記したファイルを指定します。

```
-- この下から (ファイル名 z.txt)  --
field = rectilinear

coord 1 file = ./x.txt  filetype = ascii
coord 2 file = ./y.txt  filetype = ascii
coord 3 file = ./z.txt  filetype = ascii
-- この上まで -----
```

※ xyz.txt のように 1 つのファイルに 3 カラムで記述し、offset と stride のキーワードを使って読み込むこともできます。
ユーザーズガイドの fld フォーマットに関する説明をご参照ください。

.fld ファイルが作成できたら、Read_Field モジュールでデータを読み込みます。

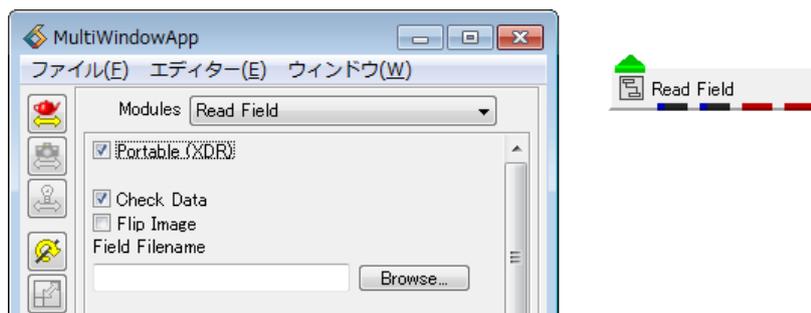


図 37 Read_Field モジュール

• Portable(XDR)

このモジュールには、Portable(XDR) というチェックボックスがあります。このチェックをオフにすると、以下のパラメーターが表示されます。

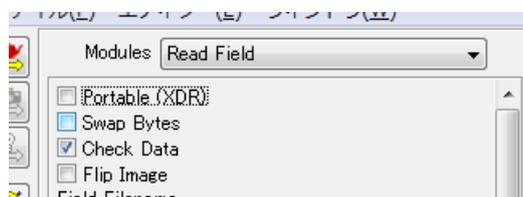


図 38 バイトスワップ

元のバイナリデータが、ビッグエンディアンかリトルエンディアンかによって、このチェックのオンオフを切り替えてください。

• リトルエンディアンの場合

Intel 系のチップ（一般的な Windows や Linux マシン）で作成されたリトルエンディアンのデータを読み込む場合には、

Portable(XDR) のチェック：オフ

Swap Bytes のチェック：オフ

にしてから、データファイルを指定してください。

(図 38 バイトスワップの状態です)

• ビッグエンディアンの場合

SPARC 系のチップのように、ビッグエンディアンのファイルを読み込むには、

Portable(XDR) のチェック：オン

のデフォルトのまま、もしくは、

Portable(XDR) のチェック：オフ

Swap Bytes のチェック：オン (バイトスワップさせる)

の設定にて、読み込みます。

Check Data は、読み込んだデータの値が極端に小さい（大きい）ときにアラートを出してくれるチェックで、そのまま構いません。

Browse ボタンをクリックし、先に作成した **out.fld** ファイルを指定します。

4) Vis5D データの読み込み

MM5 で計算された結果を **tovis5d** のツールを使って変換している場合など、**Vis5D** のデータがある場合には、**Read_Vis5D** モジュールを使って読み込むことができます。

注意：

Read_Vis5D モジュールは、以前のバージョンでは標準サポートされていましたが、諸所の事情により、現バージョンでは、製品に含まれなくなりました。以下のサイトで **Read_Vis5D** のモジュール（**AVS/Express 8.1** 用）を公開していますので、ダウンロードして適用してください。

<http://www.cybernet.co.jp/avs/support/avsexpress.html>

8.1 用の **Read_Vis5D** モジュールを適用すると、以下の **VIS5D** ライブラリページが追加されます。

Read_Vis5D モジュールと **Read_Vis5D_Topography** モジュールを使います。

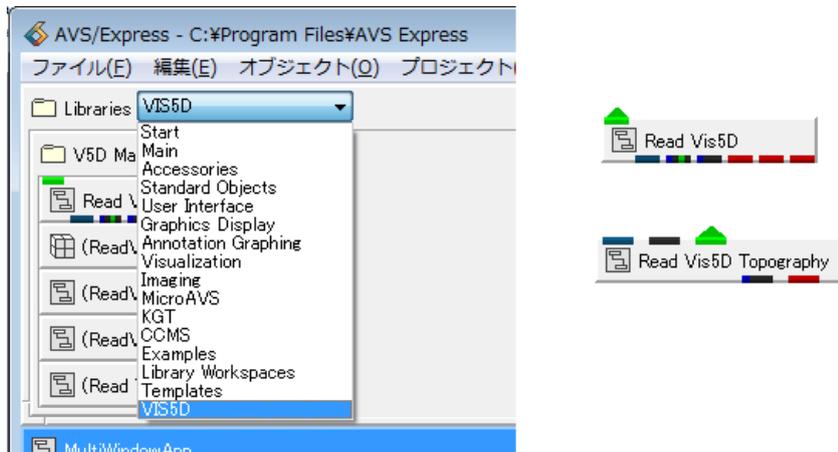


図 39 Read_Vis5D モジュール

この **Read_Vis5D** モジュールをダウンロードしたフォルダの下に **v** フォルダに **sample_1.v** や **sample_2.v** があります。

これらのサンプルを読み込んでみてください。

3.2 スケール、境界表示、軸表示

この節から、可視化に利用するモジュールと、その使い方について説明します。

まず、データを読み込んだ直後に利用する機会の多い、以下のモジュールについて説明します。

- `scale`
- `bounds`
- `Axis3D`

• `scale` モジュール

`scale` モジュールは、XYZ 方向のスケールを変化させることができるモジュールで、例えば、高さの値が 0 ~ 10000 m のようにメートルで与えられている場合、そのまま表示すると、経度緯度方向に比べて、高さ方向に極端に長くなってしまいます。z 方向のスケールを 0.01 と縮小することができます。その他、1000 ~ 200 hPa のように気圧で与えられている場合に、上下反転させる際にも利用できます。

※ ただし、風速成分（のベクトル図など）の可視化を行う際には、注意が必要です。高さ方向の成分にもスケールが必要となります。その際には、`data_math` モジュールや `coordinate_math` モジュールを使って、値そのものを修正する方法をご検討ください。

• `bounds` モジュール

`bounds` モジュールは、そのデータのメッシュの外形線（境界線）を表示するモジュールです。特にデータのメッシュ形状が正しく読み込めているかどうかを確認できます。

次の図では、`bounds` モジュールのパラメーターで `Edges` にチェックし、`Kmax` にチェックした表示を示しています。

また、先の `scale` モジュールも利用しています。

紙面では真っ白につぶれていますが、経度緯度面（高さが最大の部分）のメッシュの状態を見ることができます。

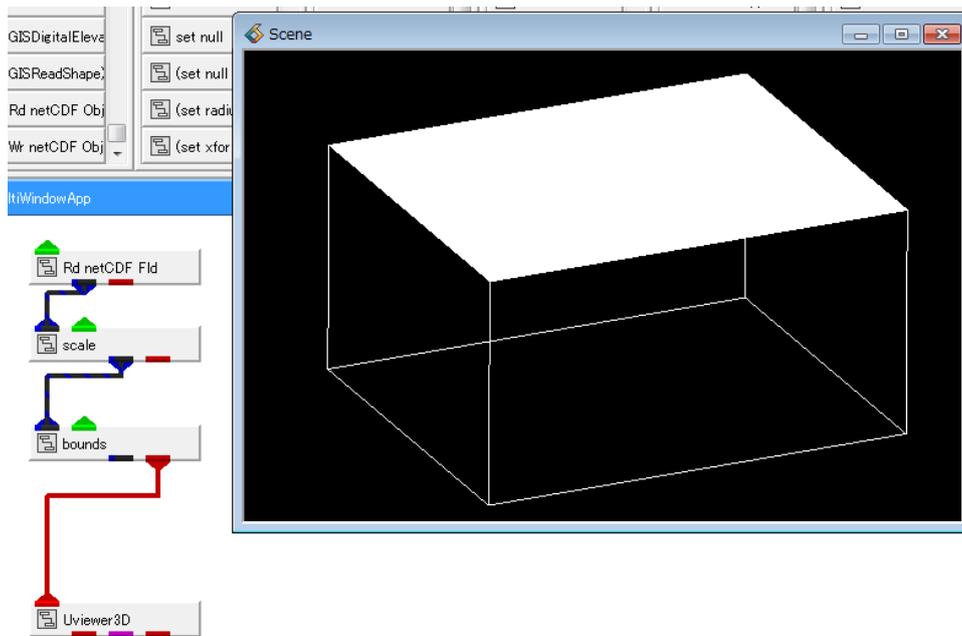


図 40 scale モジュールと bounds 表示

・ Axis3D モジュール

Axis3D モジュールは、その表示範囲に数値軸を表示するモジュールです。
以下のように接続します。

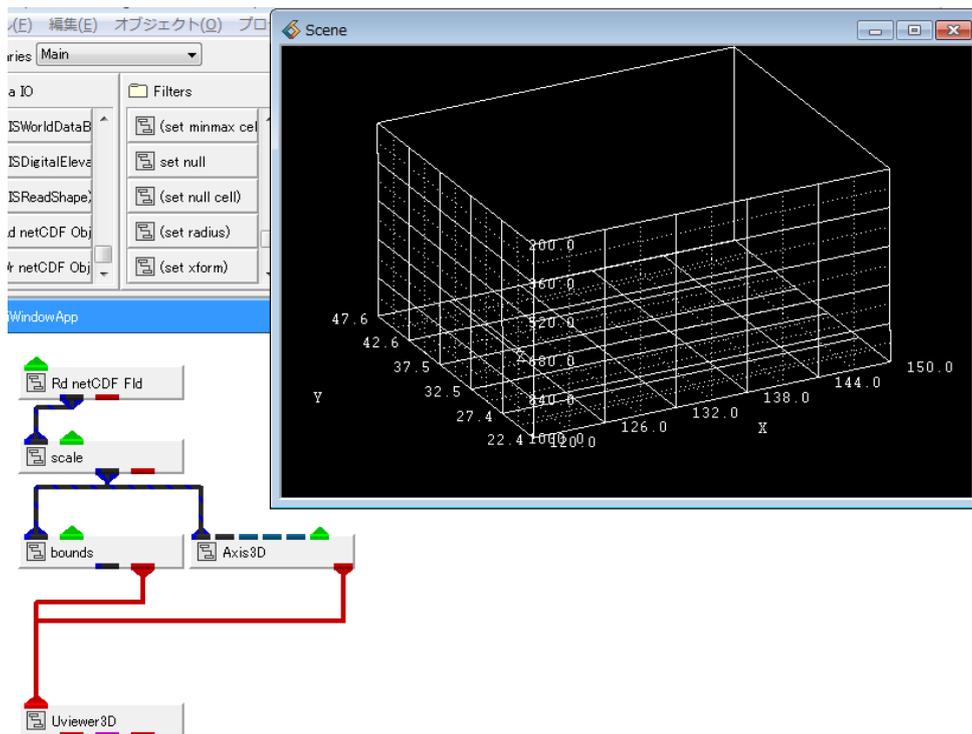


図 41 Axis3D モジュール

図の例では、scale モジュールで、Z 方向のスケールを -0.02 に指定し、反転させています。
Axis3D モジュールのパラメータに、ラベルの表示位置のオフセットなどがありますので、
適宜、変更してみてください。

3.3 等値面

等値面は、3次元空間にどのようにデータが分布しているかを見ることができる可視化機能です。等値面を作成するには、以下のモジュールを利用します。

- isosurface
- isosurface_nest
- (• extract_scalar)

• isosurface モジュール

isosurface モジュールは、スカラー値に対して、あるレベル値が分布する場所に、面を生成するモジュールです。

等高線の3次元版となります。

ネットワークを以下に示します。

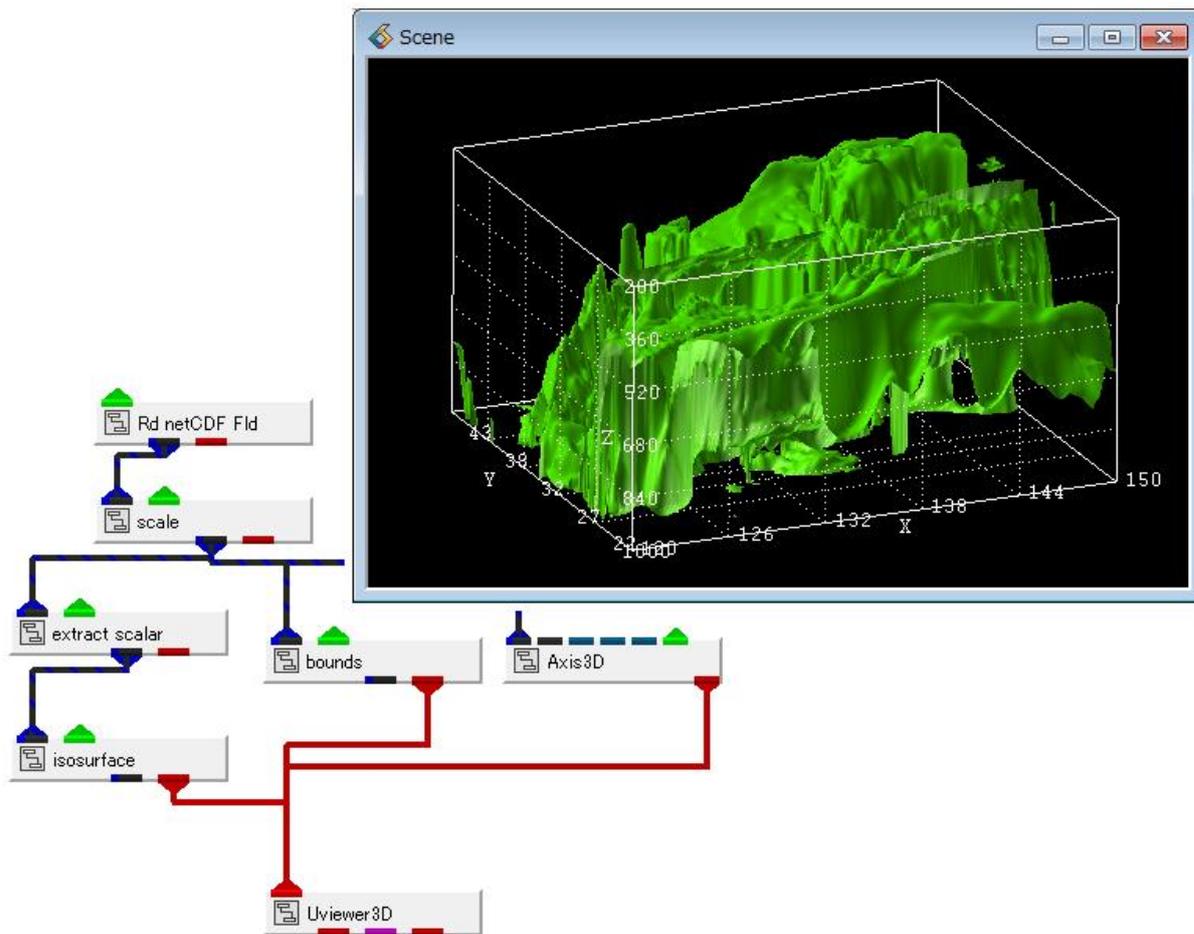


図 42 等値面表示

図の例では、isosurface モジュールの前に、extract_scalar モジュールを接続しています。複数のデータ成分がある場合、isosurface モジュールの中で選択することもできますし、あるデータ成分を明示的に選択してから、等値面モジュールを使うこともできます。

以下の図は、等値面モジュールのパラメーターです。

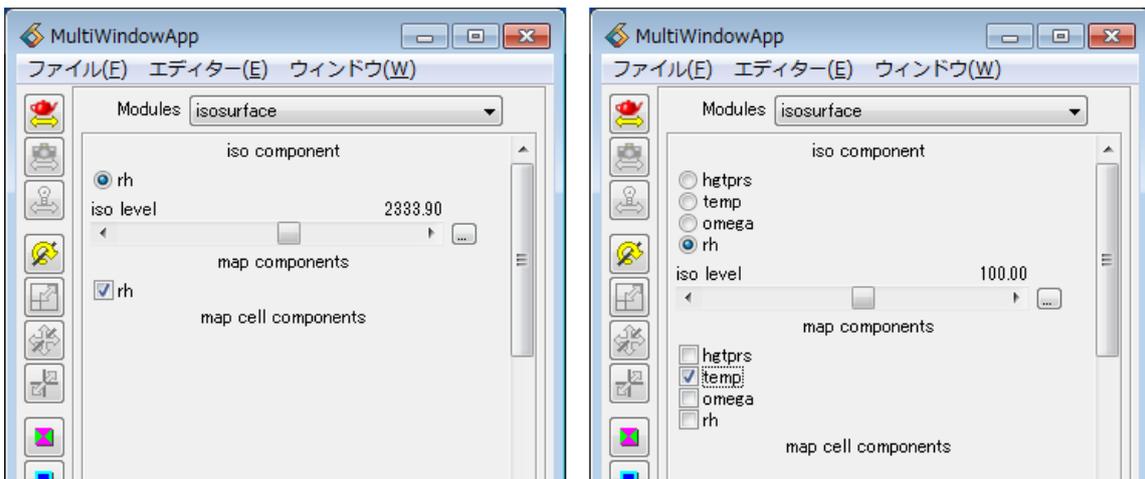


図 43 isosurface モジュールのパラメーター

左図は、extract_scalar モジュールの後に isosurface モジュールを接続した場合の例です。右図は、extract_scalar モジュールを使わずに、直接接続した場合の例です。

extract_scalar モジュールを使わない場合、右図のように、対象成分を選択できるように、レベル値のスライダーの上に、成分名がリストされます。

また、スライダーの下にある map components は、その等値面を色づけするための成分の選択があります。

左図では、等値面とその色は、同じ成分となっています。

その成分の最小最大値間を青から赤で色付けするため、下図に示すように、レベル値が低い部分では青、レベル値を高くすると赤と変化します。

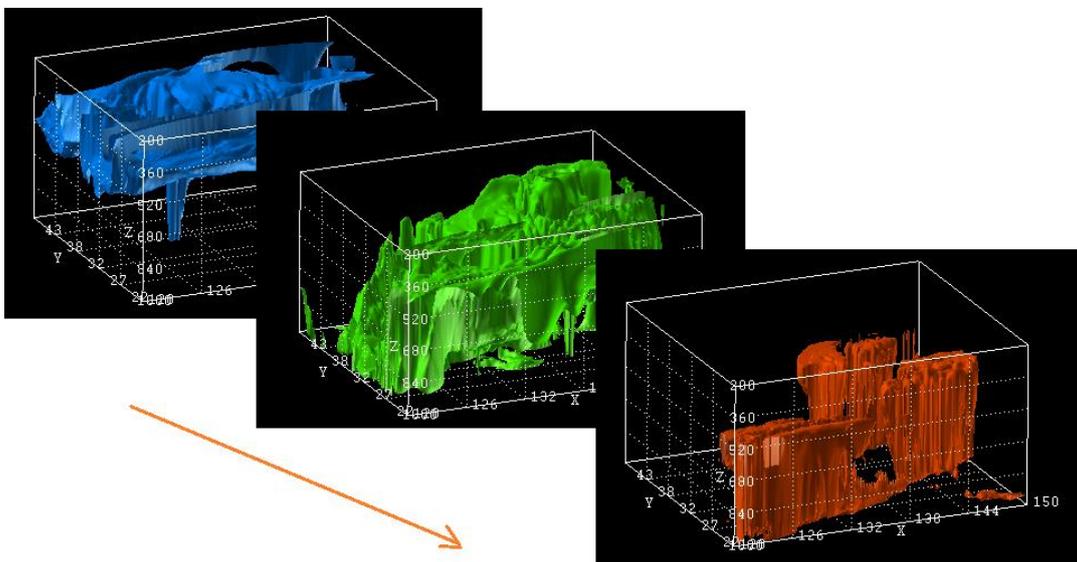


図 44 等値面のレベル値と色

一方、図 43 isosurface モジュールのパラメーターの右図の場合、別の成分を選択できます。

ある成分で等値面を作成し、その等値面上に分布している別の成分で色づけできます。

下図は、データ成分 rh で等値面を作成し、その等値面を temp で色づけした例です。

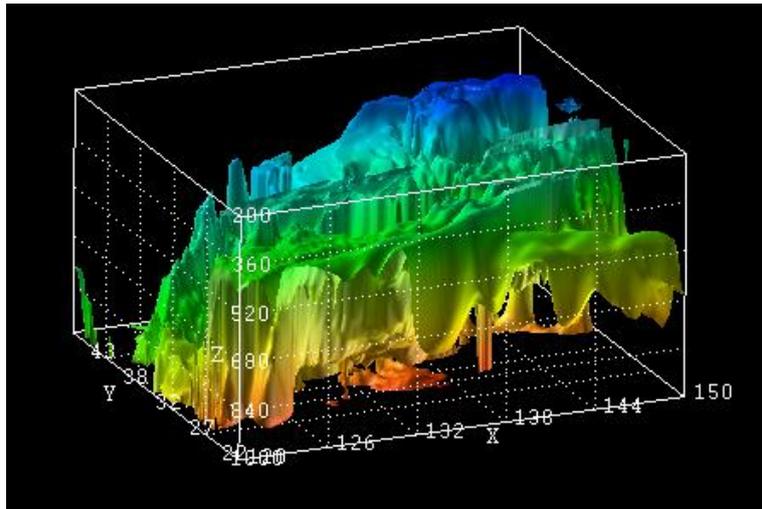


図 45 等値面の別成分による色づけ

色と数値の対応については、`set_minmax` モジュールを使います。

このモジュールについては、後述のコンター図に関する説明「3.5 断面コンター図、等圧線など」で紹介します。

なお、いずれも、この `map components` にある成分の選択をオフにすると、単色で色づけできます。

自分で適当な色を付けたい場合には、この成分の選択（チェック）を外してください。白一色になります。

※ オブジェクトに対する色づけや半透明など、各種属性の変更方法については、「色、その他の属性の変更第4章 色、その他の属性の変更」で説明します。

また、複数枚の等値面を作成したい場合は、`isosurface` モジュールを複数利用してください。もしくは、次に示す `isosurface_nest` モジュールもあります。

・ isosurface_nest モジュール

isosurface_nest モジュールは、複数枚の等値面を作成することができるモジュールです。

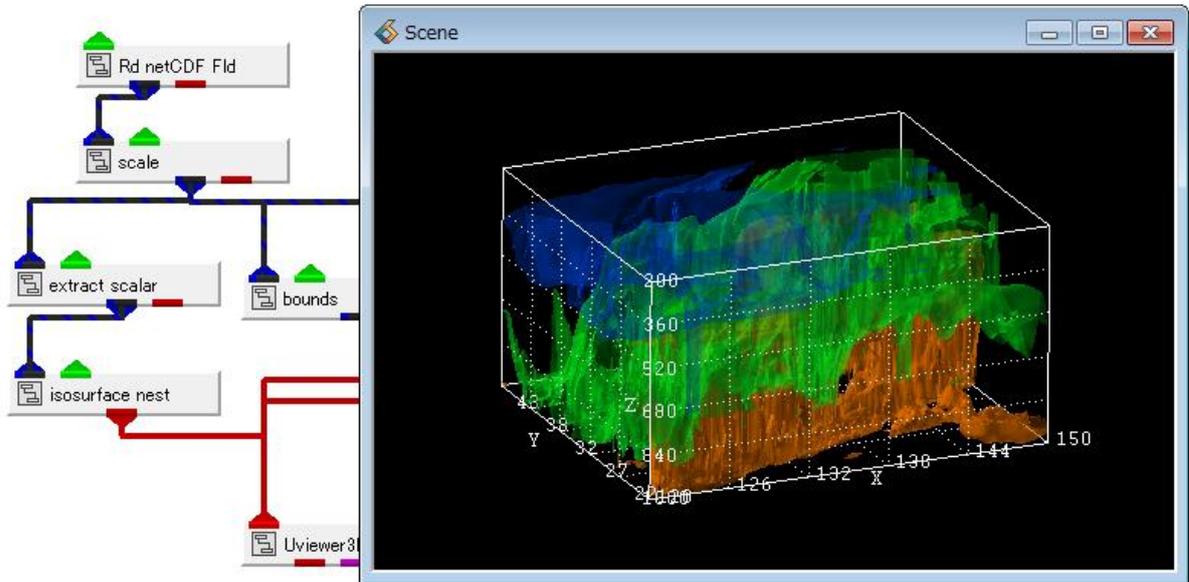


図 46 isosurface_nest モジュール

以下のパラメーターがあります。

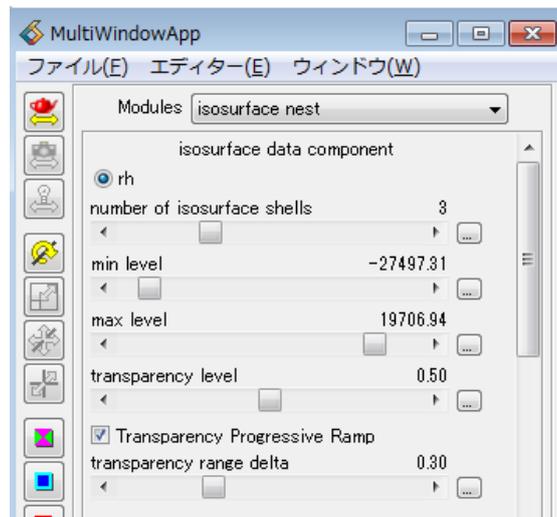


図 47 isosurface_nest モジュールのパラメーター

number of isosurface shells のスライダーにて、何枚の等値面を作成するかを指定します。min/max level で指定した値の間を 3 等分 (3 の場合) した位置に、等値面を作成します。色は、この成分で色づけされますので、図 44 等値面のレベル値と色 に示した絵が同時表示されている絵となります。

また、このモジュールでは、半透明表示を行っています。transparency level で、全体の半透明の度合いを指定します。次の Transparency Progressive Ramp にチェックすると、その 3 枚の等値面の透明度を同じ透明度ではなく、少しずつ変化させた値で表現できます。

その下の `transparency range delta = 0` の場合、3枚は同じ透明度で表示されます。
1.0 近づけると、最小レベルの等値面がより透明に、最大レベルの等値面が不透明に変化します。

- ※ 複数枚の半透明が重なった場合、正確な前後処理は行われていません。
透明度を適宜調整してみてください。
また、「4.3 その他の属性」にあるソートについてもご参照ください。

3.4 ボリュームレンダリング

ボリュームレンダリングは、等値面同様、3次元空間にどのようにデータが分布しているかを見ることができる可視化機能です。

ボリュームレンダリングを行うには、以下のモジュールを利用します。

- volume_render
(• volume_render_simple)
(• make_volume)

• make_volume モジュール

まず、AVS/Express のボリュームレンダリング機能は、直交等間隔の byte データ、もしくは、short データのみに対応しています。

そのため、数値解析の結果に対して、そのまま、ボリュームレンダリングを行うことはできません。

そこで、make_volume モジュールをサポートしています。

このモジュールは、元データから補間を行い、ボリュームレンダリングができるデータへデータ変換するモジュールです。

• volume_render モジュール

ボリュームレンダリングを行うモジュールは、volume_render モジュール、もしくは、volume_render_simple モジュールです。

この2つの違いは、カラーマップのレンジの指定のパラメーターの有無です。

以下の例では、volume_render モジュールを利用しています。

以下に、可視化のネットワークを示します。

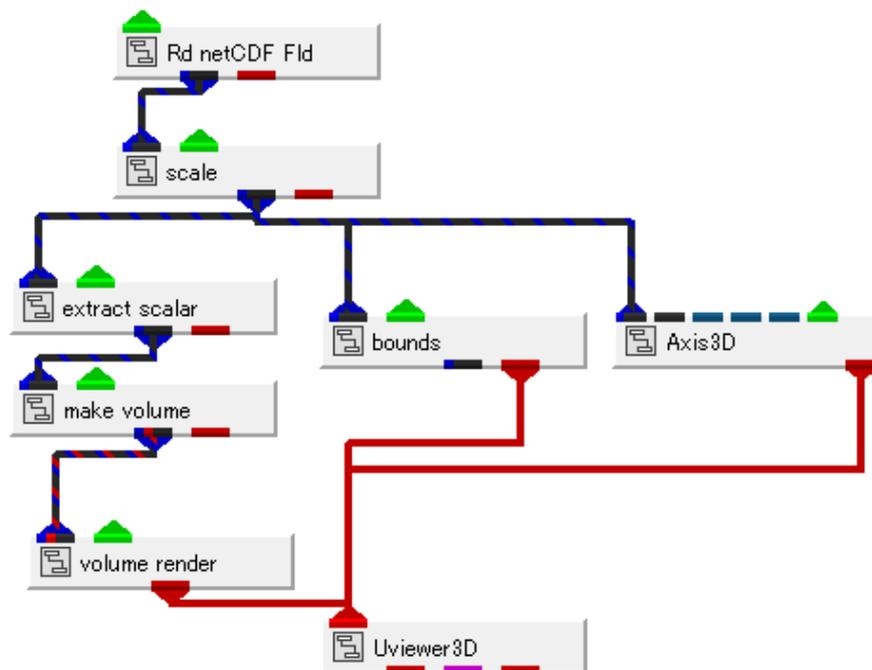


図 48 ボリュームレンダリングのネットワーク

次の図は、make_volume モジュールのパラメーターです。

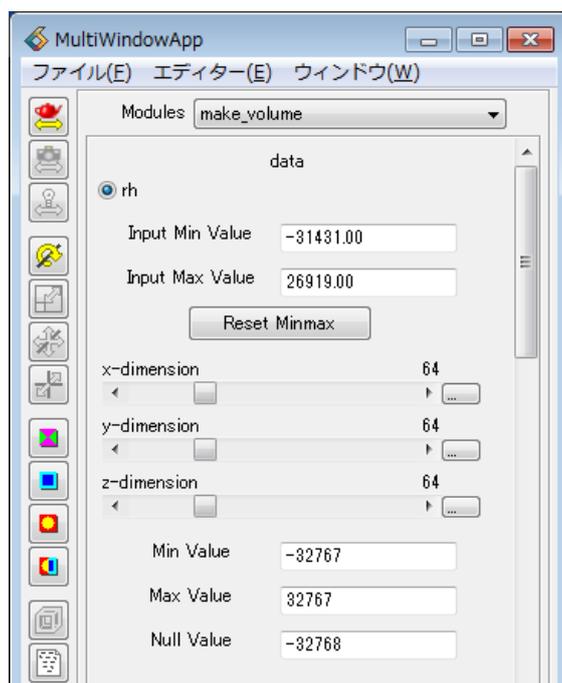


図 49 make_volume モジュールのパラメーター

※ AVS/Express 8.1 の make_volume モジュールには一部不具合がありました。パッチを公開していますので、パッチを適用してください。ユーザーインターフェースの構成が上図と異なる場合、そのパッチが適用されていません。

(Input Min/Max Value や Reset Minmax ボタンがあるかないかをご確認ください)

make_volume モジュールは、先に述べたように、直交等間隔のメッシュにデータの補間を行うモジュールです。x/y/z-dimension に指定したメッシュ数に、データを再構成します。

また、データの値も short 型に変換します。図に示すパラメーターの上段に、そのデータの最大最小値が表示されています。この値を、下段にある Min/Max Value のデータ範囲に変換します。

もし、データが時系列の場合は、この上段のパラメーターを固定するために、適当な範囲の値を設定してください。時系列の変化、全体を通して、その時系列全体の値が収まるようなデータの範囲や、ある着目したいデータの範囲を指定します。

次の図は、volume_render モジュールのパラメーターと、その結果の画像です。

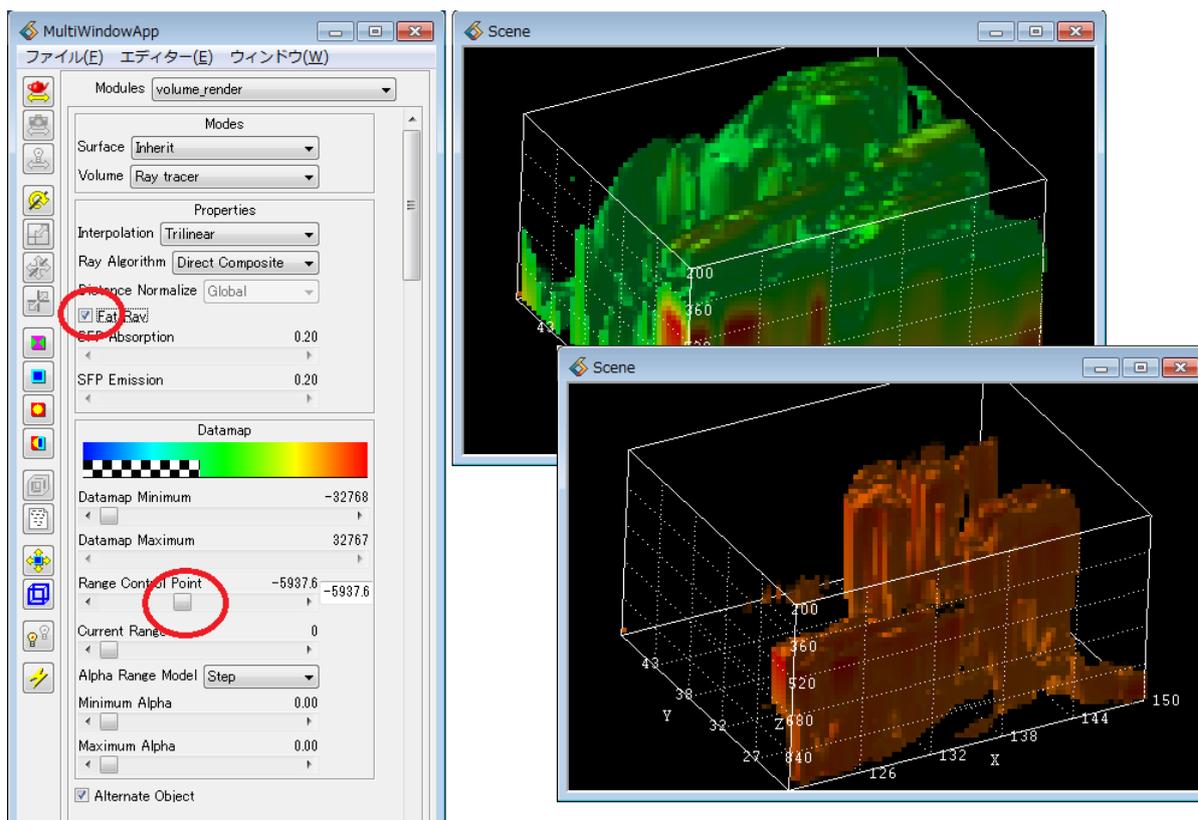


図 50 volume_render モジュールのパラメーターと表示結果

まず、図の下側にある Range Control Point を移動してみてください。先の等値面と同様、数値のレンジによって値の分布が変わる様子を見ることができます。

※ make_volume モジュールで、元のデータ値を short 型に変換している点にご注意ください。

また、上側にある Fat Ray のチェックを外すと、滑らかな表示に変わります。オンの場合は 4x4 のブロックに対してひとつの値を求めます。そのため、上図のように粗い表示となりますが、処理が速いため、パラメーターを決定するまでは、このモードがオンの状態で作業を行うと便利です。パラメーターを決定したら、このチェックを外してみてください。

このモジュールのパラメーターは、2つのカラーレンジを持っています。

Range Control Point で、まず閾値を決めます。

次に Current Range を 0 か 1 に移動し、それぞれの min/max の透明度を決定します。

次の図をご参照ください。

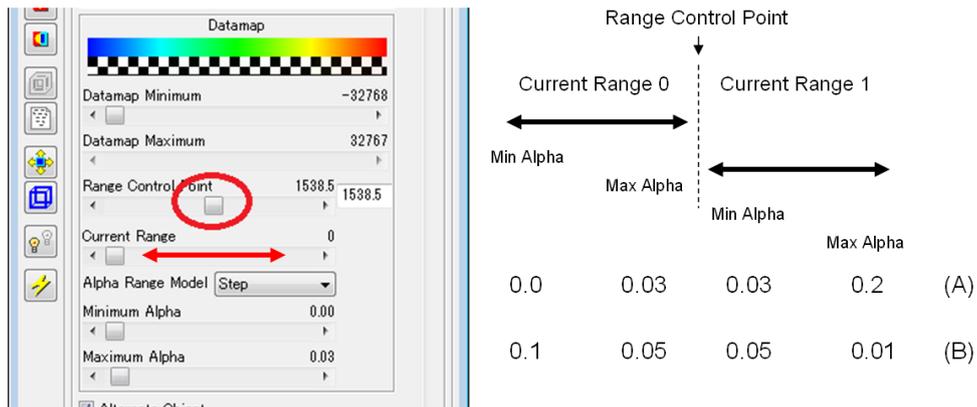


図 51 volume_render モジュールのカラーレンジ設定

Current Range 0 は、最小値とその閾値の透明度（アルファ値）を設定します。右側の Current Range 1 で閾値と最大値の透明度を設定します。

例えば、(A) は、最小値を透明（表示しない）に設定し、値が大きくなるにつれ、不透明に設定しています。(B) は、逆のパターンです。最小値の方が不透明となります。この2つの結果の画像の例を以下に示します。

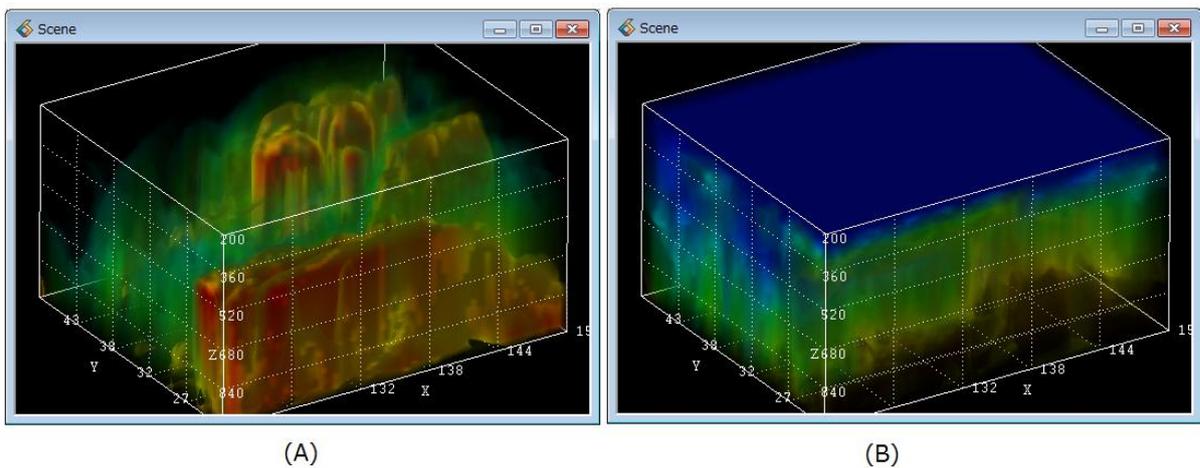


図 52 透明度（アルファ値）の設定

※ カラーマップの変更方法については、「4.1 カラーマップの色の調整」にて説明します。

注) ボリュームレンダリングは、表示画面（ビュー）の大きさ（解像度）によって、処理時間が変わります。

ビューが大きいかほど、処理の時間がかかります。

また、画面に更新がかかるとその都度、計算が行われます。

パラメーターの設定などが終わるまでは、Fat Ray をオンにする、

また、ボリュームレンダリング以外を操作する場合は、volume_render から Uviewer への接続（赤ポート）を一旦切断しておくなどして作業してください。

`volume_render` モジュールのパラメーターは、先に述べたように、ある閾値とその前後の2つのレンジの最大最小値のアルファ値を設定するため、線形に2段階で変化する透明度しか設定できません。

もう少し、任意のカーブの透明度を設定したい場合には、以下のモジュールを利用してください。

- `volume_render_simple`
- `ColormapEditor`

• `volume_render_simple` モジュール

このモジュールは、`volume_render` モジュールと同じように接続して利用します。ただし、`volume_render` モジュールにあった、閾値やカラーの設定のパラメーターを持っていません。

• `ColormapEditor` モジュール

このモジュールは、「2.3.2 `ColormapEditor` による色づけの変更」の地図の色変更について、一度説明しました。

このモジュールを使って、ボリュームレンダリングの透明度を設定します。

以下の図のように、この2つのモジュールを接続します。

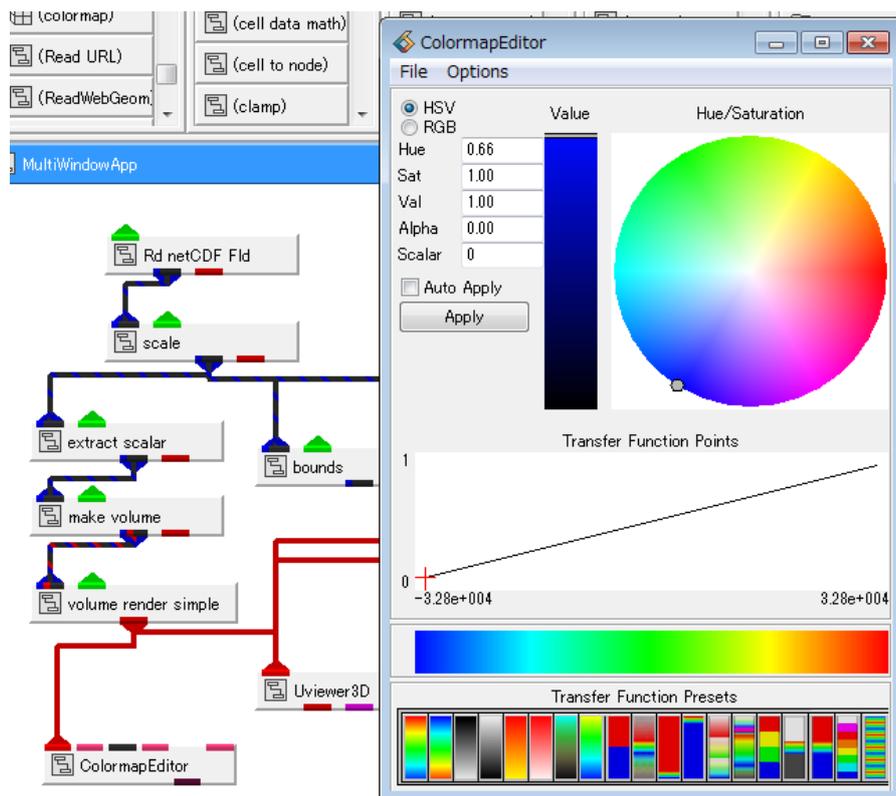


図 53 `volume_render_simple` モジュール

`ColormapEditor` のパネルの中央に斜めのラインがあります。

この部分が、透明度を示しています。デフォルトでは、最小値が 0.0 で、最大値が 1.0 になっています。

以下のように操作できます。
 まず、ライン上でマウス右クリックすると、制御点ができます。
 その制御点を、今度は、マウス左で押したまま、掴んで移動します。

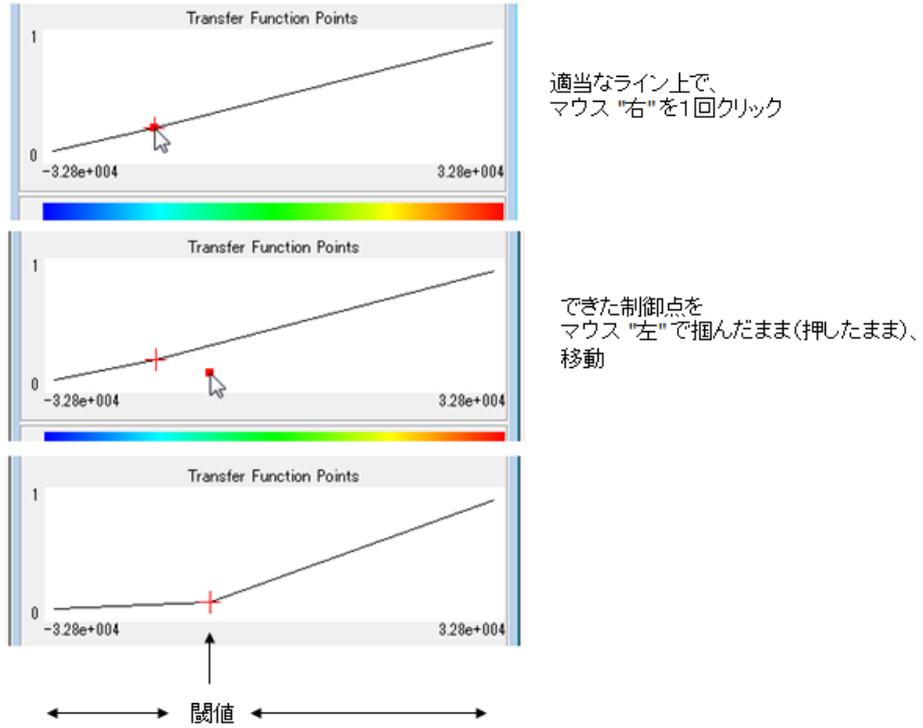


図 54 ColormapEditor の透明度 (アルファ値) の設定

上図のように、1点、制御点を作成すると、先の volume_render モジュールのパラメーターと同じ状態となります。ある閾値に対して、左側と右側の2つのレンジに対して、その最大最小値のアルファ値を設定していることになります。

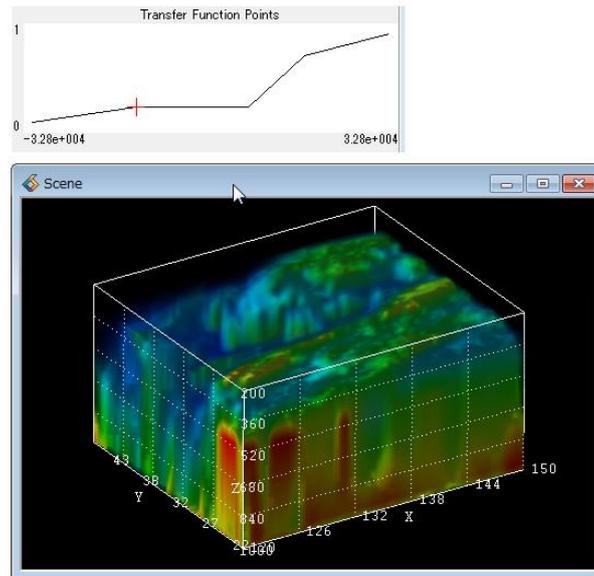


図 55 アルファ値の設定

制御点を複数作り、ある値まで線形に透明度を変化させ、その後、平坦に、また、ある値から透明度を変化させるといった、複合的な設定を行うこともできます。

また、複数の制御点で、なるべく線形よりも滑らかに曲線的に変化させることもできます。

地図データの可視化に関する「2.3.2 ColormapEditor による色づけの変更」でも述べたように、実際に、この ColormapEditor モジュールを利用する場合は、カラーのテーブルをファイルで作成するようにしてください。

注) 特にボリュームレンダリングの場合は、アプリケーションを保存、再生する際に、保存時の状態を再現できないという不具合があります。

修正については調査中ですが、現バージョンでは、必ず、次に説明する `cmp` ファイルを保存しておくようにしてください。

下図のメニューから `cmp` ファイルに保存します。

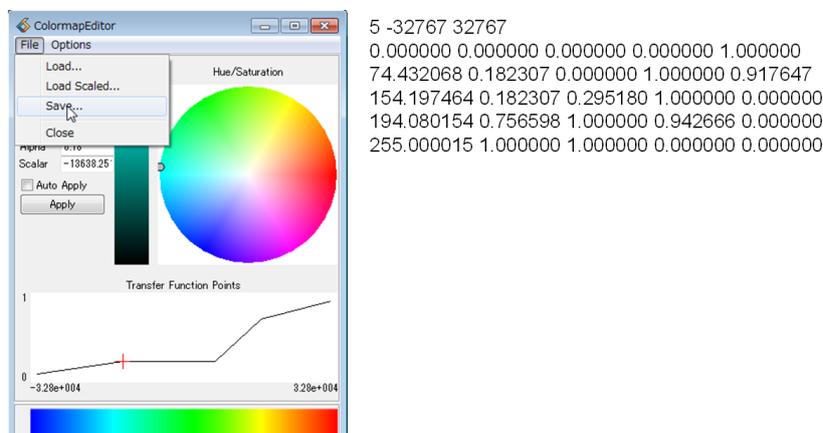


図 56 `cmp` ファイル

制御点の数(最小、最大も含め)、値の最小最大値 (`make_volume` モジュールで変換された `short` 型の値) が 1 行目に並んでいます。

その下には、制御点の数分の行が並びます。

左から、制御点の位置 (0 ~ 255 の範囲で指定します)、アルファ値、R 値、G 値、B 値 (いずれも 0 ~ 1 の範囲で指定) を指定します。

このファイルを書き換え、ご自身のデータ用のカラーマップファイルを作ってみてください。

作成できたら、ColormapEditor モジュールの `File` の `Load` メニューで読み込み、`Apply` ボタンを押して、反映させます。

注) 先の再現同様、この再生についても、すぐに反映できない場合がある不具合が報告されています。

その際は、一度、`volume_render` モジュールを削除し、再度インスタンスする、または、一度、ColormapEditor のアルファ値を操作、`Apply` してから、再度、保存している `cmp` ファイルを読み込むなど、行ってみてください。

(この不具合についても調査中です)

3.5 断面コンター図、等圧線など

その他のスカラー値の可視化方法として、ある断面でのコンター図（塗りつぶし）や、そのラインコンター図（等圧線、等高線）があります。

以下のモジュールを利用します。

まず、断面の抽出モジュールです。

- orthoslice
- slice_orthoplane
- slice_plane
- slice_arbitplane

orthoslice モジュールは、メッシュの格子断面（IJK）を抽出するモジュールです。計算メッシュ上の値を見ることができます。

一方、slice_orthoplane モジュールは、XYZ 断面を抽出するモジュールです。XYZ 座標空間の任意の直交断面を抜き出すことができます。

その他、任意断面を抽出できる、slice_plane モジュールや slice_arbitplane モジュールもあります。

以下のネットワークと可視化例をご覧ください。「3.3 等値面」の isosurface モジュールの代わりに、orthoslice モジュールを接続しています。

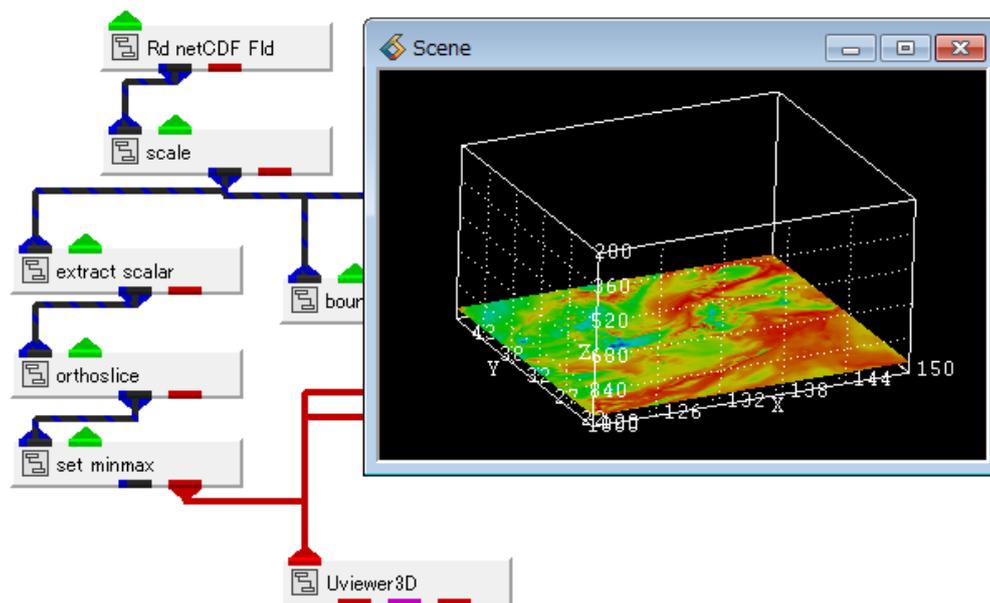


図 57 メッシュ断面によるコンター表示

また、以下の図は、orthoslice モジュールのパラメーターです。

axis で IJK の軸方向を、plane にて、そのメッシュの位置を指定します。

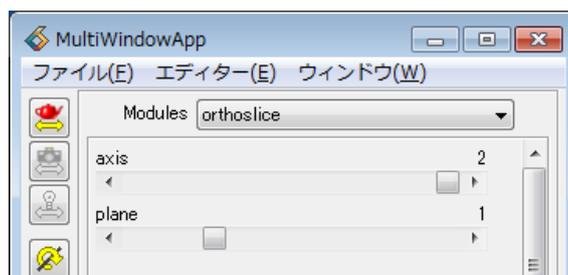


図 58 orthoslice モジュールのパラメーター

代わりに、`slice_orthoplane` モジュールを接続すれば、XYZ のある位置の直交断面を指定できます。この断面の抽出モジュールの出力である赤ポートをビューワーに接続すると、塗りつぶしのコンター図として表示できます。

上図の例では、`set_minmax` モジュールをビューワーとの間に挿入しています。

このモジュールは、カラーと数値の対応を決めるのに便利なモジュールです。`set_minmax` モジュールのパラメーターを下図に示します。

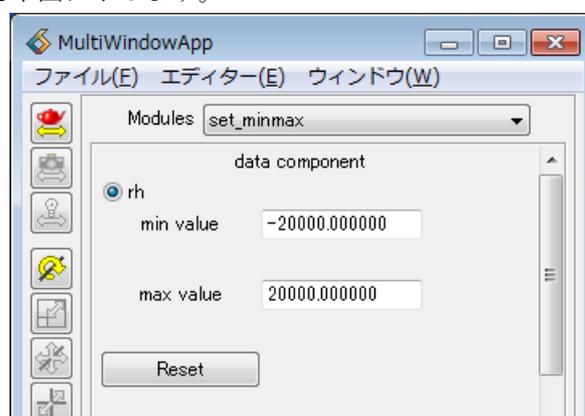


図 59 set_minmax モジュールのパラメーター

データの読み込み直後は、この `min/max value` に、その読み込んだ（このケースでは、その断面内の）データの最大最小値が表示されます。`Reset` ボタンを押すと、その状態を再確認できます。

この `min/max value` を設定することで、カラーマップの青から赤に割り当てるデータの大きさを指定できます。ちなみに、下図のように `LegendHoriz` モジュール（もしくは `LegendVert` モジュール）を接続すると、その数値のレジェンド（凡例）を表示することもできます。

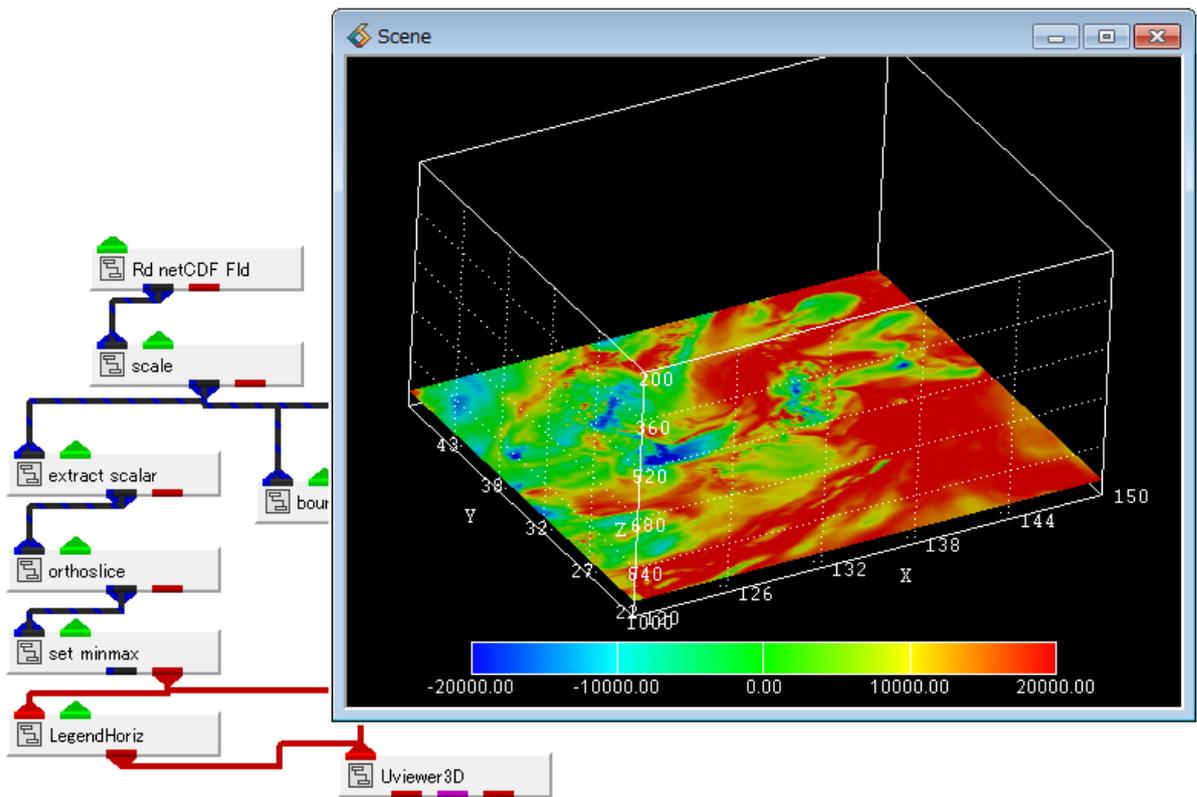


図 60 色の範囲指定とレジェンド表示

断面を抽出した後に、さらに、コンター図を作成するモジュールを接続できます。

- contour
- isoline
- solid_contour

contour モジュールは、塗りつぶしのコンター図に対して、ある値より小さい、また、大きい値を非表示にする機能を持っています。

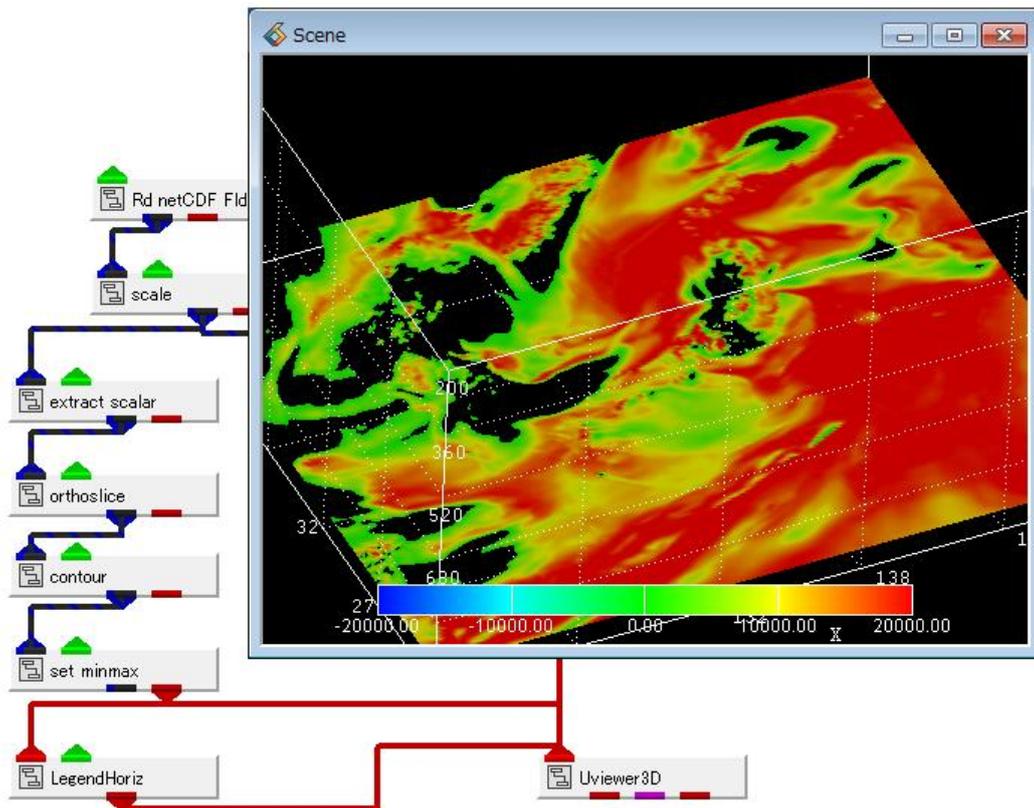


図 61 contour モジュールによる表示範囲の指定

下図は、contour モジュールのパラメーターです。

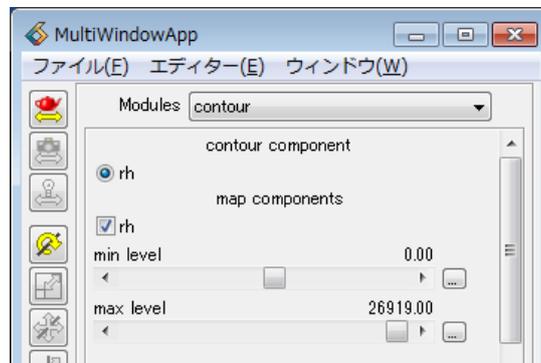


図 62 contour モジュールのパラメーター

min level を 0.0 に設定しています。値が 0.0 より小さい値は、非表示となります。よって、画面上では、0 ~ 26929（最大値）の範囲のコンター図のみが描かれています。

上図の可視化例では、set_minmax モジュールのパラメーターは、-20000 ~ 20000 のままになっていますので、色の分布は先の例と変わりません。set_minmax モジュールのパラメーターの設定を 0 ~ 20000 に設定すると、0 が青になるため、青から赤の色の分布に変わります。

isoline モジュールは、ラインコンター図を描くモジュールです。可視化の対象が圧力であれば等圧線になります。温度であれば等温線です。

contour モジュールの代わりに、isoline モジュールを接続してみてください。

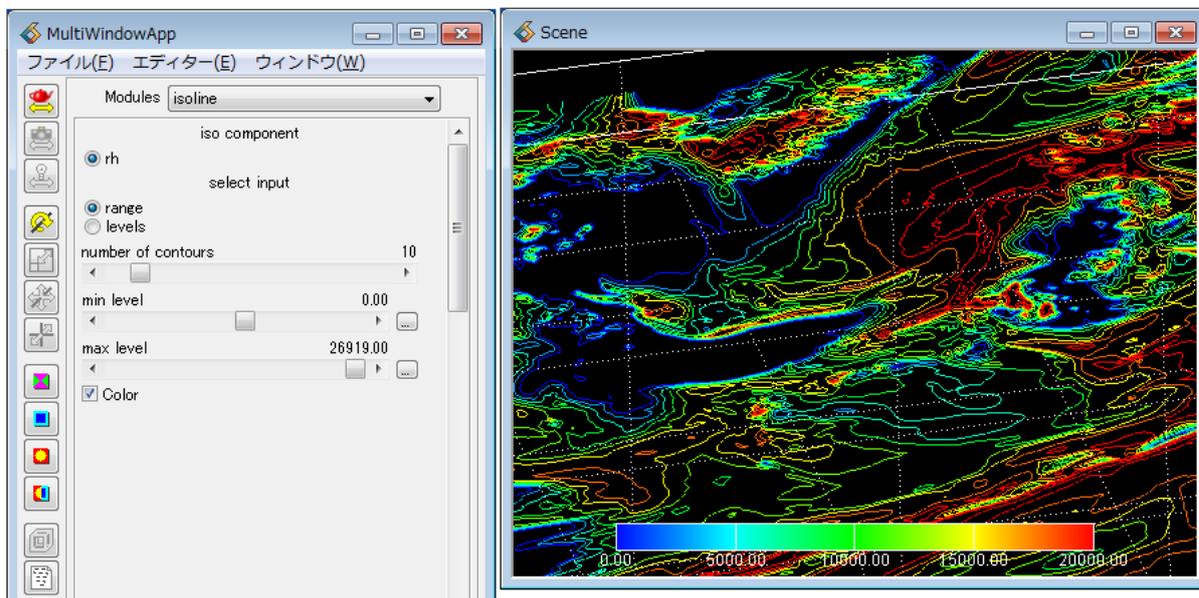


図 63 isoline モジュールによるラインコンター図

このモジュールには、範囲と何本のラインを生成するかを指定するパラメーターがあります。

また、range と levels の切り替えを levels にすると、直接、生成したいレベル値を、数値で複数指定することもできます。0, 1000, 4000, 8000, 20000 のように、カンマ区切りで指定します。

その他、solid_contour モジュールもあります。このモジュールは、「図 57 メッシュ断面によるコンター表示」や「図 61 contour モジュールによる表示範囲の指定」のように、滑らかな色の分布ではなく、色諧調を指定したコンター図を作成します。

また、このモジュールには、コンターラインのオンオフがありますので、その諧調間にラインをつけることもできます（次図「」参照）。

solid_contour モジュールには、先の contour モジュール同様、表示範囲を指定できる min/max のレンジがあります。ここで、表示範囲を決めることができます。色の諧調は、number of contours で決めます。ここで指定した諧調のコンター図となります。

なお、モジュールの接続方法として、表示範囲の指定と色の対応を指定したい場合には、set_minmax モジュールを“上流”に接続する点にご注意ください。まず、上流にて、色と数値の最大最小の対応を決めます。この範囲が solid_contour モジュールの表示範囲の最大最小になります。さらに、この solid_contour モジュールで表示範囲を絞ることができます。

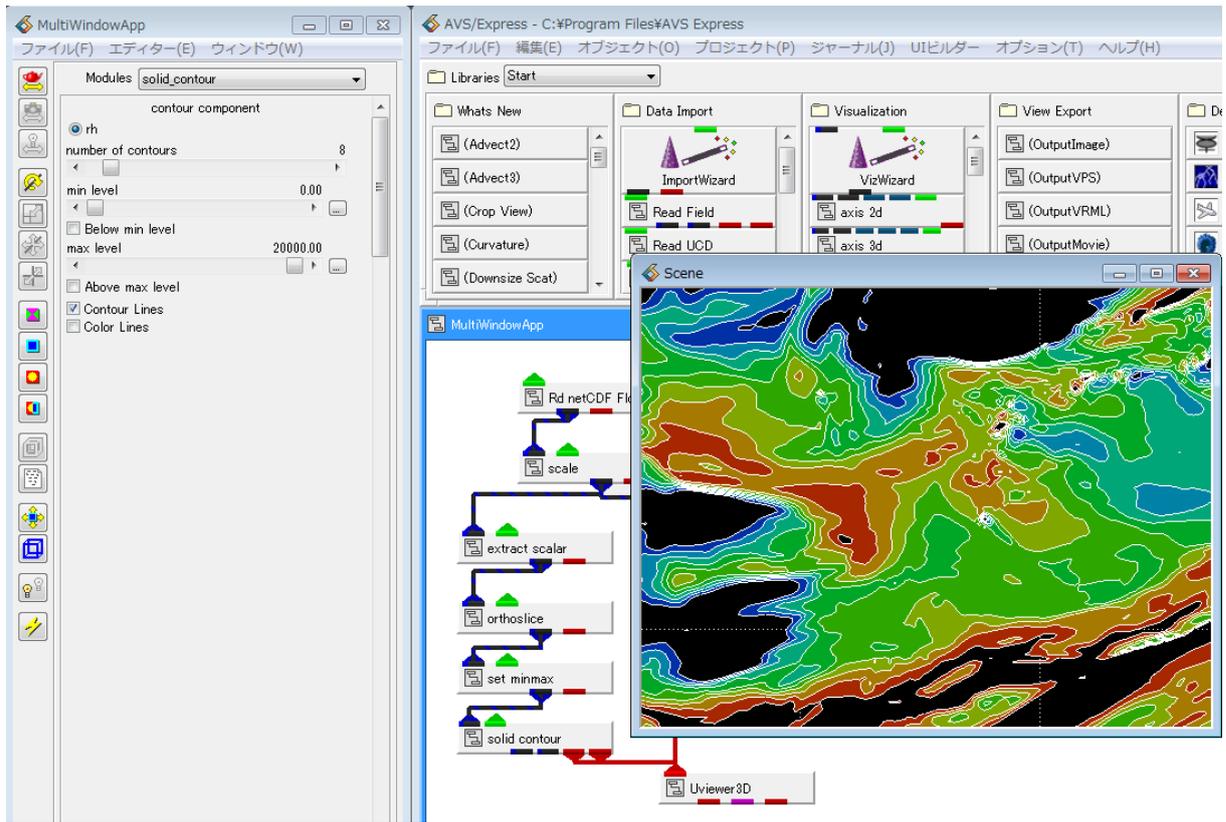


図 64 solid_contour モジュールによる表示

3.6 ベクトル図

次に紹介する可視化は、風速データ（速度ベクトル）の可視化です。まず、その風速を矢印で表現するベクトル図を作成します。

以下のモジュールを利用します。

- combine_vector
- glyph
- Arrow1, Arrow2 …
- downsize

以下に、ネットワークと可視化例を示します。

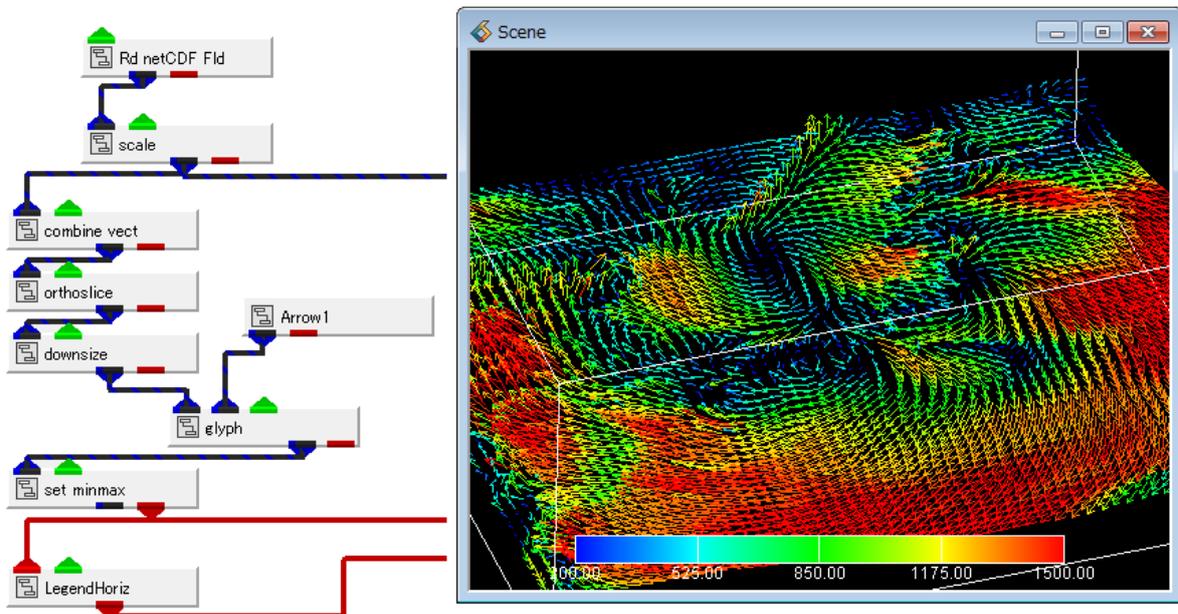


図 65 ベクトル図の表示

まず、combine_vect モジュールです。

ベクトル図を作成するには、先の等値面やコンター図で述べた対象スカラーを選択する extract_scalar モジュールの代わりに、ベクトル成分を選択する必要があります。計算結果にある U, V, W といった風向成分を選択します。

以下に combine_vect モジュールのパラメーターを示します。

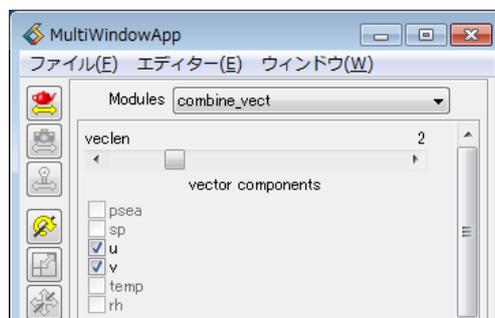


図 66 combine_vect モジュールのパラメーター

このデータには、 u と v の 2 成分の風向データがあります。
まず、**veclen** スライダーを 2 に設定してから、 u と v にチェックしてください。
3 成分のデータの場合は、スライダーを 3 に設定し、その 3 成分を選んでください。

※ このモジュールは、スライダーの数とチェックした数が合致するまで、以下のエラーが出ます。

```
--- Error detected in: module: extr_vec ---  
cannot get component array, vector is not created  
-----
```

次に接続している **orthoslice** モジュールは、先のスカラー値の可視化と同じです。ある断面を選択しています。

downsize モジュールは、フィールド型のメッシュデータの間引きを行うモジュールです。ベクトル図は、そのデータの定義点（ノード）に、図に示したような矢印を作成します。メッシュ数（サイズ）が大きい場合、ほとんどつぶれて見えなくなってしまいます。このような場合、**downsize** モジュールを使って、データを間引く（1つ飛ばし、2つ飛ばしなど）ことができます。

glyph モジュールは、あるデータの定義点（ノード）に、そのデータ値に応じて変形させた形状を配置するモジュールです。その右側に接続している **Arrow1** モジュールが、その形状です。**Arrow1** モジュールの他にも、いくつかの矢印形状がありますので、変更してみてください。

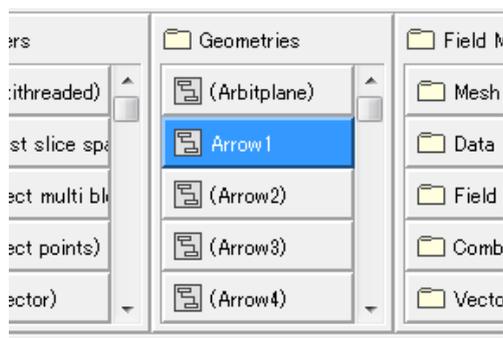


図 67 各種 Arrow モジュール

また、**glyph** モジュールのパラメーターを以下に示します。

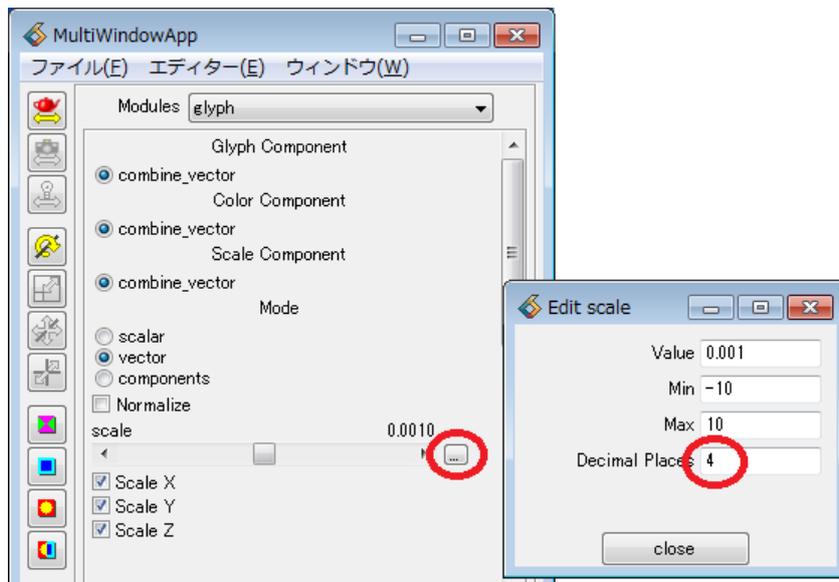


図 68 glyph モジュールのパラメーター

scale の値で、矢印の長さを調整できます。少数点がデフォルトでは下 2 桁になっていますので、それより小さい値を指定したい場合は、右側のボタンをクリックし、数値パネルで少数点の桁数を変更、Value に値を直接入力してください。

また、Normalize にチェックすると、すべての矢印の長さが同じ長さになります。

なお、この glyph モジュールの出力は、そのベクトル成分の大きさで色づけされています。「図 65 ベクトル図の表示」では、先のコンター図同様、set_minmax モジュールを使って、色と数値の範囲の調整を行っています。

単色で表示したい場合は、glyph モジュールの下に、extract_mesh モジュールを接続してください。デフォルトでは白一色で表示されます。

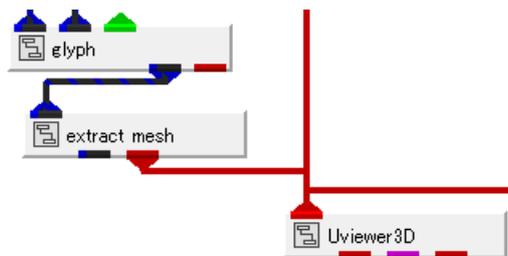


図 69 単色ベクトル図

※ オブジェクトに対する色づけや半透明など、各種属性の変更方法については、「色、その他の属性の変更第 4 章 色、その他の属性の変更」で説明します。

3.7 流線

風速データを利用し、その流れがどちらに流れるかを見ることができる軌跡を作成します。ここでは、その現在の時間のベクトルデータに対して軌跡を作成します（一般的に流線として知られている方法です）。

以下のモジュールを利用します。

- streamline
- advector

以下に、ネットワークと可視化例を示します。

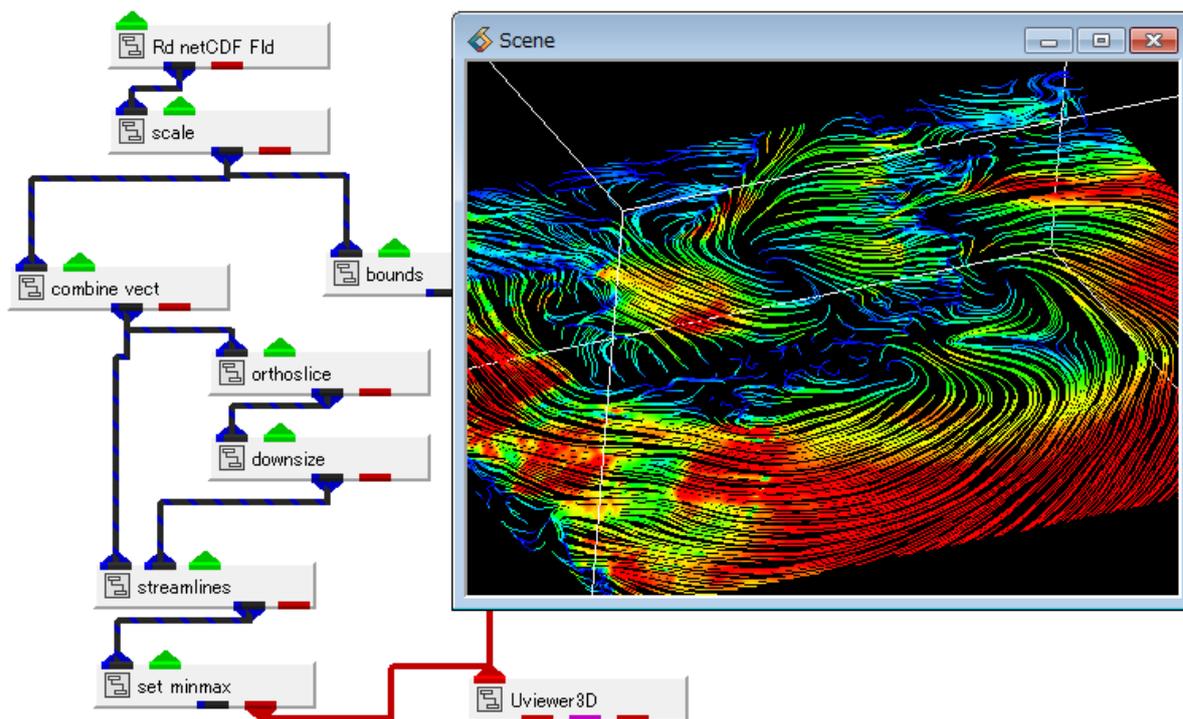


図 70 流線の表示

流線を作成するには、**streamlines** モジュールを利用します。

まず、左側のポートにベクトルデータを入力します。先のベクトル図で述べたように **combine_vect** モジュールを使って、ベクトル成分を選択します。右側のポートには、どこから流線を発生させるかの初期位置を入力します。上図の例では、**orthoslice** モジュールである断面を抽出し、さらに、その断面を間引いたノードを発生位置としています。先のベクトル図同様、出力はベクトルの大きさとなります。 **set_minmax** モジュールで色と数値の対応を設定しています。

先の例と同様、代わりに、**extract_mesh** モジュールを使えば、単色の流線を作成できます。

※ 流線の発生位置が多い場合、流線の計算に時間がかかります。

初めて実行する場合には、右側に接続する発生位置の数を少なくしてお試しください。
(**downsize** モジュールを2つ続けて使うこともできます)

advector モジュールは、流線上にある形状を動かしたアニメーションを作成できるモジュールです。パーティクルトレースとも呼んでいます。

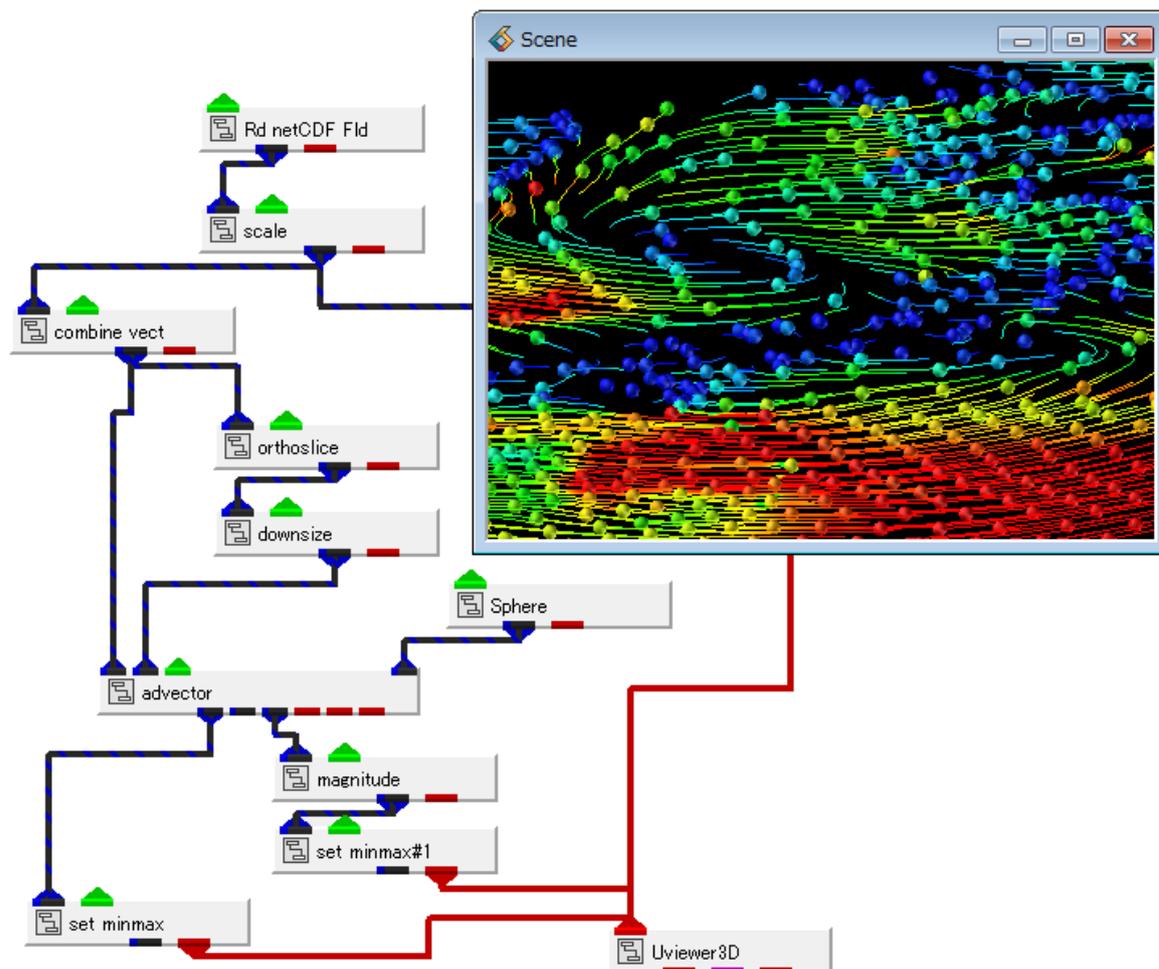


図 71 パーティクルとレース

advector モジュールを streamlines モジュールの代わりに接続しています。このモジュールには、3 番目の入力ポートがあります。ここに、そのアニメーションさせる形状を入力します。図の例では、Sphere モジュールを接続しています。「図 65 ベクトル図の表示」に述べた Arrow1 モジュールなどを接続することもできます。

出力は、一番左側の第 1 出力ポートがそのアニメーションされる形状データです。速度ベクトルの大きさになっていますので、これまで同様、set_minmax モジュールで色と数値の範囲指定を行っています。また、第 2 出力ポート（上図では接続していません）には、流線と同じ軌跡すべてが出力されます。第 3 出力ポートは、その形状の後ろの軌跡（尻尾）表示です。パラメーターで、この軌跡がオンになっている必要があります。また、この出力はベクトルデータであるため、色づけするには、magnitude モジュールを使って、ベクトルデータから大きさに変換する必要があります。

次の図に、advector モジュールのパラメーターの一部（後半部分）を示します。

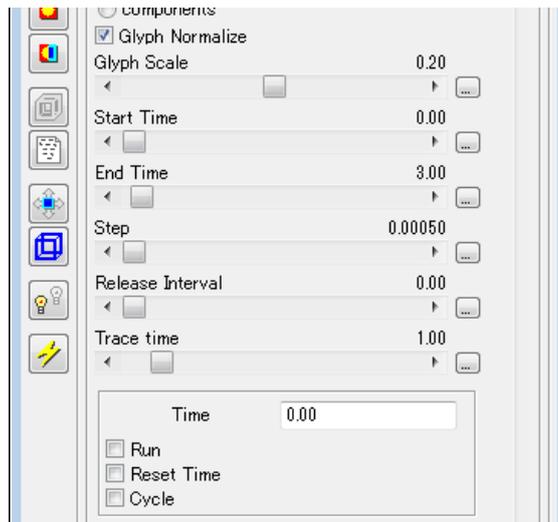


図 72 advector モジュールのパラメーター

まず、Glyph Scale で、アニメーションさせる形状の大きさ（速度の大きさで変化）にスケールリングをかけることができます。Glyph Normalize にチェックすると、同じ大きさになります。

Start Time と End Time で、いつからいつまでアニメーションを行うかを指定します。Step は、その間隔です。

この設定ができれば、Run にチェックすると、その形状のアニメーションが行われます。

Trace time は、軌跡（尻尾）の長さの設定です。第 3 出力ポートから、その軌跡が出力されます。

その他、流線の表現方法を変えることができるいくつかのモジュールがあります。

- stream_color
- stream_time
- tube
- illuminated lines

stream_color モジュールは、その流線の長さや最大、最小値、また、その発生位置で色づけすることができるモジュールです。stream_time モジュールは、同様に、その流線が発生した時点からの時間による色づけに利用できます。

※ Examples ライブラリにあるサンプルやモジュールリファレンスをご参照ください。

tube モジュールや illuminated line モジュールは、線の太さをチューブ形状に変更する、また、ライトによって反射させるなど、表現方法を変えるために利用できます。

次の図は、これらのモジュールを使って表現方法を変更したものです。

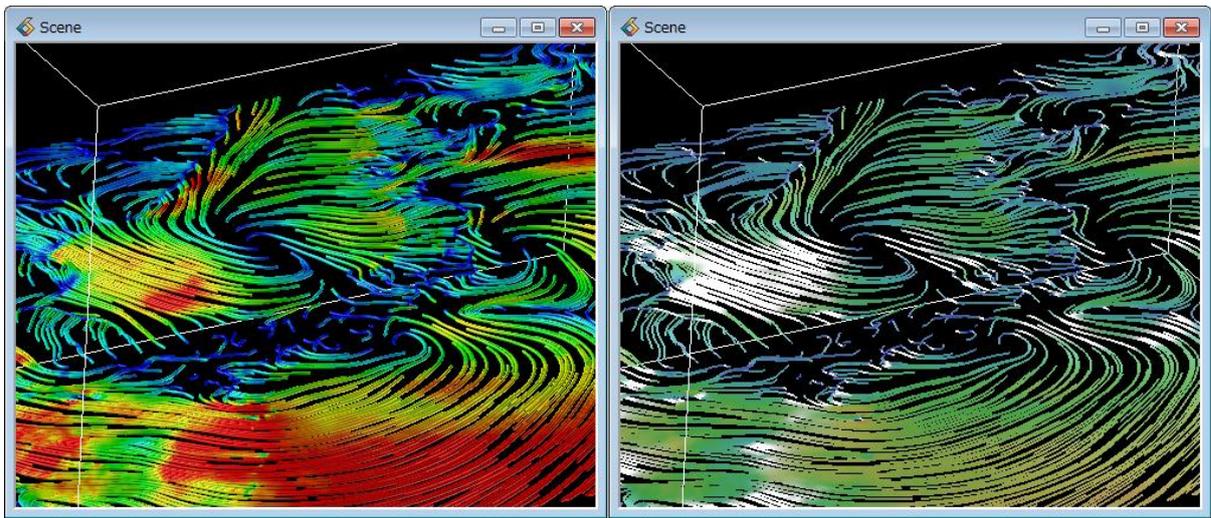


図 73 流線の表現方法の変更

- ※ illuminated line モジュールは OpenGL レンダラでのみ利用できます。
また、tube モジュールを使う代わりに、OpenGL レンダラの場合は、
ラインの属性を tube 形状に変更する方法もあります。
(「4.3 その他の属性」で説明します。)

3.8 各モジュールの組み合わせ

ここまで、スカラーデータやベクトルデータの代表的な可視化モジュールを見てきました。これらのモジュールを組み合わせ、利用することができます。

また、各モジュールの接続順番は、用途に応じて、どちらを先につなぐかについても、検討してみてください。例えば、ある断面で、スカラーデータとベクトルデータの処理を行いたい場合は、以下のように、断面を選択した後、データ成分を選ぶ `extract_scalar` モジュールや `combine_vect` モジュールを接続します。

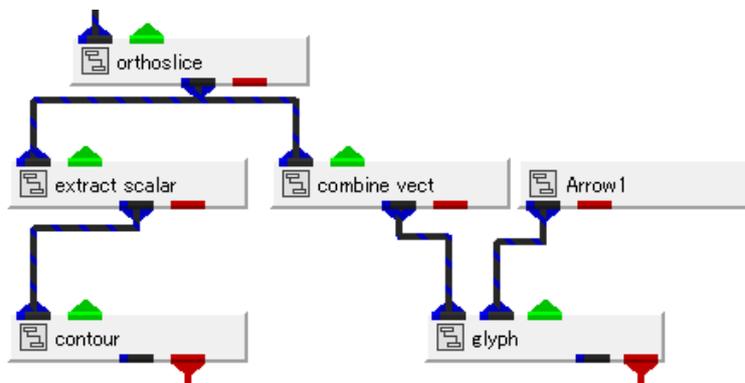


図 74 同じ断面に対する2つの処理のネットワーク

逆に、あるスカラーデータに対して、等値面の作成や断面コンター図を作成する場合には、先に `extract_scalar` モジュールでスカラー成分を選択し、その後、そのスカラー成分を処理する断面の選択や等値面モジュールを接続します。

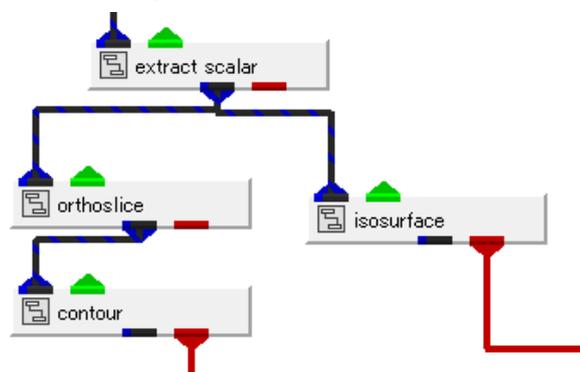


図 75 スカラー成分に対する複数の処理のネットワーク

効率的なモジュールの接続順番を検討しながら、可視化作業を行っててください。

また、「第2章 地図データの可視化」に述べた地図データの表示も一緒に行うように、ネットワークを作成します。

第4章 色、その他の属性の変更

可視化した結果の色や物体表面の反射など、属性の変更に関する内容について説明します。

4.1 カラーマップの色の調整

まず、「3.5 断面コンター図、等圧線など」に述べたように、数値データと色の分布は、通常 `set_minmax` モジュールを使って、その範囲を調整できます。

また、デフォルトでは、カラーマップは青から赤のグラデーションとなっています。このカラーマップを変更するには、以下の方法があります。

1) DataObjectColor モジュール

まず、デフォルトの青から赤のカラーマップを、簡単に以下のいずれかのカラーマップに変更できるモジュールがあります。

- ・青～白、もしくは、白～青
- ・赤～白、もしくは、白～赤
- ・青、白、赤、もしくは、赤、白、青
- ・白～黒、もしくは、黒～白

このモジュールは、Accessories ライブラリにある Graphics の中の Modules の中にあります。
(Modules をダブルクリックして開いてみてください)

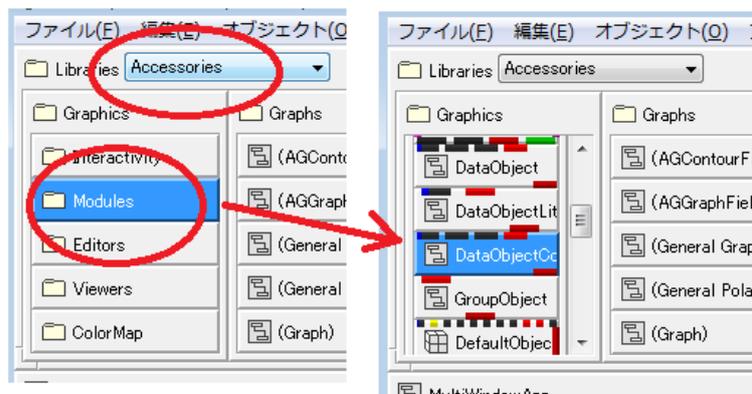


図 76 DataObjectColor モジュール

このモジュールは、赤ポートで最後にビューワーに表示を行っているモジュールの下に接続して利用します。例えば、`set_minmax` モジュールで断面コンター図などを作成している場合、以下のように接続します。

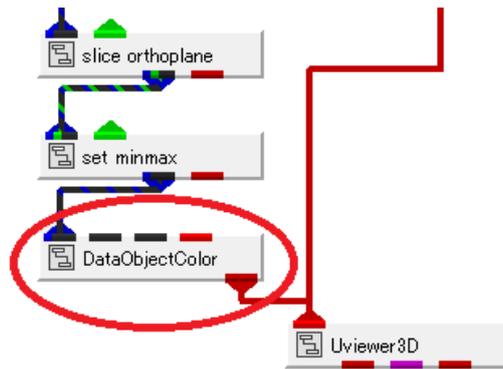


図 77 DataObjectColor モジュールの接続

モジュールのパラメーターから、色のパターンを選びます。例えば、Blue - White - Red を選ぶと、青から白、白から赤に変化するカラーマップに簡単に変更できます。

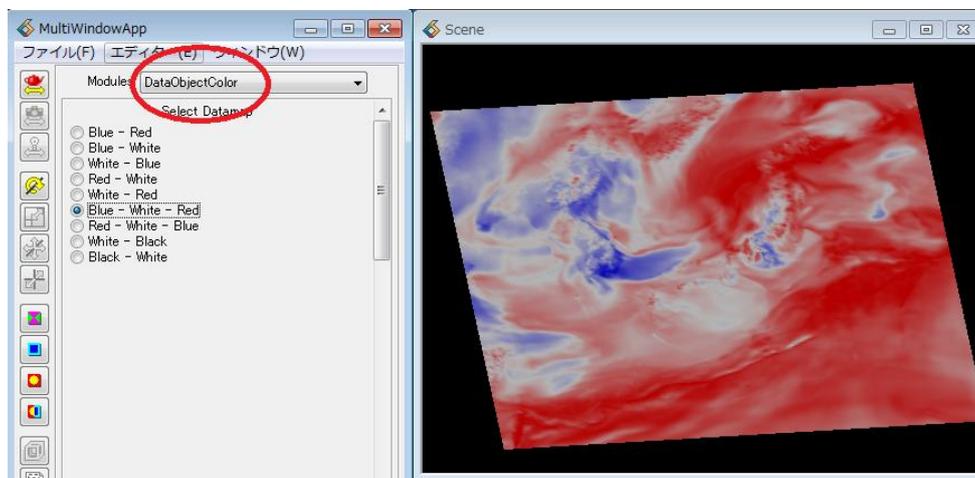


図 78 青白赤表示

2) ColormapEditor モジュール

このモジュールは、すでに、「2.3.2 ColormapEditor による色づけの変更」や「3.4 ボリュームレンダリング」で説明しました。このモジュールは、コンター図やベクトル図など、赤ポートを接続して可視化している各モジュールに接続することができます。以下のネットワークは、「図 61 contour モジュールによる表示範囲の指定」のコンター図の出力に ColormapEditor モジュールを接続した例です。

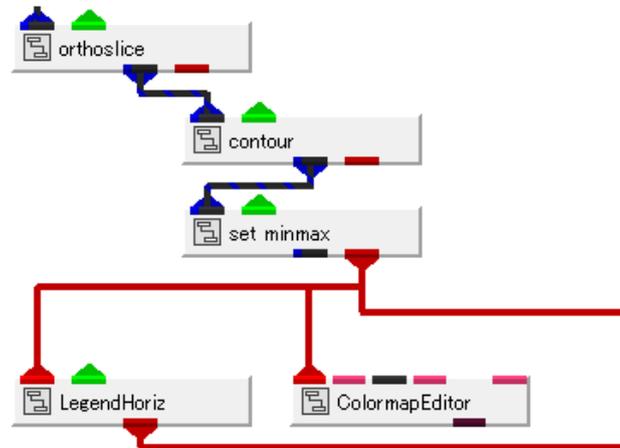


図 79 コンター図と ColormapEditor モジュール

ColormapEditor モジュールには、いくつかのプリセットされたカラーマップがあります。これらを選択し、Apply ボタンをクリックすると、そのカラーマップを適用できます（下図）。

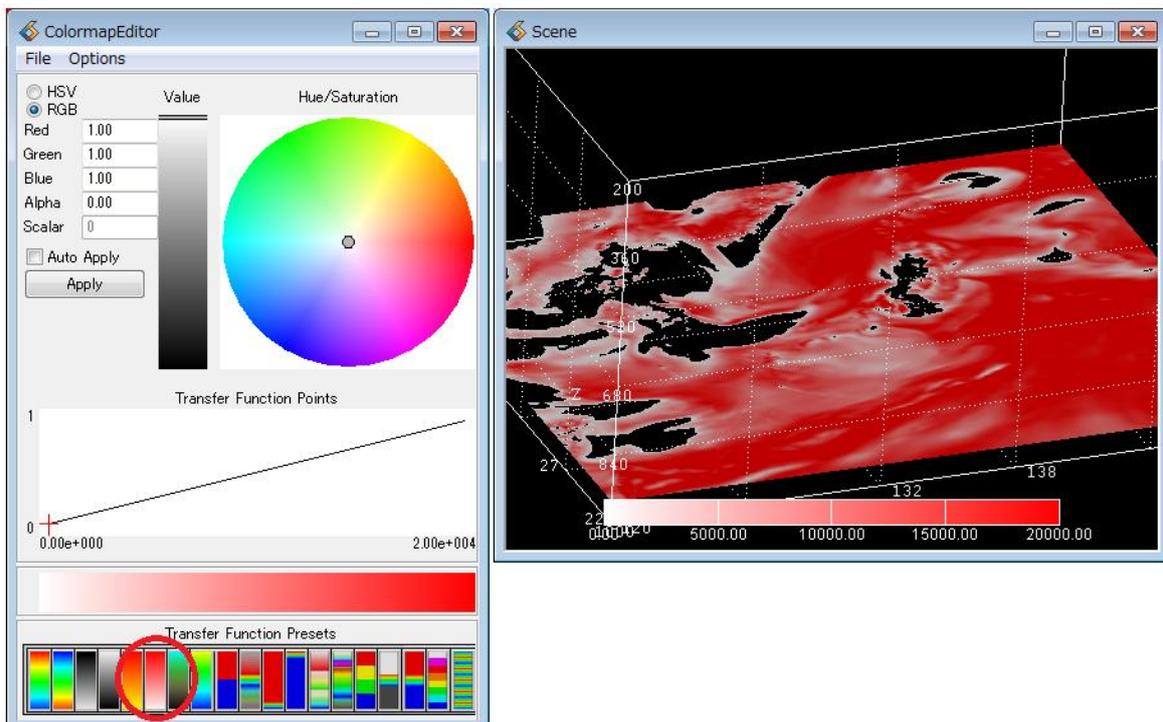


図 80 プリセットを使った変更

また、「2.3.2 ColormapEditor による色づけの変更」や「3.4 ボリュームレンダリング」で述べたように、自分でカラーマップの分布を作成できます。例えば、上図の白から赤のプリセッ

トをファイルに保存し、ファイルを書き換えます。

```
3 0 20000
0.000000 0.000000 0.000000 0.000000 1.000000
127.500000 0.500000 1.000000 1.000000 1.000000
255.000000 1.000000 1.000000 0.000000 0.000000
```

青から白、白から赤に変わるように書き換えています。この書き換え後の .cmp ファイルを読み込み、適用すると、以下の図のように、青から白、白から赤に変化するカラーマップに変更できます。

注) このカラーマップファイルを読み込む際には、ColormapEditor のパネルにあるカラーのモードを RGB にしてください (下図参照)。

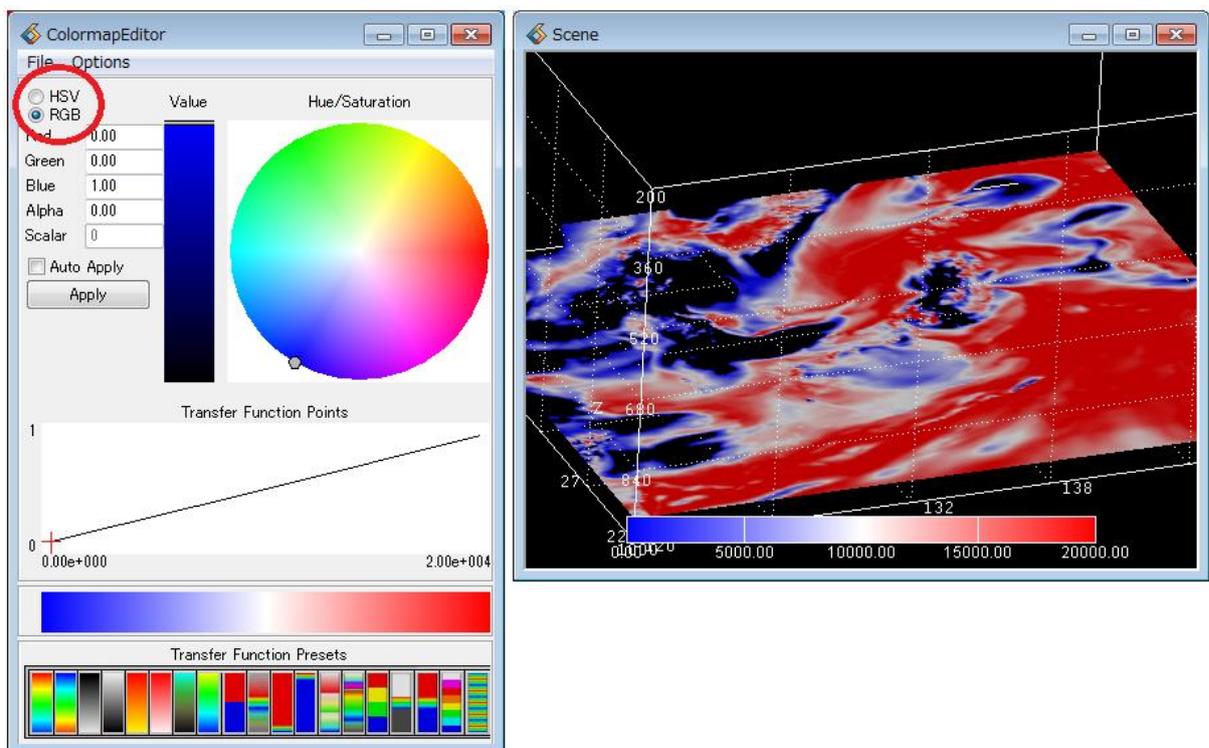


図 81 青白赤のカラーマップ

3) Datamap エディター

「図 81 青白赤のカラーマップ」の青白赤のように、3色程度の変化で表現できるカラーマップの場合は、ColormapEditor モジュールを使うよりも、以下の Datamap エディターを使った方が簡単です。

※ 複雑なカラーを作成するには、ColormapEditor モジュールの説明で述べたカラーマップファイルを作成してください。

この機能は、エディター・メニューにあります。エディター・メニューから Datamap を選びます。

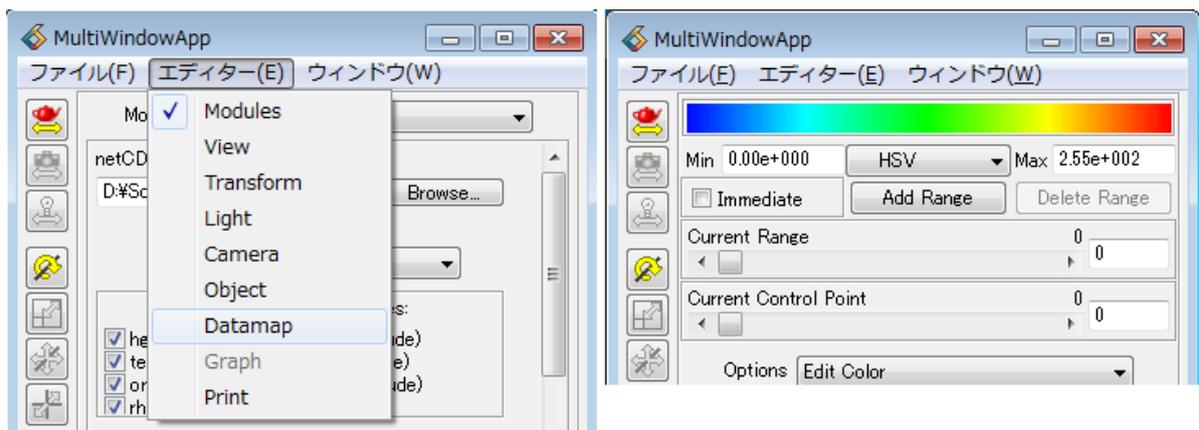


図 82 Datamap エディター

エディターにある各メニューには、最初に対象オブジェクトを選択してから、操作する機能が多くあります。この Datamap エディターも、最初に対象となるオブジェクトを選択する必要があります。

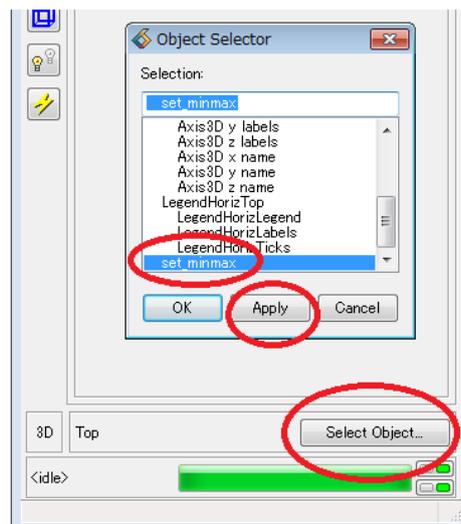


図 83 対象オブジェクトの選択

まず、Select Object ボタンをクリックし、対象オブジェクトを選びます。この例のコンター図は set_minmax モジュールで作成されていますので、set_minmax を選びます。

※ 以降の操作を行った後には、現在選択されているオブジェクトを、必ず Top に戻すようにしてください。

以下の4つの領域に着目してください。

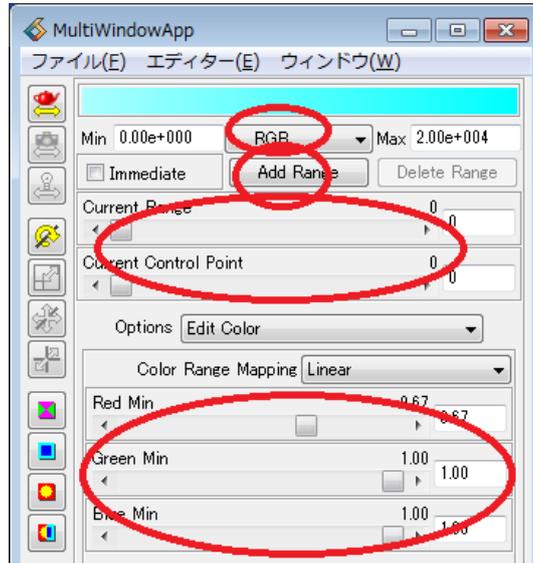


図 84 Datamap エディターのパラメーター

まず、指定を RGB で行うようにします。

次に Current Range と Current Control Point です。この Datamap エディターでは、レンジ内の左右のコントロールポイント、その端点に対して色を設定できます。

また、このレンジは、Add Range で追加することができます。

デフォルトでは、1つのレンジとその左右の値を設定するようになっています。

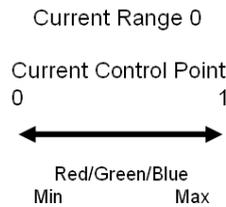


図 85 Datamap エディターのレンジ (デフォルト)

Add Range ボタンを1度クリックすると、レンジが2つ (右側に新しいレンジが追加) になります。その各レンジの両端の色を設定できます。

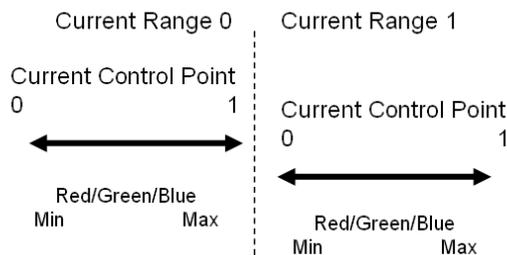


図 86 Datamap エディターの2つのレンジ

Current Range で左側 (0) か右側か (1) を選びます。

次に、Current Control Point でそのレンジ内の左側 (0) か右側か (1) を選びます。

Red/Green/Min のスライダーで、その各点の色を決定します。

先の青白赤のカラーマップを作るには、

- Add Range ボタンを1度クリックし、2つのレンジを作成

- Current Range 0 を選び、以下の値を設定

Current Control Point 0 = (Red, Green, Blue) = (0.0, 0.0, 1.0) : 青

Current Control Point 1 = (Red, Green, Blue) = (1.0, 1.0, 1.0) : 白

- Current Range 1 を選び、以下の値を設定

Current Control Point 0 = (Red, Green, Blue) = (1.0, 1.0, 1.0) : 白

Current Control Point 1 = (Red, Green, Blue) = (1.0, 0.0, 0.0) : 赤

の操作を行います。

注) このカラーマップの操作は、その各モジュールの中に設定されます。

おかしくなったり、再度作り直したい場合は、一度、そのモジュールを削除し、再度、同じモジュールをインスタンスしてから、やり直してみてください。

4.2 オブジェクトの属性の変更（色や反射）

前節では、数値データと色の対応について説明しました。

オブジェクトを単色で表現する場合（例えば「図 45 等値面の別成分による色づけ」で説明した色なしの等値面や「図 69 単色ベクトル図」など）、自分で好きな色で色づけすることができます。

その他、表示されているオブジェクトは、そのライトの設定により、反射や明るさが変化するようにレンダリングされています。その反映方法なども変更できます。

これらの表示オブジェクトの属性の変更を行うには、Object エディターを利用します。

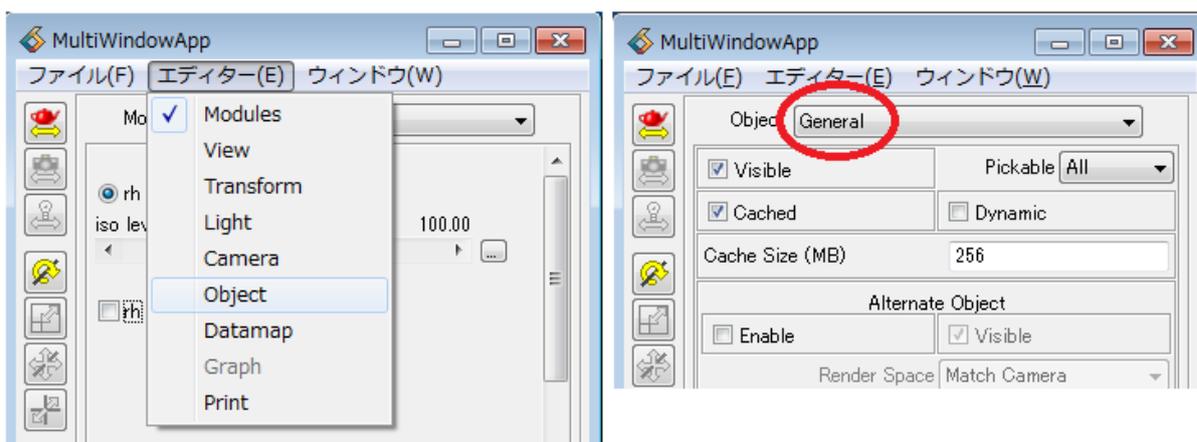


図 87 Object エディター

さらに、Object メニューでいくつかの機能毎にパラメーターが分類されています。

この機能も、まず最初に対象オブジェクトを選択する必要があります。例えば、以下の図では、isosurface（等値面）を選んでいます。

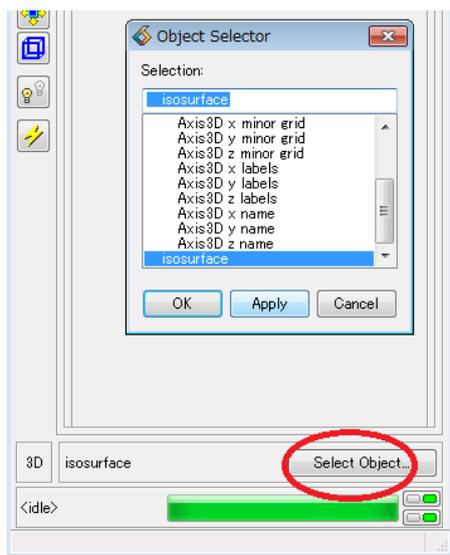


図 88 等値面オブジェクトの選択

まず、オブジェクトの色づけです。

色づけのメニューは、**Properties** に分類されています。Hue/Ssaturation/Value のダイアルを使って、その対象に色づけできます。

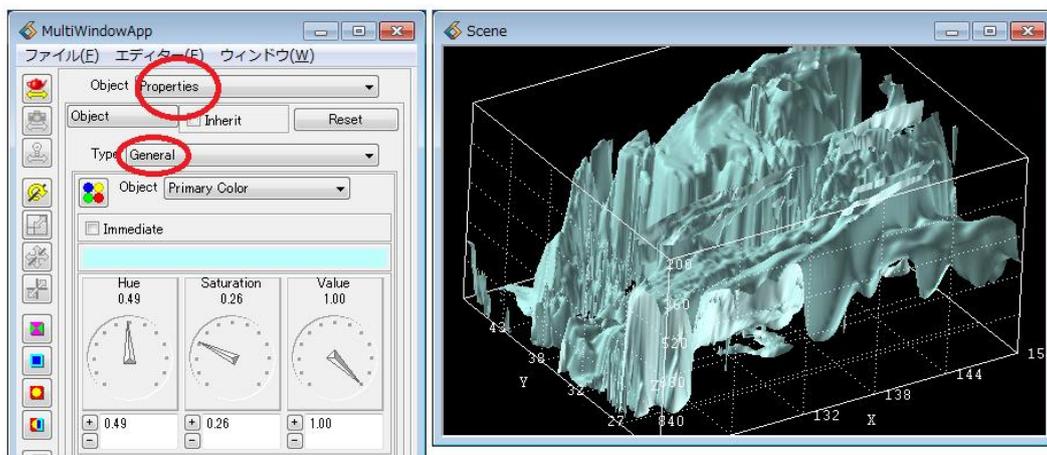


図 89 オブジェクトの色づけ

また、同じ、**Properties** の **Surface** には、その面の鏡面反射などの属性を変更できるパラメーターがあります。

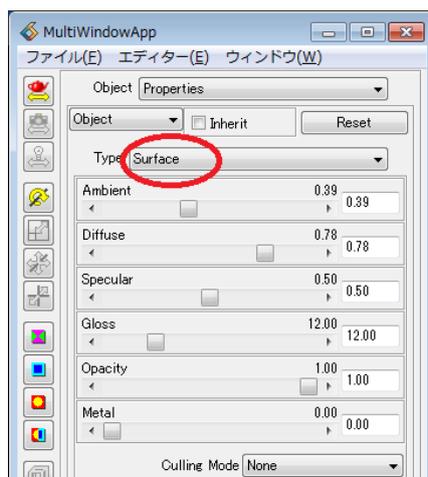


図 90 面の属性

Ambient や **Diffuse**、**Specular** を操作すると、ライト反射の度合いが変わり、キラキラした表示などに変更できます。また、**Opacity** は透明度です。半透明表示したい場合は、この **Opacity** を調整します。

- ※ 半透明表示の場合、複数の面が重なっていると、前後処理を行っていないため、きれいに表示されない場合があります。
- 上図の一番下にある **Culling Mode** を **Back**（後ろにある面を表示しない）にする、また、次節に述べるカメラのソート機能を設定するなどを行っててください。

4.3 その他の属性

その他の属性の設定について、いくつか代表的なものを紹介します。詳細、ならびに、その他の機能については、ユーザーズマニュアルの各エディターの説明をご参照ください。

1) ラインの幅、チューブ表示

また、ラインコンター図や流線など、ラインの表示に対して、以下の機能を使うと、その線幅（ピクセル幅）、また、ラインのスタイル（点線）など、表示方法を変更できます。

Object エディターの以下のメニューです。色づけなどを行った Properties ページで、Point/Line を Type で選びます。

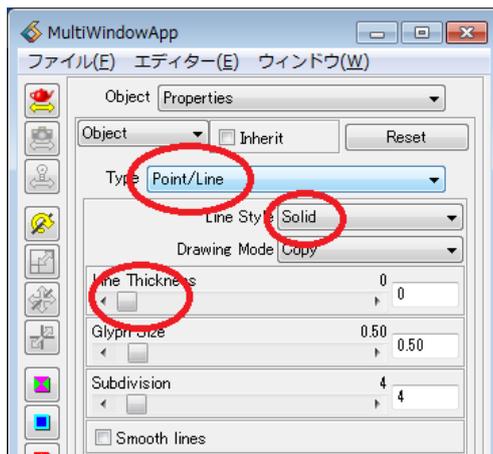


図 91 ラインの属性

また、以下のパラメーターを使うと、「図 73 流線の表現方法の変更」に説明したラインのチューブ表示と同じように、OpenGL の機能を使って、チューブ表示に変更できます。

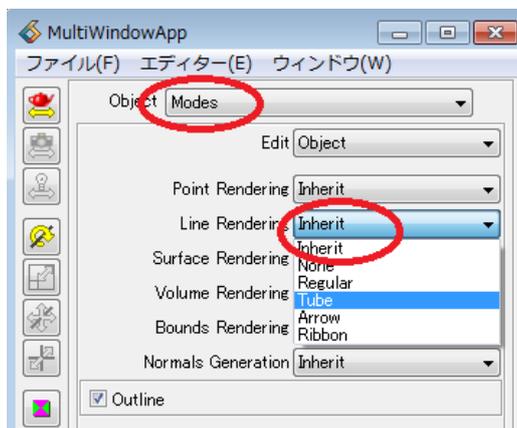


図 92 ラインのチューブ表示

いずれも、対象オブジェクトを選択してから、操作します。

2) 透視投影、平行投影

AVS/Express はデフォルトでは平行投影で表示しています。現実空間に近い表示である透視投影に変更することもできます。この切り替えは、以下のコントロールパネルにあるアイコンで行うことができます。

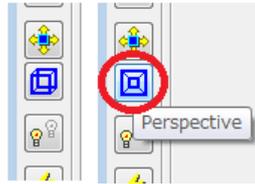


図 93 投影方法の変更

2) 線、面の上下関係（ジッター）

面と線を同じ位置に描いた場合（例えばコンター図上にベクトル図を重ねるなど）、同じ位置にあるため、線が面の中に埋もれてしまうことがあります。

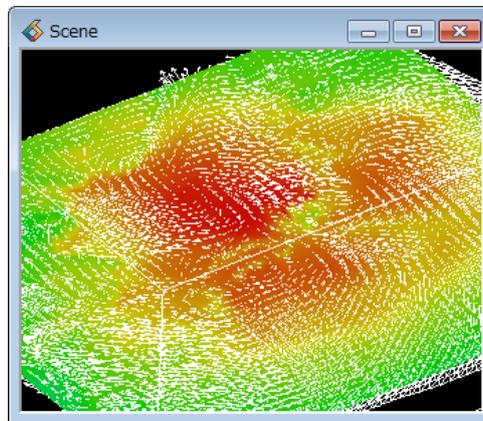


図 94 面にラインが埋もれた例

このような場合は、ジッター機能を使ってください。

対象オブジェクト（上図では、ベクトル図を作成しているモジュール名）を選択します。選択した後、以下のジッター・アイコンをクリックします。視線に対して、ラインを手前に表示します。

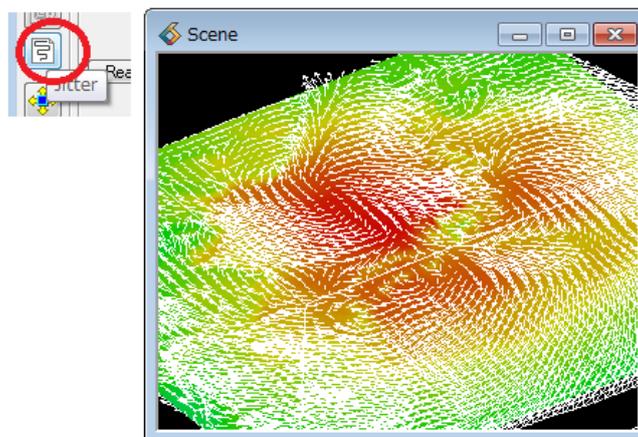


図 95 ジッター機能

3) 半透明のソート (カメラ)

この機能は、半透明をきれいに表示したい場合に、利用する機能です。通常、AVS/Express では、半透明の表示に対して、前後関係の識別を行っていません。よって、半透明の面が複数重なると、その前後関係がわかりにくくなってしまいます。

このソート機能は、視線の方向からポリゴンの奥行き関係をチェックし、その前後関係に沿って、レンダリングを行う機能です。ただし、計算に時間がかかります。また、ソフトウェアレンダリングでのみ有効です。

Camera エディター (General) にあります。以下の Depth Sort 機能です。

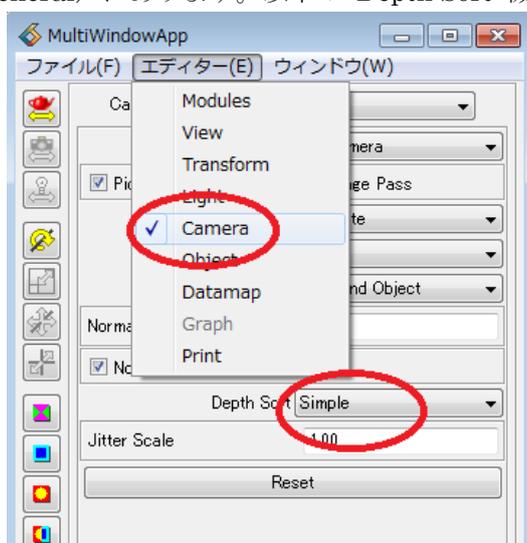


図 96 ソート機能

また、この機能はソフトウェアレンダラでのみ有効です。このレンダラの切り替えは、上図右側のアイコンで行うことができます。

Depth Sort の選択メニューを Simple に変更してみてください。

また、NNS は、さらに、詳細に前後判定を行うモードです。ただし、このモードには、かなり多くの計算時間が必要となりますので、一旦、アプリケーションとして保存した後に試すなど、利用時にはご注意ください。

※「図 90 面の属性」に述べた背面のポリゴンを非表示にする方がきれいに表示される場合もあります。

第5章 時系列データの扱い

この章では、時系列データの扱いについて、説明します。Multi_Files モジュールや Loop モジュールを利用します。

5.1 Multi Files モジュール

このモジュールは、連番ファイルで時系列データが構成されている場合に便利なモジュールです。ファイル名のリストを作成し、それを順番に読み込みモジュールに渡すことができます。

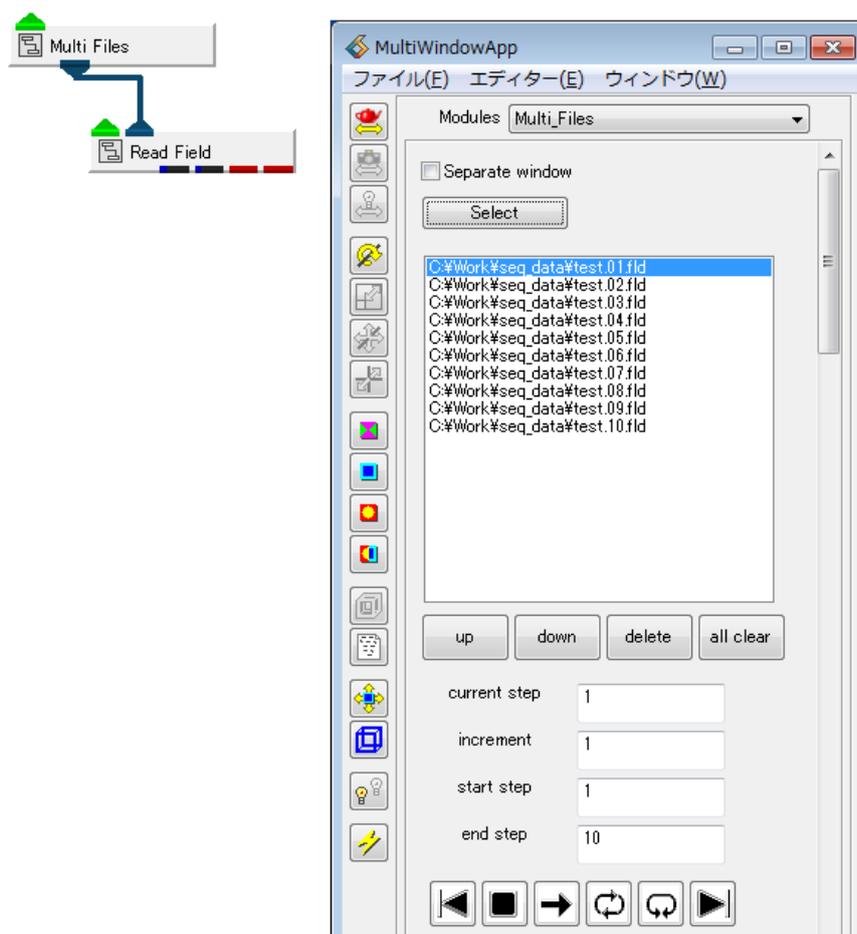


図 97 Multi_Files モジュール

Multi_Files モジュールは、上図のように、読み込みモジュールと接続して利用します。

※ 通常、このリーダーの入力ポートは表示されていません。
以降の操作でポートを出して、接続します。

また、図右は、ファイル名のリストを準備した後の表示例です。このように連番ファイルのリストを作成し、そのリストの上から順番に、ファイル名を下に送ることができます。リーダー・モジュールがそのファイル名を順番に受け取りながら、以降の処理を行います。

このモジュールを利用するには、先に述べたように、リーダー・モジュールのファイル名の入力ポートを表示させる必要があります。以下の図は、Read_Field モジュールによる例です。

まず、Read_Field モジュールの上でマウス右ボタンで表示されるメニューから、パラメータ表示を選びます。

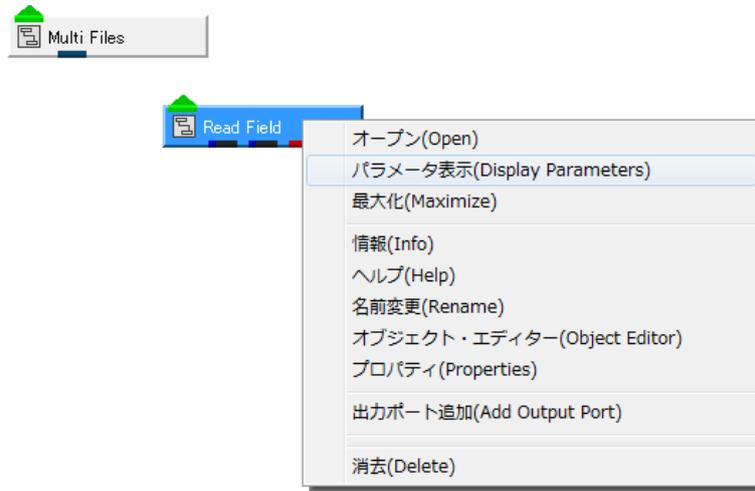


図 98 Read_Field モジュールのパラメータ表示

さらに、その中の filename の上でマウス右メニューから入力ポート追加を選びます。

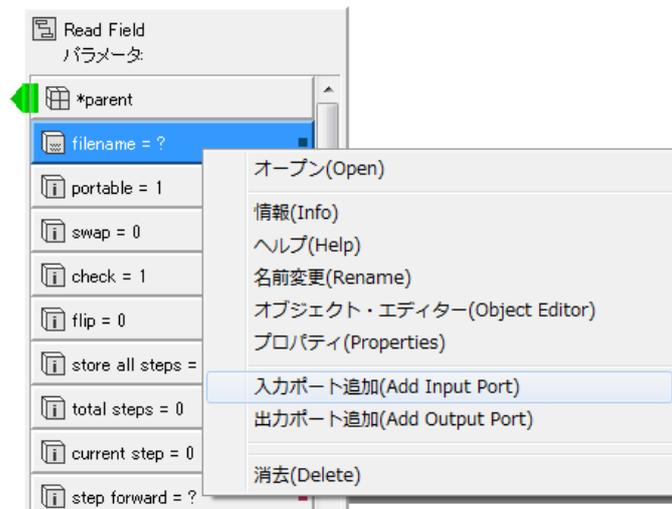


図 99 filename の入力ポート追加

次に、作成されたポートの上で、マウス右メニューから、ポートを出すを選びます。

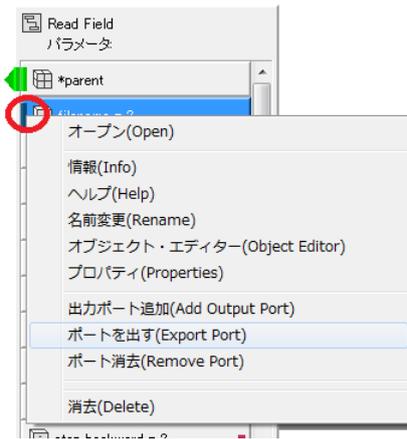


図 100 filename の入力ポートの表示

以上の作業で、ポートができます。

Read_Field モジュールをダブルクリックして、もしくは、マウス右メニューからクローズし、元に戻します。

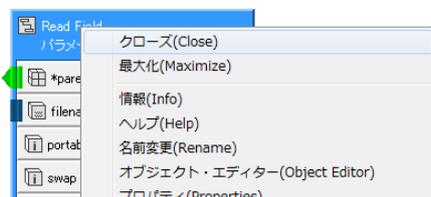


図 101 モジュールのクローズ

ポートができますので、Multi_Files モジュールの出力と接続します。

Multi_Files モジュールは、あるフォルダにある複数のファイルのリストを作成するモジュールです。まず、最初に Select ボタンをクリックします。

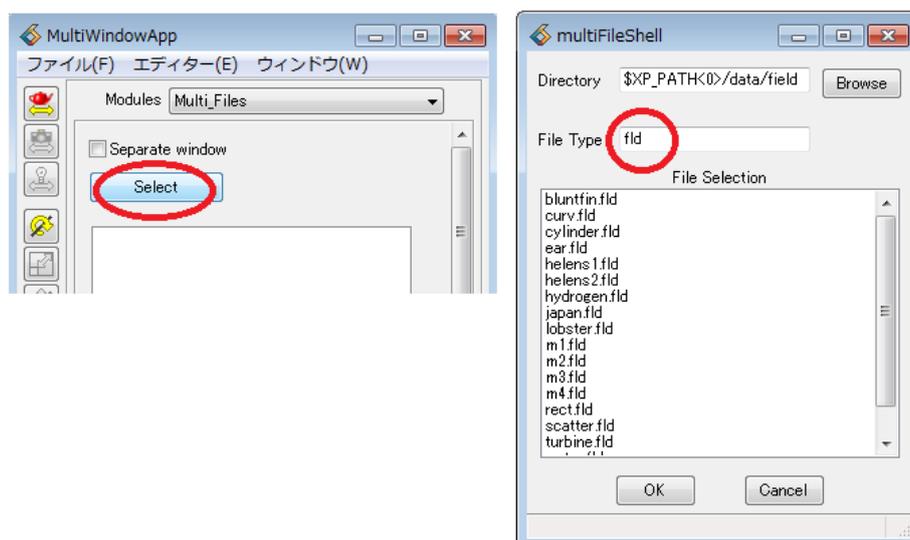


図 102 Multi_Files モジュールの Select

デフォルトでは、AVS/Express のサンプルファイルのフォルダが開きます。

ここにある File Type で拡張子を指定してください (Read_Field モジュールの場合は、fld の

ままです)。また、図に示すように、データファイルは連番である必要はありません。そのフォルダにある指定した拡張子のファイルがすべて表示されますので、**abc** 順に並んでいれば、それでも結構です。**File Selection** にリストされたファイルをマウス左で選択し、**OK** ボタンをクリックします。

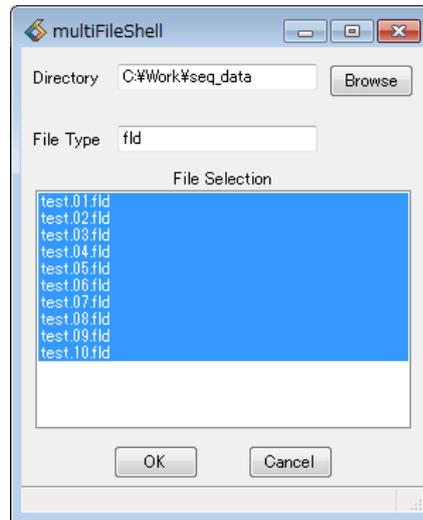


図 103 ファイルリストの選択

最初の「図 97 Multi_Files モジュール」に示すように、**Multi_Files** モジュールのパラメーターにファイル名がリストされます。このリストの中でファイル名をクリックすると、そのファイル名が選ばれ、下流のリーダー・モジュールに送られます。

また、パラメーターの下部には、**start step** と **end step**、また、**increment** を使って、自動再生の設定を行います。再生ボタンを押すと、ファイル名を自動的に送ることができます。

5.2 Loop モジュール

Loop モジュールは、ループ処理を行い、そのカウント値を生成するモジュールです。**Read_netCDF** モジュールや **Read_WRF** モジュールのように 1 ファイル内に全ステップのデータがあり、モジュールのパラメーターでその時間ステップを指定する場合に、そのステップ値を生成できます。

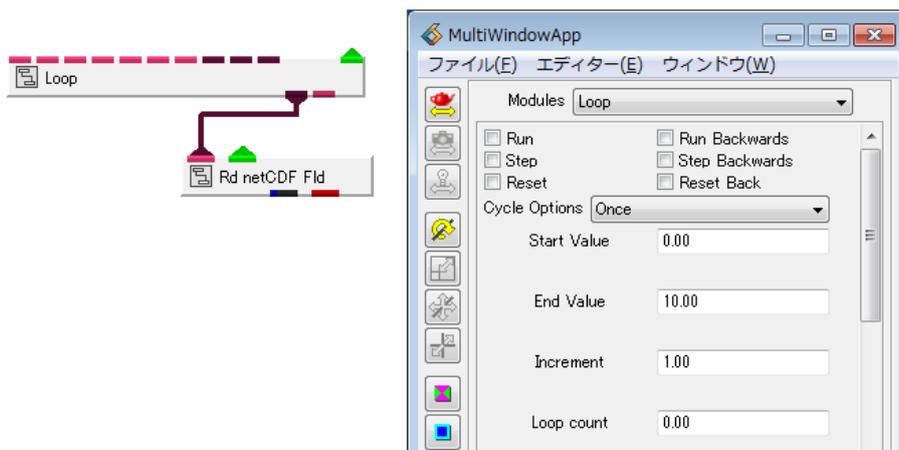


図 104 Loop モジュール

Loop モジュールは、上図のように、読み込みモジュールと接続して利用します。

※ 通常、このリーダーの入力ポートは表示されていません。
以降の操作でポートを出して、接続します。

また、右図は、Loop モジュールのパラメーターです。

Start/End Value と Increment を指定します。Run で実行すると、そのカウント値が生成されます。その値を、リーダー・モジュールの Time ステップに渡して、時間ステップのデータを連続処理します。

まず、リーダー・モジュールからポートを出力します。

該当モジュールの上で、マウス右メニューからパラメーター表示を選びます。

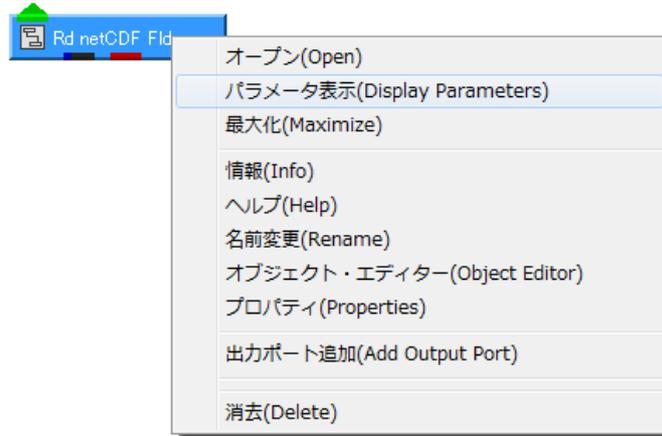


図 105 Rd_netCDF_Fld モジュールのパラメーター表示

timeStep の左側のポートの上でマウス右メニューを開き、ポートを出すを選びます。

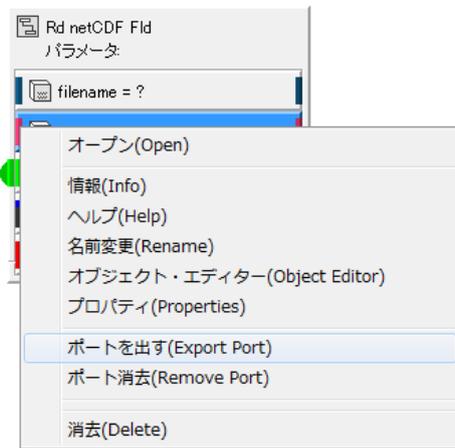


図 106 timeStep のポートの表示

Loop モジュールの count とこの timeStep を接続します。

そのデータの時間ステップにあわせて、Loop モジュールのパラメーターを設定し、Run で実行します。

第6章 画像、動画ファイルへの出力

可視化結果の保存方法について説明します。

6.1 静止画の保存

まず、静止画の保存です。静止画に保存するには、以下のモジュールが利用できます。

- OutputImage

OutputImage モジュールは、TIFF や PNG など、各フォーマットの画像ファイルを作成できるモジュールです。以下の図のように、ビューワー・モジュールの出力ピンポートと接続します。

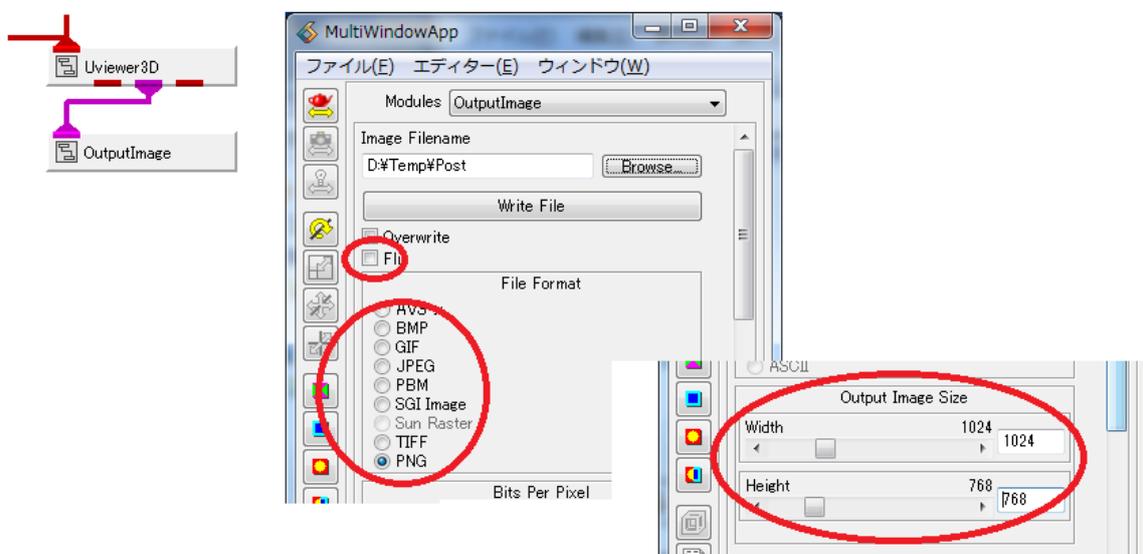


図 107 OutputImage モジュール

上図右は、OutputImage モジュールのパラメーターです。以下の設定を行います。

まず、Flip パラメーターです。画像の上下を反転させるパラメーターで、通常、オフにしてください。次に、このモジュールは、出力する画像の解像度を設定できます。一番下にあるスライダー（と数値指定）を使って、作成したい画像の解像度を設定してください。

出力するフォーマットを選び、Browse ボタンで適当な場所にファイル名を指定（拡張子はフォーマットにあわせて自動設定されます）し、Write ボタンをクリックしてください。

注) 画面の見た目の違い

このモジュールは、指定した解像度で、このモジュール内部で再度レンダリングを行っています。

そのため、現在ビューワーに表示されている絵と見た目が異なる場合があります。

ひとつは解像度です。

解像度の違いによる違いを近づけるには、ビューワーの縦横比を、出力したい画像の解像度の比率と同じ比率に設定してください。

ビューワーのサイズは、View エディターで設定できます。

もうひとつは、レンダラの違いによる違いです。

このモジュールはソフトウェアレンダラでレンダリングしています。
画面上のビューワーをハードウェアレンダリングしている場合、そのレンダラの違いで出力した画像との違いが生じる場合があります。

画面上のレンダラをソフトウェアレンダラにすることで、同じレンダラによる処理となります。

Developer 版をご利用の場合には、逆に **OutputImage** モジュールの内部を開き、レンダラをハードウェアレンダラに変更することもできます。

OutputImage.output_field.View.renderer の値を **OpenGL** に変更します。

(ただし、作成できる解像度はディスプレイのサイズまでとなります)

Vis 版の場合、ソフトウェアレンダラを使う、もしくは、ビューワーの画面を直接ダンプしてください。

Windows 版では、エディターメニューから **View** を選んだ場所にある、以下のボタンで、クリップボードにビューワーの画像をコピーすることもできます。



図 108 クリップボードへのコピー

6.2 動画の保存

動画に保存するには、以下の2つのモジュールが利用できます。

- `image_capture`
- `geom_capture`

これらのモジュールも先の **OutputImage** モジュール同様、ビューワーモジュールの出力ピンクポートと接続して利用します。

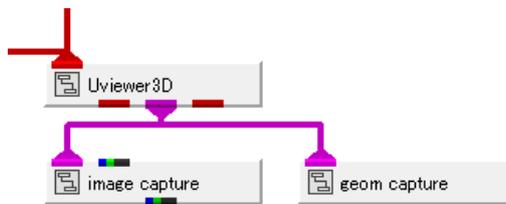


図 109 capture モジュール

1) `image_capture` モジュール

`image_capture` モジュールは、ビューの変化（画像の変化）をキャプチャすることができるモジュールです。画面上で回転などの操作を行った様子や時系列データの更新など、画面が更新される度に、その変化をキャプチャし、最終的に動画として保存できます。

操作方法は、以下の通りです。

まず、このモジュールのパラメーターを開き、**Mode** を **Capture from View** に変更します。

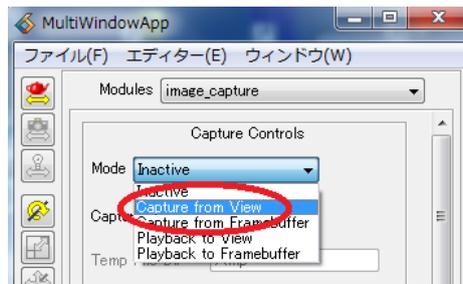


図 110 キャプチャの開始

この Mode を変更すると、以降の画面上の変化がキャプチャされますので、ご注意ください。
 この後、画面上を回転させる、時系列データを更新するなど、一連の作成したい動画の動きを再現してください。

キャプチャしたい処理が終わったら、Mode を Inactive に戻します。

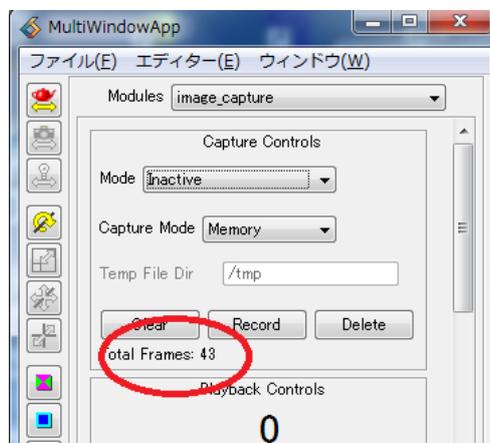


図 111 キャプチャの終了とフレーム

キャプチャできると、図のように Total Frames に記録したビューの更新枚数が表示されます。
 このモジュールは、プレビューの機能も持っています。
 今度は、Mode を Playback to View に設定し、下図の再生ボタンを押します。

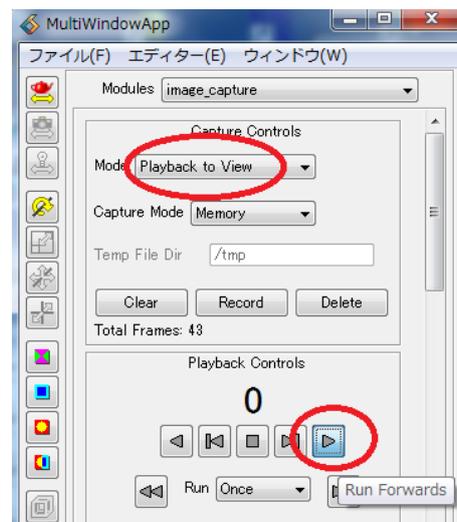


図 112 プレビュー

ビュー上に、キャプチャした画像のアニメーションが再生されます。

正しくキャプチャされているのを確認できたら、下の方にあるパラメーターを利用して、AVI や MPEG 動画に保存してみてください。

2) geom_capture モジュール

一方、geom_capture モジュールは、ビューの更新ではなく、形状の更新（形状の変化）をキャプチャすることができるモジュールです。画面の更新ではないため、単にオブジェクトの回転等の操作を行っただけではキャプチャしません。時系列データの変化に応じて、表示している等値面の形状が変わるなど、形状が変化した様子をキャプチャし、独自フォーマット（GFA）に保存することができます。

操作方法は、image_capture モジュールとほとんど同じです。

まず、Mode を Capture に変更します。

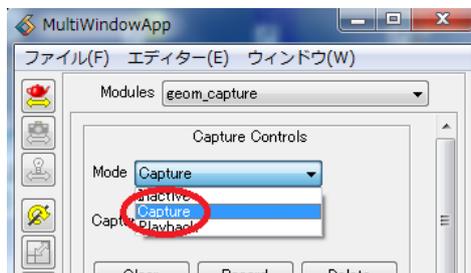


図 113 キャプチャの開始 (geom_capture)

一連の変化を再現します。

Mode を Inactive に戻します。

また、このモジュールもプレビューの機能を持っています。Mode を Playback に設定し、再生ボタンを押してみてください。ビューワー上で、その変化が再現されます。また、画像ではなく、形状をキャプチャしているため、マウスで回転などを行うことができます。

保存は、同モジュールのパラメーターの下方にある Browse ボタンでファイルを指定することで行います。gfa の拡張子のファイルに保存してください。

注) Format の指定

Format メニューから AVS gfa (32-bit) もしくは AVS gfa (64-bit) が選べます。

64bit 機種で動作させている場合も、通常は 32bit をご利用ください。

形状のサイズが 2G を超えるような大規模の場合のみに 64bit を使います。

保存した GFA ファイルは、以下のいずれかで再生できます。

- geom_replay
- 3D AVS Player

geom_replay モジュールは、GFA の再生用のモジュールです。予め作成された GFA ファイルがある場合、AVS/Express 上で再生できます。以下のように、単にビューワーと接続して利用します（ビューワーの赤ポートの入力は使いません）。

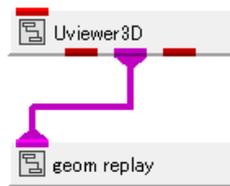


図 114 geom_replay モジュール

3D AVS Player は、サイバネットシステム（株）のホームページからダウンロードできる フリーのビューワーです。AVS/Express のインストールされていない環境で、GFA ファイルの再生を行うことができます。

参考) 3D AVS Player

<http://www.cybernet.co.jp/avs/products/avsplayer/>

6.3 連番画像の保存

連番画像として保存するモジュールもあります。

- ・ OutputSequentialImage

このモジュールは、KGT ライブラリに収録されています。

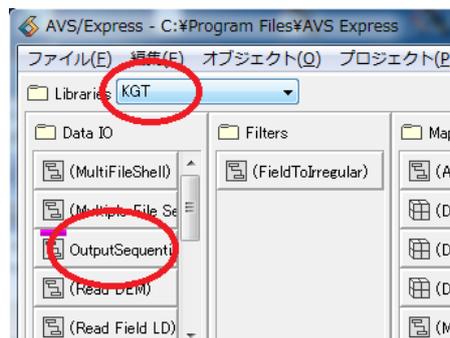


図 115 OutputSequentialImage モジュール

同様に、ビューワーモジュールのピンクのポートを接続します。

このモジュールの利用には、若干、注意が必要です。このモジュールは画面の更新がある度に、画像ファイル（連番）を作成します。まずは、画像の出力フォルダを何も含んでいない新規フォルダにするようにしてください。また、以降の手順のイメージ番号を、これから画像を生成するというタイミングで指定するようにしてください。

以下の手順で操作します。

まず、**Browse** ボタンをクリックし、適当な何も入っていない空のフォルダに、作成する画像の名前（接頭子のみ）を指定してください。

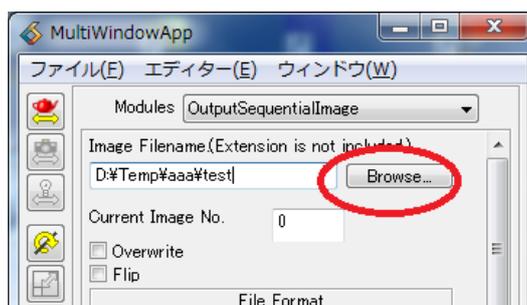


図 116 出力フォルダと名前の指定

例えば、この状態で画面のビューを回転等すると、その画面の更新の度に、画像ファイルが生成されます。また、下図の **Current Image No** の値が増加します。

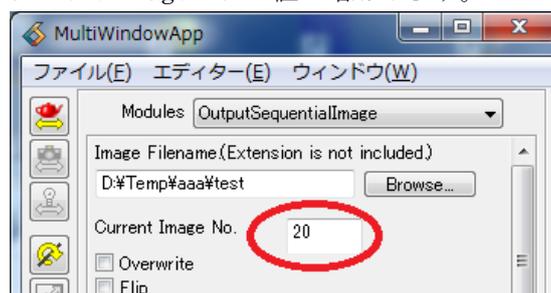
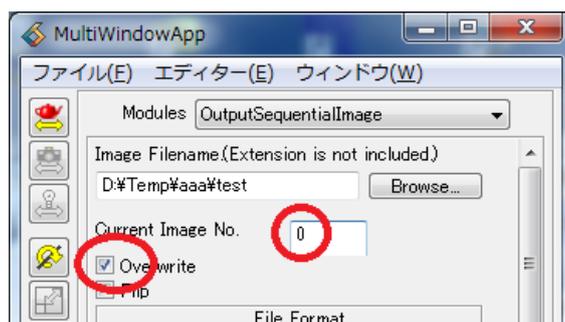


図 117 イメージ番号

そこで、通常は、このモジュールの全パラメーター（解像度も含め）を設定するまで、ピンクのポートは接続しないようにしてください。

パラメーターの設定が終わり、且つ、時系列等の記録したい内容の再現準備ができたなら、ピンクのポートを接続します。

次に、下図のように、**Current Image No** を 0 に戻し、且つ、不要ファイルがすでに行っている場合にも上書きするように **Overwrite** にチェックします。



ここまで準備できたら、時系列等の再現を行います。

指定したフォルダの下に、指定した名前 + 連番 + 拡張子 の画像ファイルが生成されます。

※ 内部的には **OutputImage** モジュールを使っています。

よって、先に述べた画面の見た目の違いについては、同じことが言えます。

6.4 連番画像の保存 (image_capture の Disk モードの利用)

連番画像を作成したい場合に、Windows 版では、image_capture モジュールを利用することもできます。

先に述べた image_capture モジュールの使い方の中で、以下のパラメーターを先に設定してください。

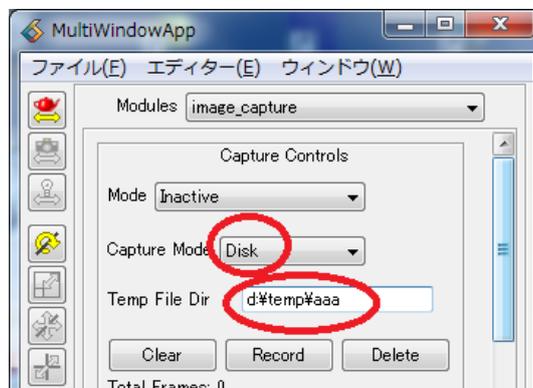


図 118 Disk モードの指定

キャプチャを開始する前に、Disk モードを指定し、その下の入力欄で、何も入っていない空のフォルダを指定します。

通常、デフォルトでは、動画のキャプチャはメモリ上に行っています。

この Capture Mode は、メモリではなく、Disk に展開しながら、キャプチャすることができる機能です。

この状態で、キャプチャを行ってみてください。

指定したフォルダに、img00_00000.bmp , img00_00001.bmp ... のような、BMP ファイルが作成されます。

このフォルダにできた BMP ファイルを、適当なフォルダに、“コピー”することで、連番ファイルを取得できます。

注) この BMP ファイルは、image_capture モジュールそのものを削除したり、Clear ボタンでキャプチャした内容をクリアすると、自動的に削除されます。よって、キャプチャしている間に、別のフォルダに“コピー”してください。

- ※ 本書に記載されている事項は、予告なく変更されることがありますので、ご了承ください。
- ※ 本書に記載の内容については、正確であることに努めていますが、本書の利用によって、生じた結果については、一切責任は負いません。

- ※ 本書の内容を、サイバネットシステム（株）の許可無しに、全部、または一部を無断で使用、複製等することはできません。

AVS/Express による気象データの3次元可視化

2013年7月22日 第1版
2016年7月12日 改訂

Copyright 2016
CYBERNET SYSTEMS CO., LTD.

発行 サイバネットシステム株式会社
ビジュアライゼーション部
AVS サポートセンター