

*AVS/Express* による  
材料データの 3次元可視化

---

(ボール&スティック、電荷密度分布表示)

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 データの読み込み	2
2.1 利用するデータタイプ	2
2.2 UCD データと読み込み	2
2.3 Field データ (離散点) と読み込み	8
2.4 Field データ (3次元配列) と読み込み	11
2.5 一般的なフォーマット	13
2.5.1 Lammmps データの読み込み	13
2.5.2 PDB データの読み込み	14
2.5.3 Gaussian データの読み込み (log ファイル)	16
2.5.4 Gaussian データの読み込み (Cube ファイル)	18
2.5.5 VASP データの読み込み	19
2.5.6 XYZ データの読み込み	20
2.5.7 ATK netCDF データの読み込み	21
2.5.8 TOMBO データの読み込み	21
2.5.9 SCIGRESS データの読み込み	23
2.5.10 OCTA データの読み込み	23
第3章 データの可視化 (ボール&スティック)	24
3.1 ソフトウェア球の表示 (UCD)	24
3.2 ソフトウェア球の表示 (Field や Lammmps)	29
3.3 ポリゴン分割球	30
3.4 ポリゴン分割球 (glyph モジュール)	32
3.5 その他の形状	36
3.6 スティックの表示 (チューブ形状)	38
3.7 スティックの表示 (ライン形状)	40
第4章 データの抽出	43
4.1 空間に対する抽出	43
4.2 空間に対する抽出 (カット)	49
4.3 データ値を利用した抽出	55
第5章 データの可視化 (メッシュデータ)	60
5.1 データの準備	60
5.2 断面コンターによる表示	60
5.3 等値面表示	62
5.4 サンプル	62
5.5 ボール&スティックとの重ね合わせ	63
第6章 色、各属性の変更	64
6.1 カラーマップの色の調整	64
6.2 オブジェクトの属性の変更 (色や反射)	70
6.3 その他の属性	72
第7章 時系列データの扱い	76
7.1 時系列フォーマット	76

7.2 Multi Files モジュール.....	77
7.3 Loop モジュール.....	80
第8章 画像、動画ファイルへの出力.....	83
8.1 静止画の保存.....	83
8.2 動画の保存.....	84
8.3 連番画像の保存.....	87
8.4 連番画像の保存 (image_capture の Disk モードの利用) .....	89

---

## 第1章 概要

高分子や金属材料、ゴム材料、ナノチューブ、グラフェンなど、さまざまな物質、材料に関する研究が行われています。

本書では、特に、このような物質科学、材料系のデータを、AVS/Express を用いて、ボール&スティック形式で可視化する方法について紹介しています。また、後半では、メッシュデータに関する可視化とその合成表示などについても述べます。

粒子やボンドといった表示方法の他、空間的に間引いて表示する方法、また、値によってデータを抽出（閾値処理）する方法などについて紹介しています。

また、電荷密度分布など、メッシュ状のデータの可視化についても触れます。

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 データの読み込み

まず、この章では、データの読み込みについて説明します。

### 2.1 利用するデータタイプ

AVS/Express では、数値データを扱う形式として、2つのデータタイプ、UCD（ユーシーデー）と Field（フィールド）があります。

- UCD

点や線、三角形などの要素の集まりで表現できるデータです。

- Field

配列で定義できるデータで、点列を表す1次元配列や画像のような縦横の2次元配列、また、Z方向に積み上げられた3次元配列を扱うことができます。

材料データの分子や原子といった粒子群を表現するには、UCDデータの Point タイプ、もしくは、Field の1次元配列で扱うことができます。

分子間の結合もあわせて表現するには、UCDデータの Point と Line のタイプを利用します。一方、電荷密度のようにメッシュ状のデータの場合には、Field の3次元配列データを利用します。

その他、Lammps や VASP、Gaussian など、一部の一般的なフォーマットを読み込むモジュールもサポートされています。ご利用のデータをこれらのリーダーで直接読み込むことができないかについても、あわせてご検討ください（詳細は後述）。

独自のソルバーやフォーマットがサポートされていないデータを AVS/Express で読み込む場合には、UCD や Field の形式にデータを変換してください。

※ データフォーマットの詳細については、ユーザーズ・ガイドをご参照ください。

### 2.2 UCD データと読み込み

この節では、UCDデータのフォーマットについて説明します。

分子と分子間結合など、粒子とボンドを作成するには、UCDデータの Point と Line 要素を利用します。

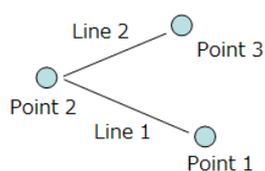


図 1 Point と Line 要素

UCDデータでは、データの定義点（XYZの座標位置）をノードと呼びます。また、要素をセルと呼んでいます。上図の場合は、ノードが3点（Point1, Point2, Point3の位置）、セルは2種類（Point

と Line セル) を利用しており、且つ、Point セルが 3 個、Line セルが 2 個あります。

データフォーマットについては後述しますが、まず、ID と座標値を準備します。例えば、0, 1, 2 の ID にそれぞれの座標値 (x,y,z) を準備します。

```
0  0.0, 0.0, 0.0
1  -1.0, 0.5, 0.0
2  -0.2, 1.0, 1.0
```

次に、要素の種類と、その要素がどの ID で構成されるかを示す接続リストを準備します。

```
pt  0
pt  1
pt  2
line 0 1
line 1 2
```

Point のタイプを示す pt が 3 個、Point は 1 点で構成されるため、その ID は 1 つです。Line は 2 点で構成されるため、どの ID を接続するかを 2 つの ID で示します。

※ 上記の記述は説明のために抜き出したもので、実際のフォーマットでは、要素の ID などにも記述します。  
実際のフォーマットについては、後述します。

座標と要素のタイプ、その接続リストが準備できたら、そのノードやセルに対して、データ (物理量) を持たせることができます。データの定義点上のデータをノードデータ、要素の中心で定義されるデータをセルデータと呼びます。

例えば、以下の図をご参照ください。

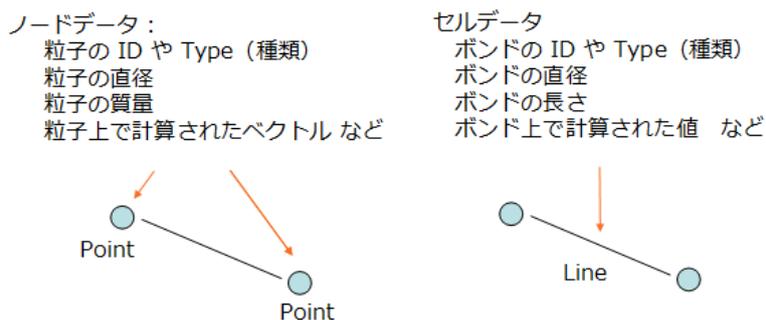


図 2 ノードデータとセルデータ

粒子の ID や Type をノードデータとして与えておくことで、可視化時に、その粒子の ID や Type で色づけしたり、また、その Type の値の範囲等から、一部の粒子だけを表示上、抜き出す (抽出) ができるようになります。

同様に、セルデータは、Line 要素に対して定義でき、例えば、そのラインの長さといったデータを持たせると、分子間の距離で色づけするようなこともできます。

UCD データでは、以上のように、座標値とセルのタイプ、セルの構成を示す接続リストでその形状を表現し、データの定義点やセルに対して、ノードデータやセルデータを与えます。

次のページに、実際のアスキーフォーマットのサンプルデータ（拡張子 `.inp`）を示していますので、あわせてご参照ください。

このサンプルには、以下の図のように、座標値は6点、また、要素が計10個（Point が6個、Line が4本）あります。

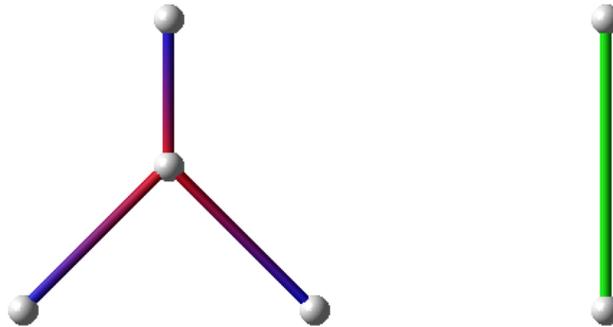


図 3 UCD サンプル

次のページに、実際のアスキーフォーマットのサンプルデータ（拡張子 `.inp`）を示していますので、あわせてご参照ください。

各座標値の先頭にはノード ID を、また、各セル（要素）の先頭にはセル ID を定義している点にご注意ください。これらの ID をセルの接続リストやノードデータやセルデータの対応する ID として利用しています。

その他、各セルには、材料 ID という ID も設定しています。この ID が設定されている場合、AVS/Express 上で、セルが区別して管理されます。構造解析のような面データに対しては有用な場合がありますが、ここでは利用しません。すべて同じ値を設定してください。

ノードデータやセルデータは、最初にそのデータ個数を定義しています。データの内訳の部分（2 1 1 の行）にご注意ください。1つのデータに、複数成分のベクトルデータを持たせることもできます。この例では、2つのデータは、スカラーデータが1つずつのため、2 1 1 と記述しています。例えば、ベクトルデータがある場合、2 3 1 のように記述し、1つ目のデータが3成分のデータであることを示すといった記述も可能です。実際のデータの定義では、4つのデータが並ぶこととなります。

※ フォーマットの詳細は、AVS/Express ユーザーズガイドをご参照ください。

なお、データの名前は、AVS/Express 上の UI（ユーザーインターフェース）上に表示される名前で、日本語や特殊文字は不可です。単位名は利用していません（同様に日本語等は不可）。

# AVS UCD File → 最初の # はコメントです。

1 → ステップ数 : 複数ステップの場合 **step1** 以降を繰り返します。

data\_geom → 複数ステップの変化方法を指定します。

**step1** → ステップ 1 つ目のデータの区切り文字です。

6 10 → ノードが 6 点、セルが 10 個あります。

0 1.000000 1.000000 0.000000 → ノード ID、X 座標、Y 座標、Z 座標

1 0.000000 0.000000 0.000000 → (スペース区切り)

2 2.000000 0.000000 0.000000 → (6 点分並びます)

3 1.000000 2.000000 0.000000

4 4.000000 0.000000 0.000000

5 4.000000 2.000000 0.000000

0 1 pt 0 → セル ID、材料 ID (使いません)、セルの名前、接続リスト

1 1 pt 1 → Point セルは点のみですので、接続リストは 1 つです。

2 1 pt 2 → (接続リストはノード ID で指定します)

3 1 pt 3 → このデータでは、6 個の粒子があります。

4 1 pt 4

5 1 pt 5

6 1 line 0 1 → Line セルの接続リストは 2 つ必要です。

7 1 line 0 2 → ノード ID、0 と 2 を接続するボンドです。

8 1 line 0 3

9 1 line 4 5

2 1 → ノード上に 2 種、セル上に 1 種のデータ (物性値) があります。

2 1 1 → ノード上の 2 種のデータの内訳で、スカラー値が 2 つあります。

name\_00,unit\_00 → ノード上の 1 つ目のデータの名称と単位名です。

name\_01,unit\_01 → ノード上の 2 つ目のデータの名称と単位名です。

0 3.000000 1.000000 → ノード ID、1 つ目の値、2 つ目の値 (スペース区切り)

1 1.000000 2.000000 → 粒子上に定義されたデータをノードデータに変換します。

2 1.000000 3.000000

3 1.000000 4.000000

4 2.000000 5.000000

5 2.000000 6.000000

1 1 → セル上の 1 種のデータの内訳でスカラー値が 1 つあります。

c\_name\_00,c\_unit\_00 → セルデータの名称と単位名です。

0 0.000000 → セル ID、1 つ目の値 (スペース区切り)

1 0.000000 → 0 ~ 5 番までは粒子の Point セルです。

2 0.000000 → 粒子上のデータはノードデータで表現していますので、

3 0.000000 → ここでは 0 としています。

4 0.000000

5 0.000000 → ここまでが Point セルのデータです。

6 10.000000 → ここ以降が Line セルのデータです。

7 20.000000 → ボンド上に定義されたデータをセルデータに変換します。

8 50.000000

9 20.000000

複数ステップの時間変換のデータを記述することもできます。

複数ステップがある場合、最初にステップ総数を記述します。

step1 の記述が終わった後に（前ページの最後の行が終わった後に続けて）、step2 と記述し、以降、次のステップのデータを記述します。

複数ステップの変化方法には、座標値とデータの両方が変化する `data_geom` と、座標値のみが変化する `geom`、データ値のみが変化する `data` の3つの方法を指定できます。

`geom` や `data` の場合には、step2 以降は、その変化する部分のみを記述します。

複数ステップの場合、連番ファイル名を利用し、ファイルを分けて作っても構いません。例えば、以下のように、何らかのルールで、複数のファイルを作成します。

```
data.001.inp
data.002.inp
data.003.inp
.....                (桁数なども適宜)
```

AVS/Express 上でこれらのファイルを順番に読み替えて可視化することができます。各ファイルは、すべて、ステップ数は1、step1 のファイルとして作成します。

1つのファイルに複数ステップの記述を行うよりも、各ファイルを読み込んで可視化して見ることができますので、何らかエラー等が発生した際のデバッグは楽になります。

その他、UCD データには、バイナリフォーマットもあります。

本書では詳細は説明しませんが、以下のようなコントロールファイル（アスキー）と実際のデータを記述したバイナリファイルを作成します。

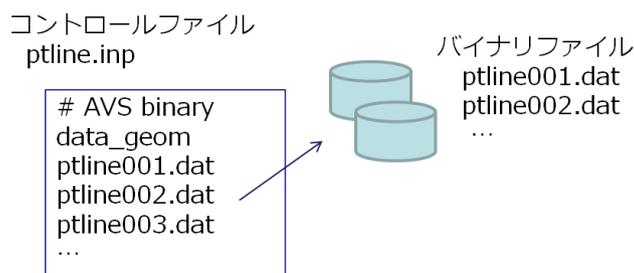


図 4 UCD バイナリファイル

データ総数や容量が大きくなる場合には、バイナリフォーマットをご利用ください。

詳細は後述しますが、UCD データを読み込むには、Read\_UCD モジュールを利用します。

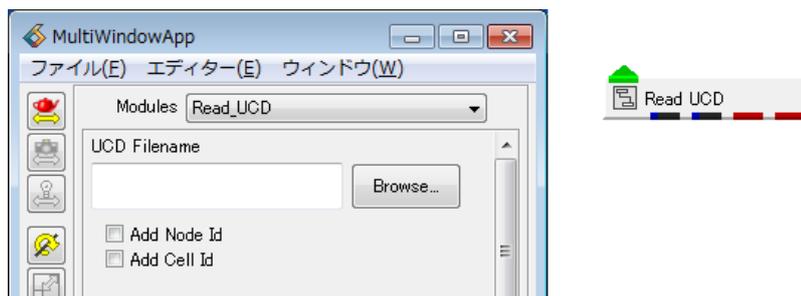


図 5 Read\_UCD モジュール

作成した .inp ファイルを Browse ボタンを押して指定します。

まずは、次図に示すように直接ビューワー・モジュールと接続し、データを読み込めているかどうかをご確認ください。次の図は、先のサンプルデータを読み込んだ例です。

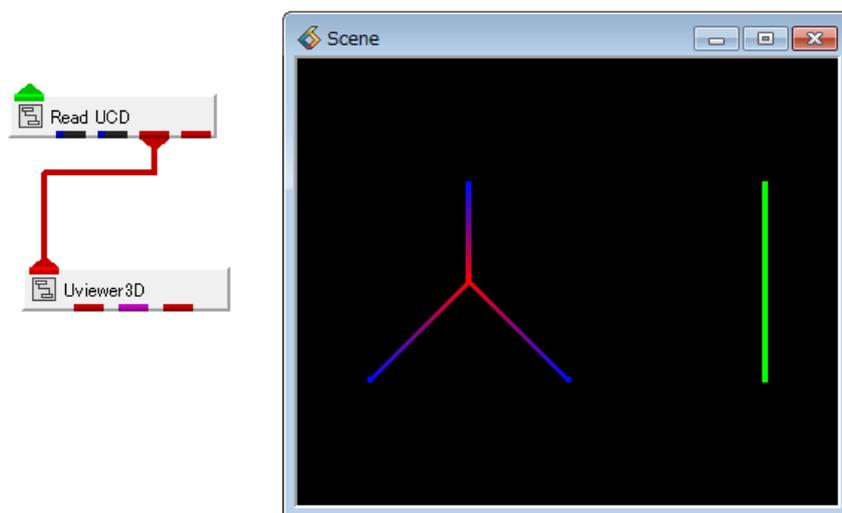


図 6 Read\_UCD の確認

デフォルトでは、点はピクセルで表現されますので、画面上ではよくわかりません。Line セルが 4 個あるのを見ることができます。

## 2.3 Field データ（離散点）と読み込み

単に粒子の動きのみを可視化したいような場合、Field データの 1 次元データを利用することもできます。

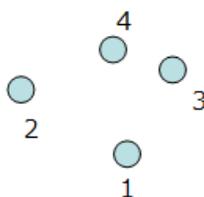


図 7 離散点

例えば、上図の例では、1 から 4 個の点が分布しています。このような離散点は Field データを使って読み込むことができます。

以下にサンプルデータを示します。

Field データの場合、データがどのように並んでいるかを記述することで、既存のデータに手を加えずに、データを読み込むことができます（読み込み方を記述できる場合に限りです）。

```
X Y Z DATA1 DATA2    → ヘッダー行があることを想定
0.0 0.0 0.0 1.0 10.0  → このデータでは X Y Z DATA1 DATA2 を想定
1.0 0.0 0.0 2.0 20.0
2.0 0.0 0.0 3.0 30.0
1.0 1.0 0.0 4.0 40.0
```

例えば、上記のデータファイル（atom.dat）があった場合、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。

```
# AVS                → 1 行目の #半角スペース AVS は必須
#                   → 2 行目以降の # はコメント
ndim = 1             → 1 次元配列で表現できるデータ
dim1 = 4             → 配列の大きさ（点が 4 個）
nspace = 3           → 3 次元空間（xyz）に分布
veclen = 2           → データ成分は 2 個
data = float         → データのタイプは float 型（double 型もあり）
field = irregular    → 離散点の場合は任意の並びを示す irregular を指定
label = DATA1 DATA2 → データ成分 2 個の名前（日本語不可、スペース区切り）
variable 1 file=./atom.dat filetype=ascii skip = 1 offset=3 stride=5 → DATA1
variable 2 file=./atom.dat filetype=ascii skip = 1 offset=4 stride=5 → DATA2
coord 1 file=./atom.dat filetype=ascii skip = 1 offset=0 stride=5 → X 座標
coord 2 file=./atom.dat filetype=ascii skip = 1 offset=1 stride=5 → Y 座標
coord 3 file=./atom.dat filetype=ascii skip = 1 offset=2 stride=5 → Z 座標
```

Field アスキーヘッダーファイルでは、まず、データの個数などを最初に定義します。  
次に、`variable` や `coord` 行を使って、既存のデータファイルの読み込み方法を指定します。

この例では、データが `XYZ DATA1 DATA2` と並んでいることを想定しています。

まず、`coord1` の行に着目してください。

既存のデータファイルを `file = ./atom.dat` で指定しています。

`skip` は読み飛ばし行で、ヘッダー行を読み飛ばすため、`1` を指定しています。

`offset` で読み込みの開始位置を指定します。

このデータの1カラム目が `X` ですので、`offset = 0` としています。

`stride` は、同じデータが何回毎に並んでいるかを指定するためのキーワードです。

`YZ DATA1 DATA2 X` と5個目に登場しますので、`stride=5` を指定します。

同様に `coord 2` と `coord 3` で `YZ` 座標を、`variable 1` と `2` でデータを読み込んでいます。

このように離散的な点のみの場合には、Field データを利用できます。

※ 先の UCD で Point 要素だけを使って表現することもできます。

内部構造上、使用メモリや速度的には Field の方が少なく済みますので、  
粒子のみのような場合には、Field データをご利用ください。

#### 重要)

この例では説明が簡単なように、`XYZ DATA1 ...` と1行に1つの点の情報が  
並んでいるデータを仮定し、その読み込み方法を説明しました。

この並びの場合、`stride` というキーワードを使って、各カラム毎にデータを  
読み飛ばしながら処理する必要があります。

この方法は、データの読み込み時間が一番遅い方法となります。

実際にデータを準備される場合には、すべての `X` が記述された後にすべての `Y` が  
並ぶといったように、`stride` を使わない方法をお勧めします。

もしくは、成分毎にファイルを準備 (`x.dat` や `y.dat` など) する方が  
読み込み速度を向上できます。

さらに、アスキーではなく、バイナリデータでも同じようにアスキーヘッダーで  
バイナリファイルを指定できますので、データ総数や容量が大きい場合は、  
バイナリデータをご利用ください。

※ カンマ区切りのデータはこの方法では読み込むことはできません。

Field データにも時系列に対応したフォーマットがあります。詳細はユーザーズガイドをご参照  
ください。また、複数の連番ファイルで準備することもできます。

Field データを読み込むには、Read\_Field モジュールを利用します。

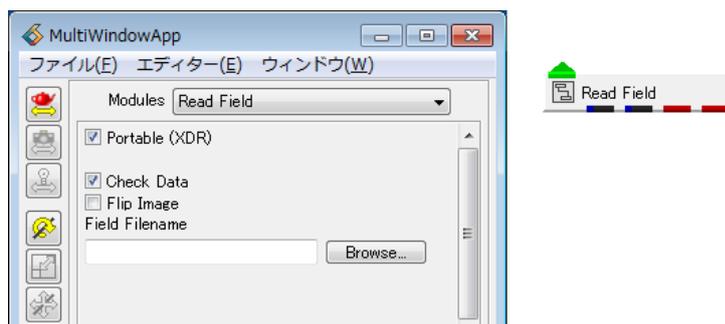


図 8 Read\_Field モジュール

作成した .fld ファイルを Browse ボタンを押して指定します。

まずは、次図に示すように直接ビューワー・モジュールと接続し、データが読み込めているかどうかをご確認ください。次の図は、先のサンプルデータを読み込んだ例です。

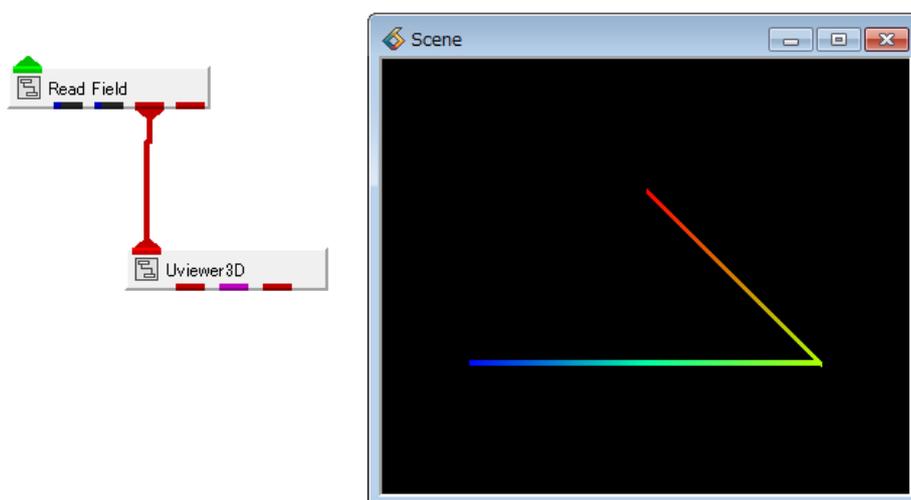


図 9 Read\_Field の確認

Field データは、配列で表現できるデータです。そのため、内部的にも配列で管理されています。この例では 4 つの点列を読み込んでいますので、直接ビューワー・モジュールに接続すると、その 4 つの点を結んだ線が表示されます。

この 4 点を粒子として表現する可視化の方法については、後半の章で説明します。

## 2.4 Field データ（3次元配列）と読み込み

電荷密度のように、3次元メッシュデータの場合には、Field の3次元配列データを利用します。以下に、アスキーヘッダーファイルの例を示します。

# AVS	→ 1行目の #半角スペース AVS は必須
#	→ 2行目以降の # はコメント
ndim = 3	→ 3次元配列で表現できるデータ
dim1 = 64	→ I 方向のメッシュ数
dim2 = 64	→ J 方向のメッシュ数
dim3 = 64	→ K 方向のメッシュ数
nspace = 3	→ 3次元空間 (xyz) に分布
veclen = 1	→ データ成分は1個
data = byte	→ データのタイプは byte 型 (float/double 型もあり)
field = uniform	→ 直交等間隔のメッシュタイプ
variable 1 file=../volume/hydrogen.dat filetype=binary skip = 3	→ 読み込み指定

このアスキーヘッダーファイルは、AVS/Express のインストールフォルダにある以下のサンプルファイルです。

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

このファイルで読み込んでいるデータファイル（実際のファイル）は、上記ヘッダーにあるように、以下のフォルダにあります。

```
c:\Program Files\AVS Express\data\volume\hydrogen.dat
```

このデータファイルはバイナリファイルで、最初に I 方向のメッシュ数、J 方向のメッシュ数、K 方向のメッシュ数がそれぞれ byte 型で入っています。よって、この 3byte を読み飛ばすため、skip = 3 を指定しています。

※ アスキーファイルの場合は、skip では、読み飛ばす行数を指定します。  
バイナリファイルの場合には、バイト数を指定します。

その後、このデータファイルには、I 方向、次に J 方向、最後に K 方向に並んだ byte 型のデータが出力されています。

データが読み込めたかどうか、以下のようなネットワークで確認できます。

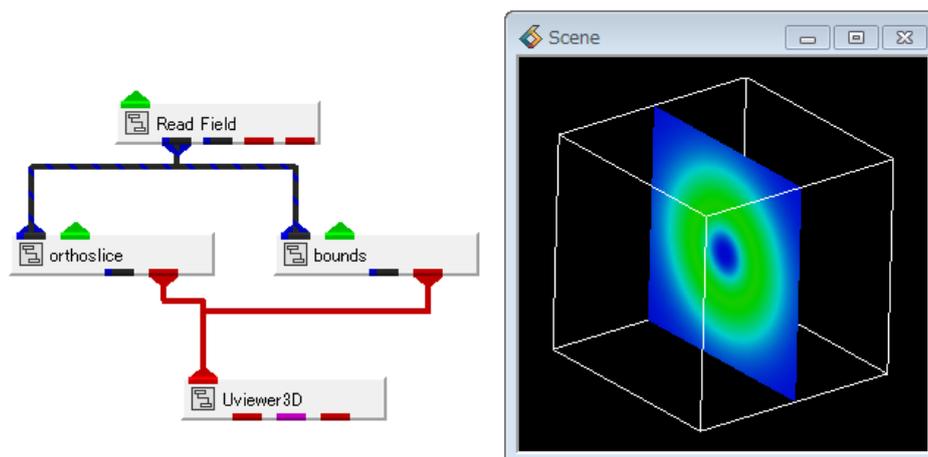


図 10 3次元 Field データの確認

Read\_Field モジュールのパラメーターで、作成した .fld ファイルを指定します。

例えば、上図では、orthoslice モジュールを使って、メッシュの断面コンター図を作成しています。orthoslice モジュールのパラメーターで、IJK の断面を移動することができます。

注) Portable(XDR) のチェック

Read\_Field モジュールのパラメーターには、Portable(XDR) というチェックがあります。



デフォルトでは XDR フォーマット (Big Endian で XDR) を読み込もうとします。

Little Endian のデータ (例えば、Windows/Linux Intel チップ上で作成されたデータ) を読み込む場合は、オフにしてください。

また、オフにすると Swap Bytes というチェックが表示されます。

ここにチェックすると、Big Endian 形式のデータとして読み込みます。

## 2.5 一般的なフォーマット

材料系のデータフォーマットとして、一般的なフォーマットにも対応したリーダー（や変換ツール）があります。

- LAMMPS
- PDB
- Gaussian
- VASP
- XYZ
- ATK netCDF
- TOMBO
- SCIGRESS

### 2.5.1 LAMMPS データの読み込み

LAMMPS（一部フォーマットを除く）ファイルに対応したリーダー・モジュールがあります。

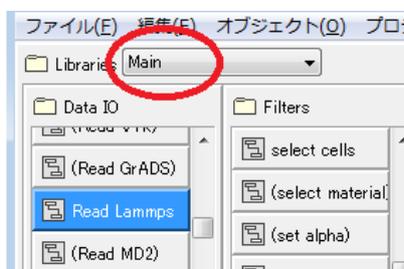


図 11 Read\_LAMMPS モジュール

Data ファイルと Dump ファイルを読み込むことができます。  
パラメーターについては、モジュールリファレンスをご参照ください。

Read\_LAMMPS モジュールは、先述の UCD データと同様、読み込んだデータを Point と Line のセルとして扱います。このモジュールを使った可視化については、後述します。

※ LAMMPS モジュールには、ユーザー定義のフォーマットなど、多くのデータがあります。  
データのフォーマット等によっては読み込めない場合もあります。

## 2.5.2 PDB データの読み込み

PDB ファイルに対応したリーダー・モジュールがあります。  
KGT ライブラリページの Data\_IO ライブラリにあります。

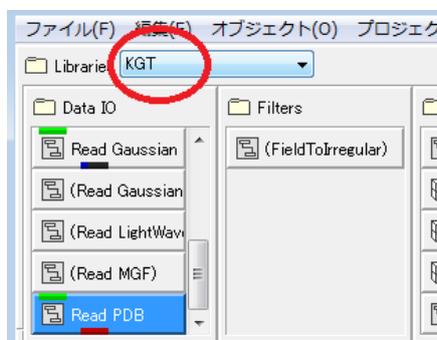


図 12 Read\_PDB モジュール

※ KGT ライブラリは `-mavs` オプションをつけて起動した際に表示される追加のライブラリです。  
(ヘルプページにある KGT ライブラリについてのドキュメントをあわせてご参照ください)

このモジュールは、読み込み時に指定したパラメーターによって、データの表示のみを行うことができる機能です。

UCD データや Lammps データのように、読み込んだデータに対して、各モジュールを使ってフィルタリング処理などを行うことはできません。

このモジュールには、赤色の出力ポートのみがあります。直接ビューワー・モジュールと接続します。

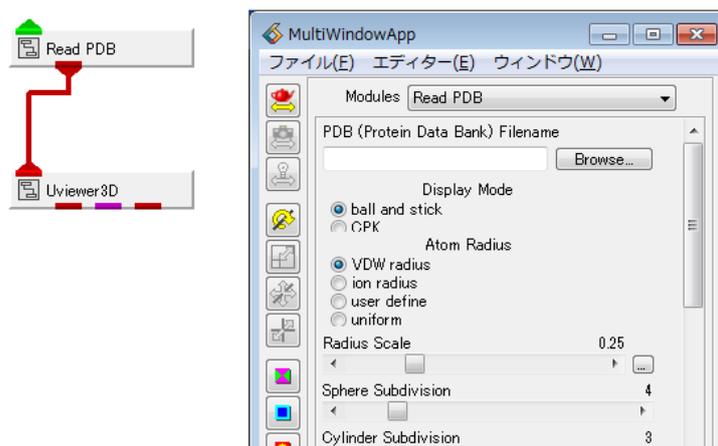


図 13 Read\_PDB モジュールのパラメーター

パラメーターに、**Browse** ボタンがありますので、このボタンをクリックし、PDB ファイルを指定します。

他の一般的なデータの読み込みモジュールと異なり、可視化のためのパラメーターもその下に並んでいます。

表示方法（ボール&スティックか CPK）の選択や球の大きさ（Scale）など、各パラメーターを操作してみてください。

※ 各パラメーターについては、オンラインヘルプからアクセスできる、KGT ライブラリについてのドキュメントをご参照ください。

※ ハードウェアレンダラかソフトウェアレンダラかによって、表現方法が異なります。

※ ハードウェアレンダラの場合、ポリゴン分割された球を作成します。

**Sphere Subdivision** パラメーターを使って、その分割数を指定できます。

（値を大きくすると、滑らかな球になりますが、パフォーマンスに影響します）

チューブも **Cylinder Subdivision** を使って、滑らかさを変更できます。

ソフトウェアレンダラの場合、点と半径で表現されるソフトウェア球で表示され、きれいな球となります。

ただし、ソフトウェアレンダラの場合には、スティックはライン表示のみとなります。

CPK 表示（球のみ）を行う場合は、ソフトウェアレンダラの方が便利です。

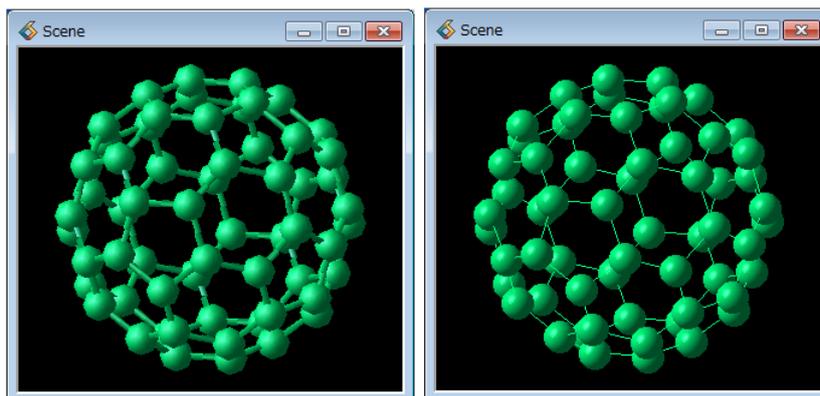


図 14 ポリゴン球とソフトウェア球

補) ハードウェアレンダラとソフトウェアレンダラ

AVS/Express には、2つのレンダラがサポートされています。

ハードウェアレンダラは、OpenGL（とグラフィックスボード）を使ってレンダリングする機能で、高機能のグラフィックスがサポートされている場合、高速に処理できます。

ソフトウェアレンダラは、AVS 独自で、CPU で処理をする機能で、OpenGL は利用しません。

この切り替えは、以下のアイコン（黄色がハードウェアレンダラ）で行うことができます。



（オンオフで切り替え）

本文中にあるように、特に球の表示で大きな違いがあります。

### 2.5.3 Gaussian データの読み込み (log ファイル)

Gaussian の log ファイルに対応したリーダー・モジュールがあります。

- CCMS.Gaussian.Read\_Gaussian\_Log\_Ang
- CCMS.Common.Read\_XYZ
- CCMS.Gaussian.Gaussian\_Vibration

※ CCMS ライブラリは `-mavs` オプションをつけて起動した際に表示される追加のライブラリです。  
東北大学 金属材料研究所殿の成果を製品化させていただいたものです。  
(ヘルプページにある CCMS ライブラリについてのドキュメントをあわせてご参照ください)

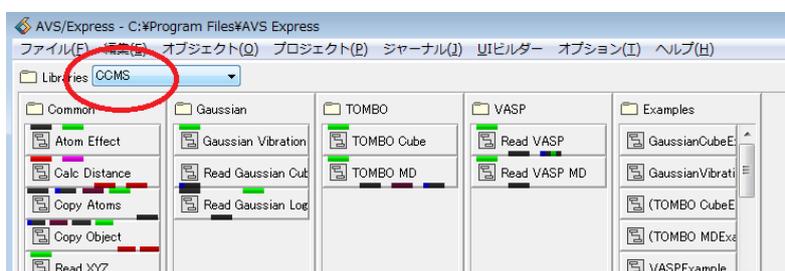


図 15 CCMS ライブラリページ

以下のようにモジュールを接続します。

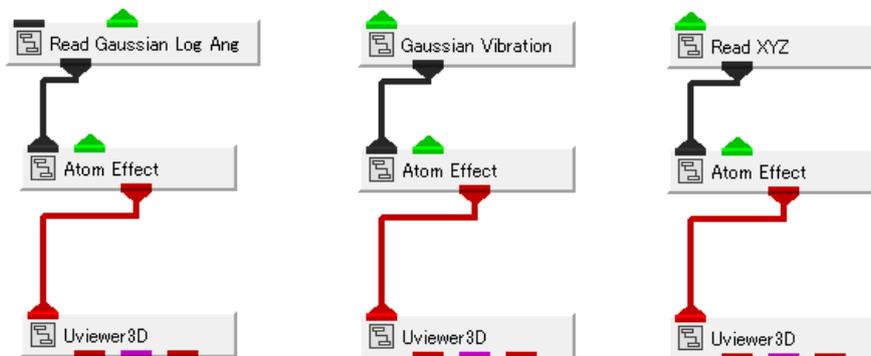


図 16 Gaussian log ファイルの読み込み

いずれのモジュールもデータの指定方法は同じです。  
モジュールのパラメーターに、ファイル名を指定する **Browse** ボタンが表示されますので、**Gaussian log** ファイルを読み込んでみてください。

- ※ **Read\_XYZ** モジュールを利用する場合は、ファイルのタイプの指定がありますので、**Gaussian** を選んでください。  
また、データの中に、**Vibration** データが含まれている場合は、**Gaussian Vibration** モジュールを利用してみてください。  
このモジュールには、**Gaussian Log** データの読み込みに加えて、下図のように

Vibration データの表示のためのパラメーターが追加されています。  
 Vibration データのグラフやアニメーションの再生などを行うことができます。  
 (詳細は CCMS ライブラリに関するマニュアルをご参照ください)

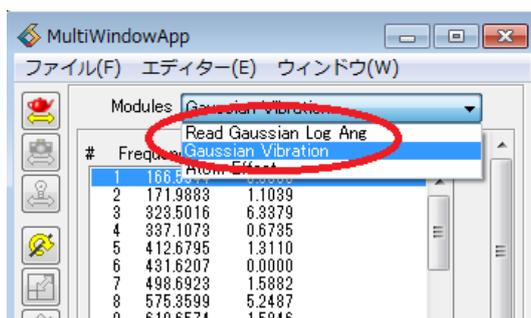


図 17 Vibration データの表示

AtomEffect モジュールは、読み込んだデータをボールやスティックで表示するモジュールです。  
 表示方法 (ボール&スティックか CPK) の選択や球の大きさ (Scale) など、  
 各パラメーターを操作してみてください。

※ Atom\_Effect モジュールの利用に関する注意点：

隣接する 2つの原子半径の合計×Distance Coef. (パラメーターで指定) が  
 Maximum Distance (パラメーターで指定) よりも小さい場合に、原子間のスティックを  
 作成します。

データによってはスティックが作成されない場合がありますので、適宜、これらの  
 パラメーターを調整してみてください。

※ 各パラメーターについては、オンラインヘルプからアクセスできる、  
 CCMS ライブラリについてのドキュメントを参照してください。

※ このモジュールは、レンダラによって、球の表現方法が異なります。  
 ハードウェアレンダラの場合、ポリゴンで分割された球を作成します。  
 Sphere Subdivision パラメーターを使って、その分割数を指定できます。  
 (値を大きくすると滑らかな球になりますが、パフォーマンスに影響します)  
 ソフトウェアレンダラの場合、ソフトウェア球で表示され、きれいな球となります。

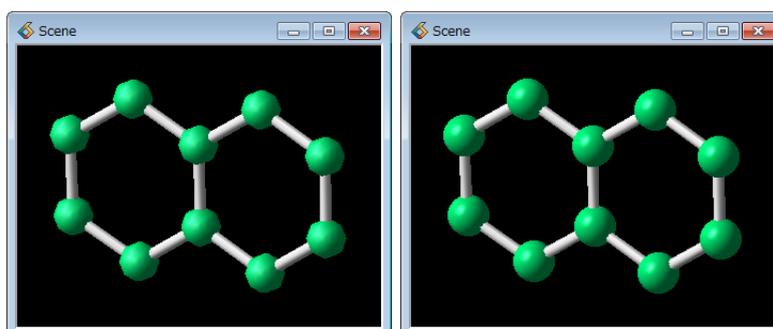


図 18 ポリゴン球とソフトウェア球 (Gaussian)

PDB の可視化と異なり、Gaussian の可視化の AtomEffect モジュールでは、  
 ソフトウェア球の場合にも、スティックはチューブ形状のまま、表示できます。

## 2.5.4 Gaussian データの読み込み (Cube ファイル)

Gaussian の cube ファイルに対応したリーダー・モジュールもあります。

- CCMS.Gaussian.Read\_Gaussian\_Cube\_Ang

以下のようにモジュールを接続します。

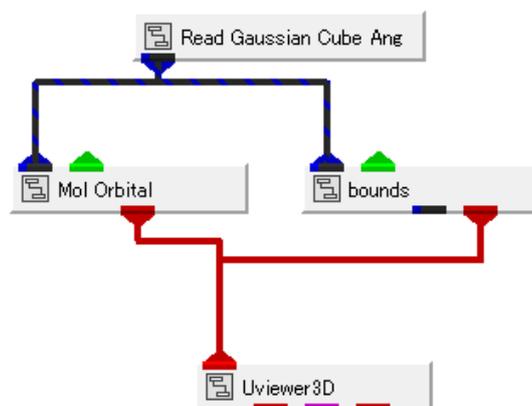


図 19 Gaussian Cube ファイルの読み込み

Mol\_Orbital モジュールは、KGT ライブラリページにあるモジュールです。下図にあるように、電荷密度の等値面表示を行います。

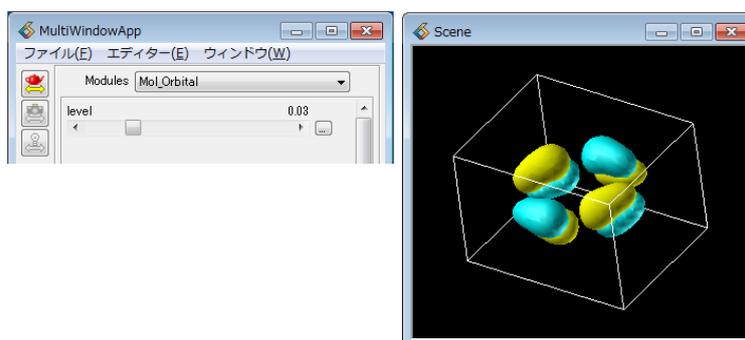


図 20 Mol\_Orbital モジュールによる表示

Mol\_Orbital モジュールは、標準の isosurface モジュールと同様、等値面を作成しています。電荷密度の正負の値に対して、等値面を作成しています。また、色は予め設定された黄色（マイナス）と水色（プラス）で表示しています。（isosurface モジュールを 2 個利用し、しきい値をそれぞれ、プラスとマイナスの値に設定すると、同じ結果を得ることができます）

## 2.5.5 VASP データの読み込み

VASP データの読み込みモジュールは、Gaussian 同様、CCMS ライブラリにあります。

- CCMS.VASP.Read\_VASP
- CCMS.VASP.Read\_VASP\_MD

Read\_VASP モジュールは、CHGCAR、POTCAR ファイルを読み込みます。

以下のようにモジュールを接続します。

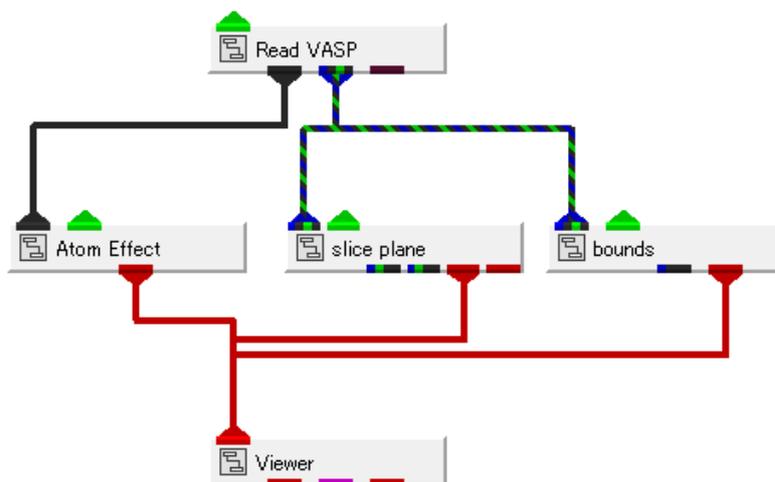


図 21 Read\_VASP モジュール

Read\_VASP モジュールは、2つの出力ポートを持っています。左側の出力ポートから原子情報のパラメータが出力されます。

Gaussian データの読み込みで述べた Atom Effect モジュールを利用して、ボール&スティック表示を行うことができます。

※ Atom\_Effect モジュールの利用に関する注意点：

隣接する2つの原子半径の合計×Distance Coef. (パラメーターで指定) が Maximum Distance (パラメーターで指定) よりも小さい場合に、原子間のスティックを作成します。

データによってはスティックが作成されない場合がありますので、適宜、これらのパラメーターを調整してみてください。

中央の出力ポートからは電荷密度のフィールドデータが出力されます。例えば、上図では、標準の slice\_plane モジュールを使って、断面でのコンター図を作成しています。

Read\_VASP\_MD モジュールは、XDATCAR、POTCAR、POSCAR (または CONTCAR) ファイルを読み込み、指定されたステップの原子情報を出力するモジュールです。

Read\_VASP モジュール同様、Atom Effect モジュールを利用して、ボール&スティック表示を行います。

また、XYZ の領域 (ボックス) を表示するには、CCMS ライブラリの Examples にある VASP\_MDExample アプリケーションを利用してください。

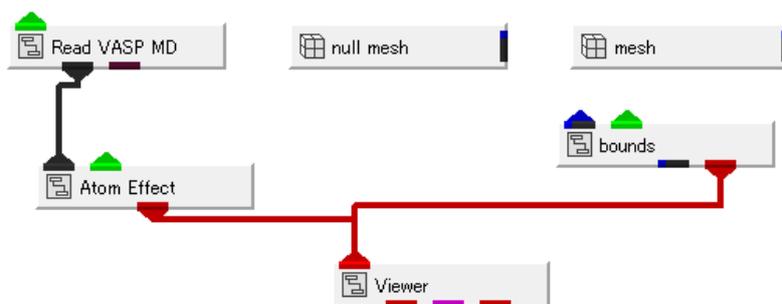


図 22 VASP\_MDExample

### 2.5.6 XYZ データの読み込み

以下のフォーマットで表現される一般的な XYZ ファイルフォーマットに対応しています。

```

<number of atoms>
comment line
atom_symbol1 x-coord1 y-coord1 z-coord1
atom_symbol2 x-coord2 y-coord2 z-coord2
...
atom_symboln x-coordn y-coordn z-coordn

```

ただし、上記のセットを繰り返したアニメーション・データには対応していません。  
連番ファイルなどに分割してください。

この XYZ データは、CCMS ライブラリの Read\_XYZ モジュールで読み込むことができます。

- CCMS.Common.Read\_XYZ

以下にネットワークの例を示します。

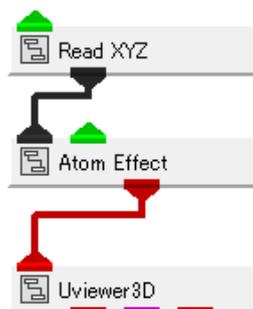


図 23 XYZ データの読み込み

「図 16 Gaussian log ファイルの読み込みに述べたモジュールと同じです。  
モジュールのパラメーターにある File Type を XYZ Standard にして、XYZ ファイルを  
指定してください。  
Atom\_Effect モジュールとペアで利用します。

※ Atom\_Effect モジュールの利用に関する注意点：

隣接する2つの原子半径の合計×Distance Coef. (パラメーターで指定) が Maximum Distance (パラメーターで指定) よりも小さい場合に、原子間のスティックを作成します。

データによってはスティックが作成されない場合がありますので、適宜、これらのパラメーターを調整してみてください。

### 2.5.7 ATK netCDF データの読み込み

ATKで作成された netCDF ファイルを読み込むことができます。

以下のモジュールを利用します。

- CCMS.TOMBO.TOMBO\_Cube

電荷密度分布など、構造データの可視化を行うことができます。

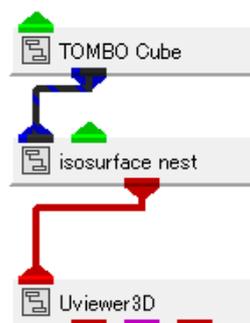


図 24 TOMBO\_Cube モジュール

※ ATK の XYZ ファイルと netCDF ファイルはそのスケールの情報がないため、同時に重ねて表示することはできません。

### 2.5.8 TOMBO データの読み込み

TOMBO データを読み込むには、以下のモジュールを利用します。

- CCMS.Common.Read\_XYZ
- CCMS.TOMBO.TOMBO\_Cube
- CCMS.TOMBO.TOMBO\_MD

Read\_XYZ モジュールは、TOMBO の原子情報ファイル(log.out)を読み込むモジュールです。読み込んだデータに対して Atom\_Effect モジュールを使って、ボール&スティック表示を行うことができます。

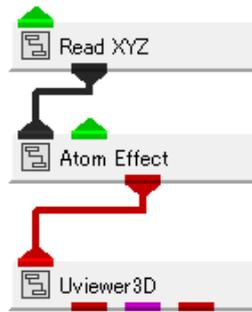


図 25 XYZ データの読み込み

※ **Atom\_Effect** モジュールの利用に関する注意点：

隣接する 2 つの原子半径の合計 × Distance Coef. (パラメーターで指定) が **Maximum Distance** (パラメーターで指定) よりも小さい場合に、原子間のスティックを作成します。

データによってはスティックが作成されない場合がありますので、適宜、これらのパラメーターを調整してみてください。

**TOMBO\_Cube** モジュールは、**TOMBO** の電荷密度ファイル (avs.out) を読み込むモジュールです。

例えば、下図のネットワークでは電荷密度に対して複数の等値面を作成します。

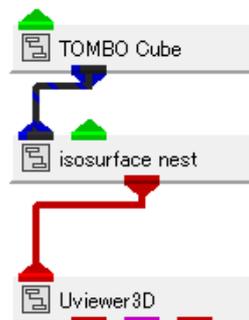


図 26 TOMBO\_Cube モジュール

**TOMBO\_MD** モジュールは、**TOMBO** が出力する計算結果 trajectory.out と atomxyz.in を読み込み、原子のアニメーション表示を行うための原子情報を出力します。

**Read\_XYZ** モジュールの出力同様、**Atom\_Effect** モジュールに接続することで、ボール&スティック表示を行います。

また、左から 3 番目のフィールド出力に、**bounds** モジュールを接続することで、ユニットセルの格子枠を表示することができます。

一番右側のポートは時刻ステップのラベル文字です。**TextTitle** と接続して利用します。

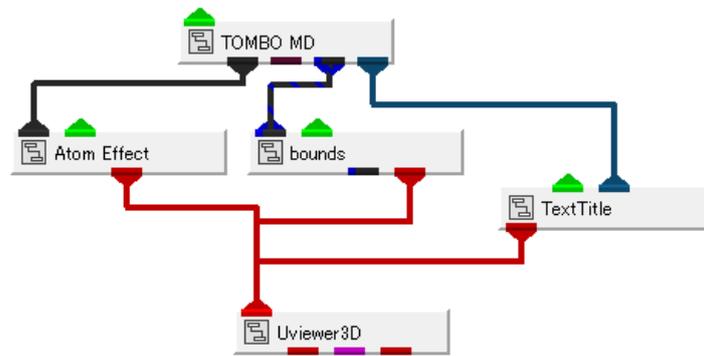


図 27 TOMBO\_MD モジュール

### 2.5.9 SCIGRESS データの読み込み

SCIGRESS の MD 出力フォーマット (.sim) ファイルに対応した変換ツールがあります。  
.sim ファイルを読み込み、UCD ファイルに変換します。

この変換ツールは、標準には含まれていません。

サポート窓口までお問い合わせいただけましたら、適宜、ツールのご提供などにて、対応させていただきます。

(サイバネットシステムの AVS サポート窓口までお問い合わせくださいませ)

### 2.5.10 OCTA データの読み込み

OCTA (COGNAC や SUSH) に関して、一部のデータではありますが、  
UDF ファイルに対応した変換ツールがあります。

この変換ツールは、標準には含まれていません。

サポート窓口までお問い合わせいただけましたら、適宜、ツールのご提供などにて、対応させていただきます。

(サイバネットシステムの AVS サポート窓口までお問い合わせくださいませ)

### 第3章 データの可視化（ボール&スティック）

可視化に利用する代表的なモジュールについて紹介します。

この章では、まずは、ボール&スティックによる表示について説明します。UCD や LAMMPS データで読み込んだ分子とその結合、また、Field で定義した粒子のみのデータの可視化を行います。

※ 「2.5 一般的なフォーマット」のデータ読み込み方法で紹介した CCMS ライブラリにある

Atom\_Effect モジュールを使って直接、ボール&スティック表示している場合は、CCMS ライブラリのドキュメントをご参照ください。

#### 3.1 ソフトウェア球の表示（UCD）

離散点に対して球を表示する代表的なモジュールに、set\_radius モジュールがあります。

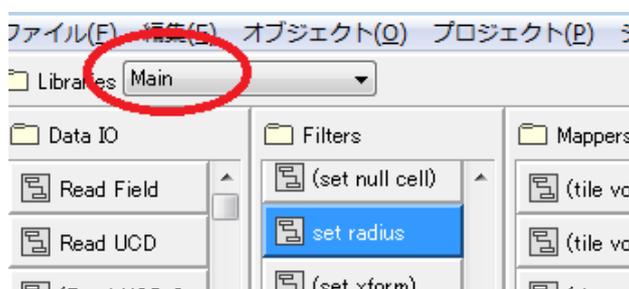


図 28 set\_radius モジュール

##### ・set\_radius モジュール

set\_radius モジュールは、ソフトウェア球を作成するモジュールです。

ソフトウェア球とは、AVS/Express のソフトウェアレンダラに対して有効な機能で、あるデータの定義点に、その点と半径の情報から球をレンダリングする機能のことを呼んでいます。

以下のネットワークを作成してみてください。

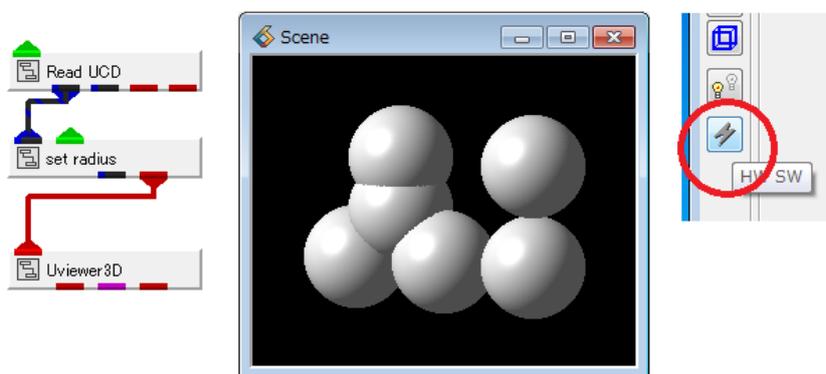


図 29 set\_radius モジュールとレンダラ

図の例では、データの読み込みで説明した、「図 6 Read\_UCD の確認と同じデータを読み込んでいます。

この絵を作成するには、レンダラをソフトウェアレンダラに変更します。

コントロールパネルにあるレンダラの切り替えアイコン（上図右）をオフにしてください。

set\_radius モジュールには、以下のパラメーターがあります。

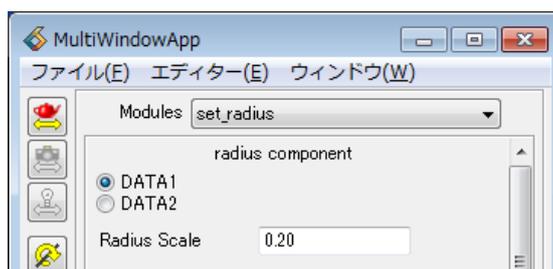


図 30 set\_radius モジュールのパラメーター

その球を作成するデータ成分の選択と、その球の半径に対するスケール値を指定するパラメーターがあります。

まず、スケール値の指定方法によって、以下の2つの違いがあります。

- ・ 正の値の入力（例 0.2）
- ・ 負の値の入力（例 -0.2）

正の値を入力した場合、選択したデータ成分の大きさによって、球の大きさを変更します。例えば、そのデータ成分が 1 ~ 10 の場合、10 の値を持つ点は大きな球で、1 の値を持つ点は小さな球で表示します。

一方、負の値を入力した場合、データ成分の大きさに関わらず、すべてのデータ定義点に同じ大きさの球を表示します。

「図 29 set\_radius モジュールとレンダラの例では、負の値を設定しているため、すべての球が同じ大きさとなっています。

次に、set\_radius モジュールの出力は、デフォルトでは白一色で表示されます。  
(陰影によって白からグレーで表現されています)

- ※ 単一色で表現し、その色を変更するには、オブジェクト・エディターにある、形状に対する色づけを利用します。  
「第6章 色、各属性の変更」をご参照ください。

ノードデータに物理量を持っている場合、球をそのデータ成分の大きさで色づけすることができます。

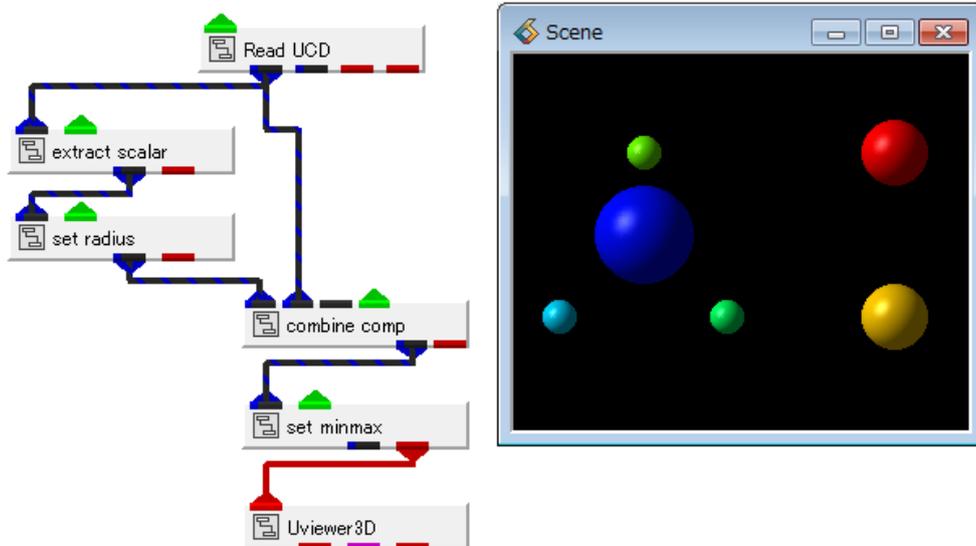


図 31 UCD データの球表示

以下のモジュールを追加しています。

- `extract_scalar` モジュール
- `set_minmax` モジュール
- `combine_comp` モジュール

以下の処理（とパラメーターの設定）を行っています。

- `extract_scalar` モジュール  
 複数のデータを持っている場合、`extract_scalar` モジュールはそのデータ成分の選択に利用できます。  
 この例のデータでは、`name_00` と `name_01` の2つの成分を持っています。

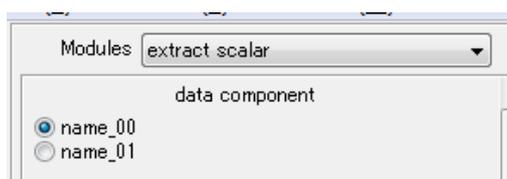


図 32 スカラー値の選択

`extract_scalar` モジュールから `set_radius` モジュールに接続していますので、ここでは、球の半径を変更するための成分を選択しています。  
`name_00` の値の大きさに応じて、球の半径を変化させます。

- ※ `set_radius` モジュールに直接、接続し、`set_radius` モジュールのパラメーターで成分を選択しても同じです。  
 ここでは、モジュールの説明のため、また、明示的にスカラー値を選択する方法として追加しています。

- **set\_radius** モジュール

先に述べたソフトウェア球を作成するモジュールです。

**extract\_scalar** モジュールで選択した成分（すでに選択しているため、このパラメーターでは表示のみとなっています）に応じて、球の大きさを変化させるために、

**radius** パラメーターに正の値のスケールを設定しています。

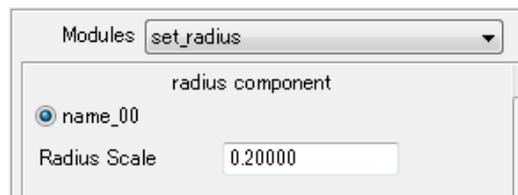


図 33 set\_radius のパラメーター

- **combine\_comp** モジュール

**set\_radius** モジュールに色づけするには、特殊なモジュールの接続を行います。

**set\_radius** モジュールの出力には色なしの点と半径情報だけが出力されています。

この球の出力に載せたい色のデータ成分を合成します。

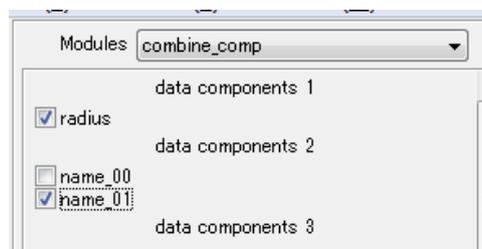


図 34 色データの合成

**data components 2** には、**Read\_UCD** モジュールの接続に含まれる、2つのデータ成分名、**name\_00** と **name\_01** が表示されます。

色づけのために合成したい成分として、上図の例では、**name\_01** を選択しています。

この選択によって、球の大きさは **name\_00** のデータで変化し、球の色は **name\_01** で変化しています。

球の大きさも色も同じ **name\_00** で変化させたい（大きな球は赤で、小さな球は青）場合は、この上図の **name\_01** をオフにして、**name\_00** をオンにしてください。

- **set\_minmax** モジュール

**set\_minmax** モジュールは、色（デフォルトでは青から赤）と数値の対応を設定するモジュールです。

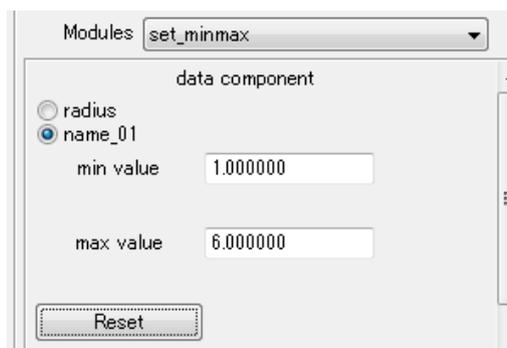


図 35 set\_minmax モジュール

まず、`data_component` に変数名が表示されていますので、色づけの成分である、`name_01` を選択してください。

デフォルトでは、このデータが持つ `min/max` 値が表示されます。

(Reset ボタンを押すと、データの `min/max` 表示に戻ります)

このデータには、1 ~ 6 の値があります。

1 が青で、6 が赤、その間は、Hue の青から赤のグラデーションの中から、該当する場所の色が割り当てられています。

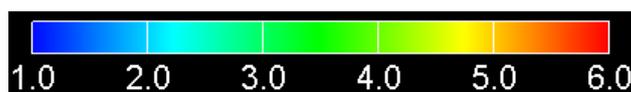


図 36 色の割り当て

※ デフォルト以外のカラーマップの利用や凡例の表示については、「第6章 色、各属性の変更」をご参照ください。

例えば、この `max` 値に 4 を設定すると、4 より大きな値はすべて赤で表現され、図のように、3つの球が赤で表現されるようになります。

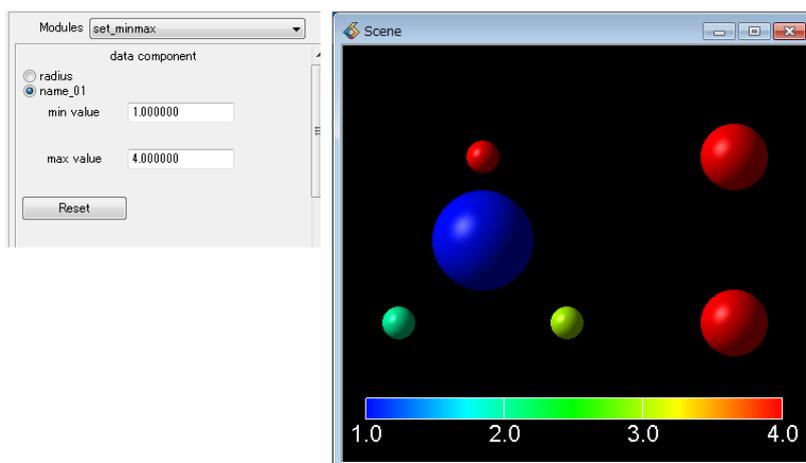


図 37 min/max の変更

### 3.2 ソフトウェア球の表示 (Field や Lammps)

set\_radius モジュールは、Field データや Lammps データなど、球を表現したいデータに対して、同じように利用できます。

下図は、「2.3 に述べた点のデータを同じように可視化した例です。データの読み込みモジュール、Read\_Field モジュール以外は同じです。

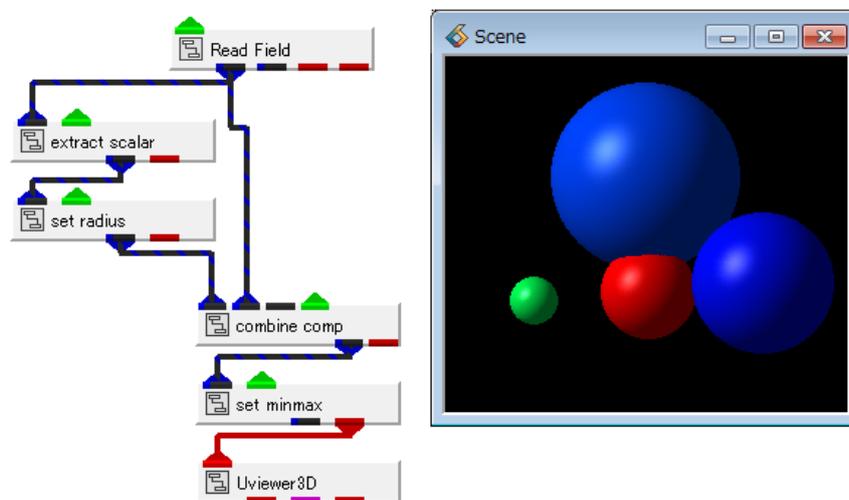


図 38 Field データの例

同じように、Read\_Lammps モジュールの出力に対しても利用できます。

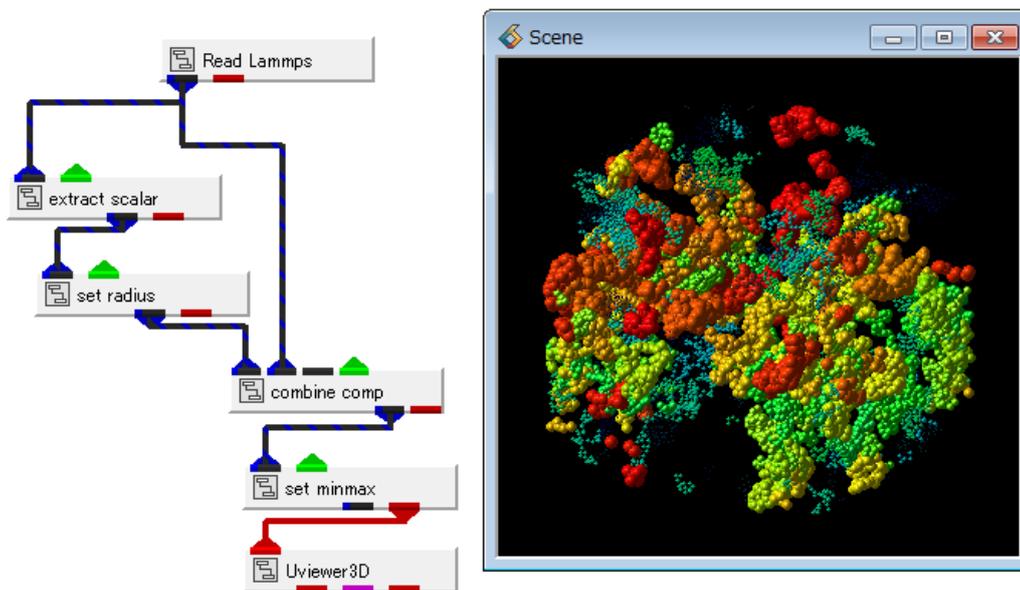


図 39 Lammps データの例

### 3.3 ポリゴン分割球

前節のきれいな球は、ソフトウェアレンダラでのみ有効です。

ハードウェアレンダリングを行うと、次図のように、ポリゴンで分割された球となります。

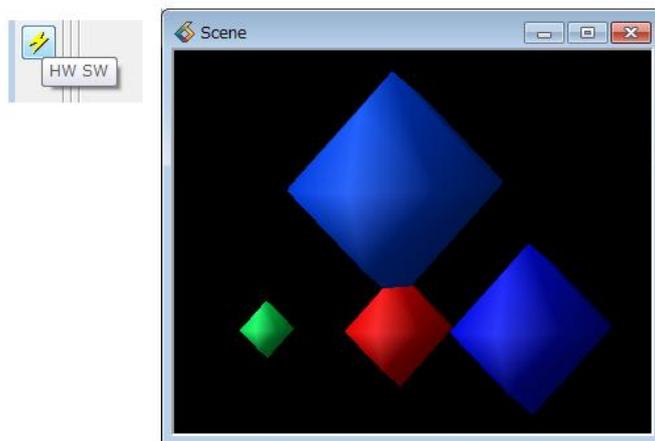


図 40 ポリゴン分割球

デフォルトでは、24 個の三角形で分割された球で表現されます。

set\_radius モジュールで作成された球の分割精度を上げるには、以下のメニューから行います。

#### 1) Select Object を使って、カレント・オブジェクトを選択

まず、Select Object のボタンをクリックし、オブジェクトを選択します。

前節の方法で球を作成している場合、set\_minmax が球のオブジェクトとなります。

Object Selector で set\_minmax を選び、Apply ボタンをクリックします。

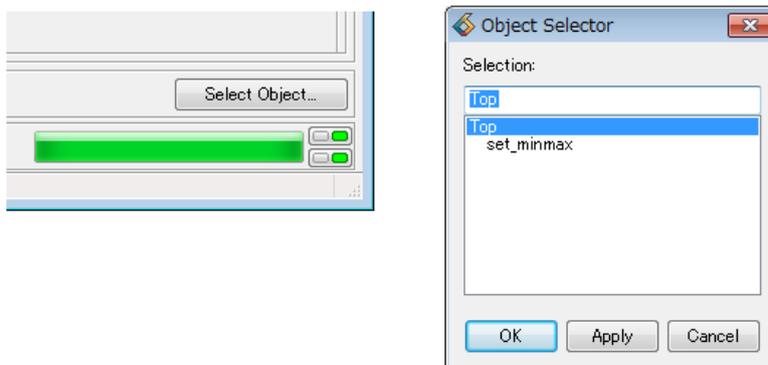


図 41 カレント・オブジェクトの設定

#### 2) オブジェクト・エディターから精度を設定

次に、エディター・メニューからオブジェクトを選びます。

さらに、開いたメニューで、Object メニューから Properties を選択し、

Type メニューから Point/Line を選びます（下図）。

この中にある Subdivision が球の分割数を変更するプロパティです。

この値を少しずつ、増やしてみてください。

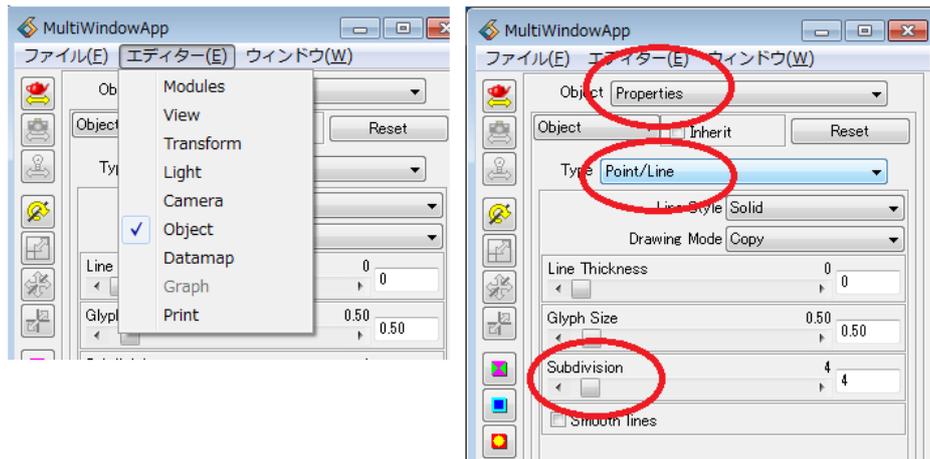


図 42 オブジェクトの Subdivision プロパティ

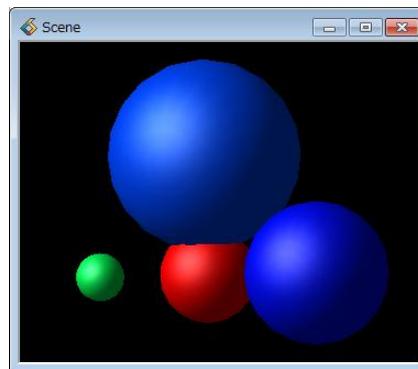


図 43 分割数を増加した例

### 3) カレント・オブジェクトを Top に設定

精度を上げたら、1) で設定したカレント・オブジェクトを、Top の全体が選択されている状態に忘れずに戻してください。

ポリゴン分割球の精度を上げると、球がきれいな形状に近づきますが、パフォーマンスに与える影響が大きくなります。上記操作を行う際には、パフォーマンスを確認しながら、少しずつ変更してみてください。

また、ご利用のグラフィックスボードのメモリなど、マシン環境によっては、AVS/Express がクラッシュ、または、グラフィックスのメモリ不足でグラフィックス画面の表示そのものが一端停止（して再開）するような場合もありますので、ご注意ください。

この方法は、直接レンダラで分割球を作成しています。そのため、ポリゴン情報をデータとして取得することはできません。例えば STL や PLY などの形状ファイルなどに出だし、3D プリンタへ送りたい場合などには、次節で説明する glyph モジュールを利用してください。

### 3.4 ポリゴン分割球 (glyph モジュール)

その他、球を表現する方法として、glyph (グリフ) モジュールを利用した可視化方法があります。

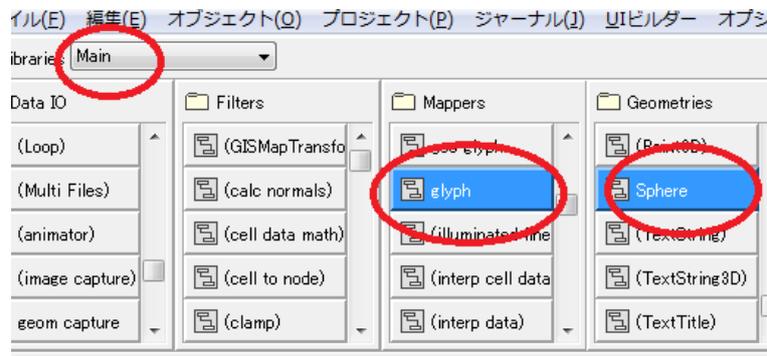


図 44 glyph と Sphere モジュール

以下のように接続します。

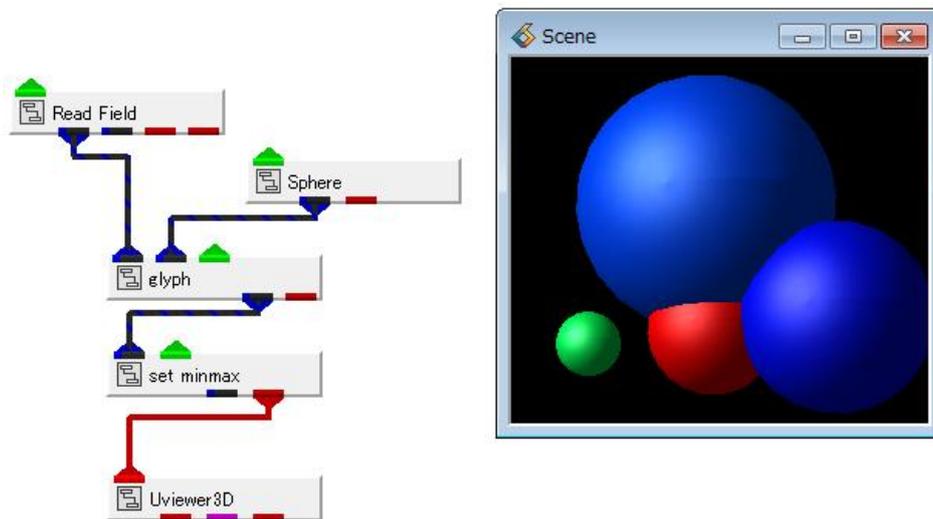


図 45 glyph モジュールの接続

glyph モジュールには、2つの入力ポートがあります。

左側にデータを接続し、右側に形状 (ここでは Sphere モジュール) を接続します。

#### • glyph モジュール

glyph モジュールは、データの定義点に、ある形状 (グリフと呼んでいます) を作成することで、データを表現するモジュールです。

その形状の大きさや色を変化させることができます。

以下のパラメーターを設定しています。

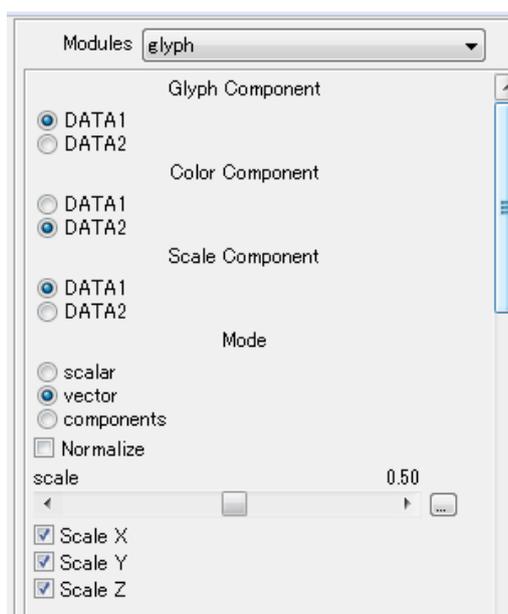


図 46 glyph モジュールのパラメーター

Glyph Component から、グリフの大きさを変化させるデータ成分を選択します。このデータには、2種類の DATA1 と DATA2 があり、DATA1 をグリフの大きさを変化させるデータとして選択しています。

Color Component から、グリフの色を変化させるデータ成分を選択します。先の例と同様、DATA1 で大きさを変化させ、DATA2 で色を変化させています。

また、Normalize のチェックと scale スライダーは、大きさに関する設定です。Normalize をオンにすると、すべてのグリフは同じ大きさで表現されます。

(Glyph Component の大きさに関わらず)  
scale は、大きさに対する倍率の調整値です。

#### • Sphere モジュール

Sphere モジュール分割球を作成するモジュールです。以下のパラメーターがあります。

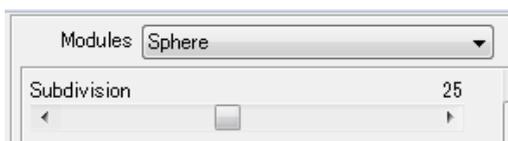


図 47 Sphere モジュールの分割パラメーター

球をどれくらいの精度（分割数）で表現するかを指定します。

先の set\_radius モジュールのハードウェアレンダラによる分割と同様、表示の状況を見ながら、少しずつ変化させてみてください。

(set\_radius モジュールの分割とは分割方法が異なりますので、同じにはなりません)

#### • set\_minmax モジュール

set\_minmax モジュールは、先にも述べたように、数値と色の対応を設定しています。

ただ、glyph モジュールと接続した場合は、以下のように、データ成分に NO NAME という

選択が表示されますので注意してください。

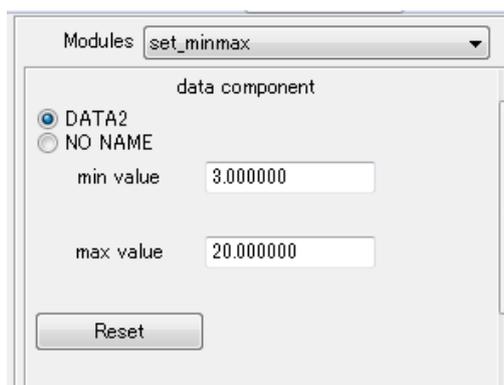


図 48 glyph に対する set\_minmax モジュールのパラメーター

1 つ目のデータ成分を選択して、適宜、色の調整を行います。

上記のように、通常、AVS/Express では、ノードデータがある場合、そのデータと色の分布を作成し、形状に色づけを行います。単色のグリフを作成したい場合は、set\_minmax モジュールを使わず、代わりに extract\_mesh モジュールを接続します。

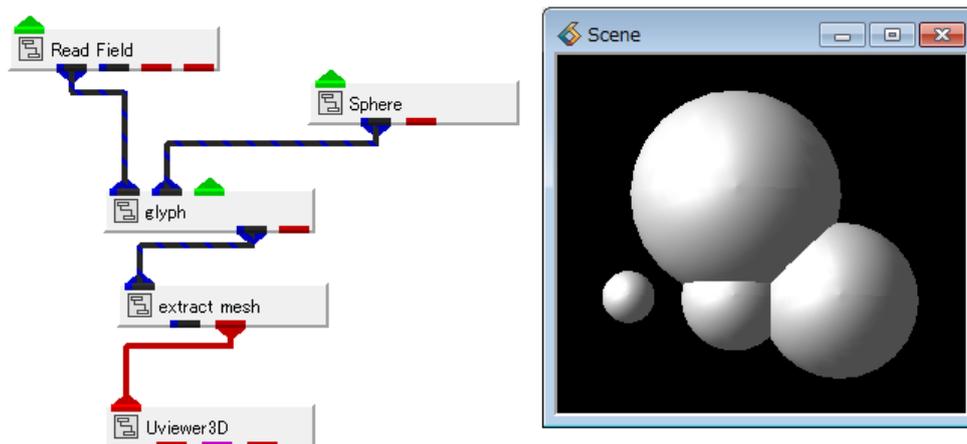


図 49 粒子の単色表示

#### ・ extract\_mesh モジュール

extract\_mesh モジュールは、ノードデータを削除するモジュールです  
ノードデータを削除すると、単色で色づけできるようになります。

デフォルトでは白で色づけされ、この色は「第6章 色、各属性の変更」で説明する  
オブジェクト・エディターで変更できます。

注) 後述しますが、セルデータがある場合には、ノードデータを削除しても、  
セルデータで色づけが行われます。

セルデータの場合は、セルデータを削除するモジュールはないため (V8.3 現在)、  
カラーマップを使って、単色に設定する必要があります。

補足)

**glyph** モジュールは、そのモジュールの接続方法によっては、以下の図のように、その形状の外形線が表示されます。

例えば、上記の例で、**set\_minmax** モジュールを使わずに、直接ビューワーに接続した場合には、球の分割状態を示すラインが表示されます。

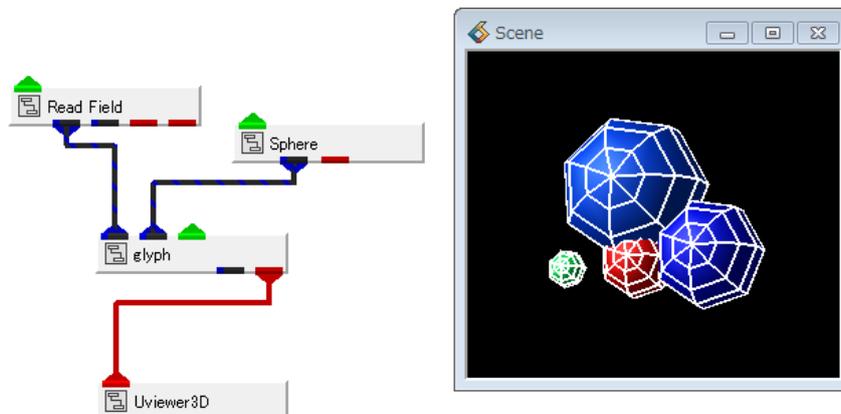


図 50 glyph モジュールの形状ライン

このラインは、オブジェクト・エディターの機能でオンオフを行うことができます。

- 1) カレント・オブジェクトを選択します。  
Select Object ボタンをクリックし、カレント・オブジェクトを選択、Apply ボタンをクリックします。  
glyph モジュールを直接接続している場合は、**glyph** という名前です。

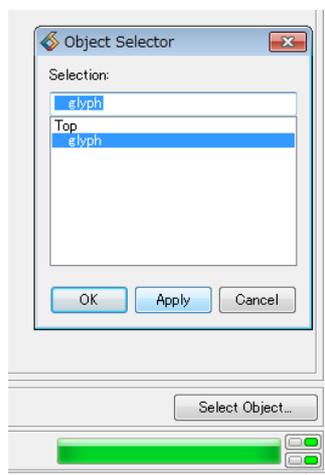


図 51 オブジェクトの選択

- 2) オブジェクト・エディターを開き、モードを変更します。  
下図のメニューからラインのレンダリング方法を変更します。

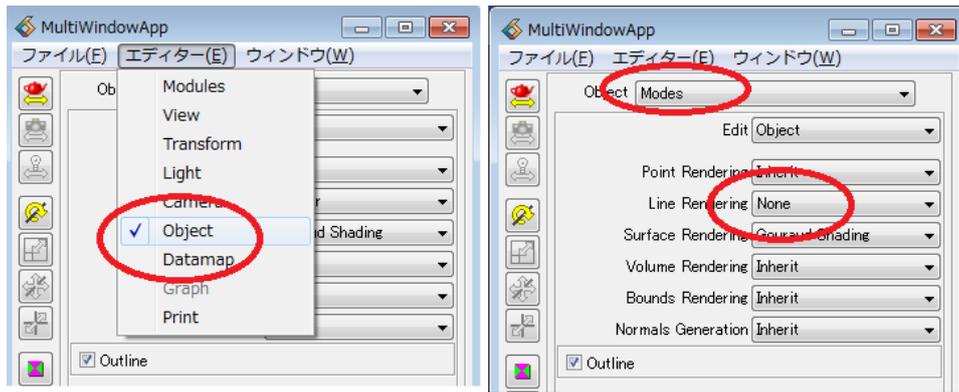


図 52 オブジェクトのモードの設定

- 3) カレント・オブジェクトを Top に戻します。  
忘れずに、1) の選択を元に戻してください。

### 3.5 その他の形状

その他にも、glyph モジュールに接続できる形状モジュールがあります。

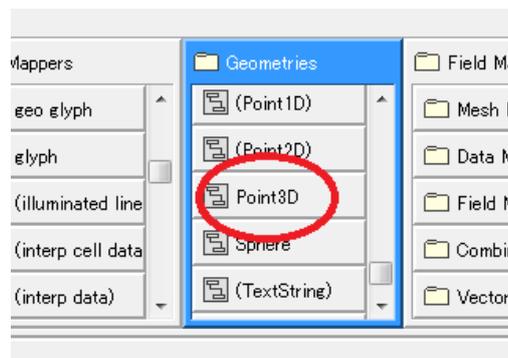


図 53 Geometries ライブラリ

Sphere モジュールと同じライブラリに、例えば、点や2本線のクロス（十字）形状のモジュールがあります。

- Point3D モジュール
- Cross モジュール
- Box モジュール
- Diamond3D モジュール

図 50のネットワークの Sphere モジュールの代わりに接続することで、その形状で表示できます（下図）。

例えば、大量の粒子を点（と色）だけで表現したい場合には、Point3D モジュールが利用できます。

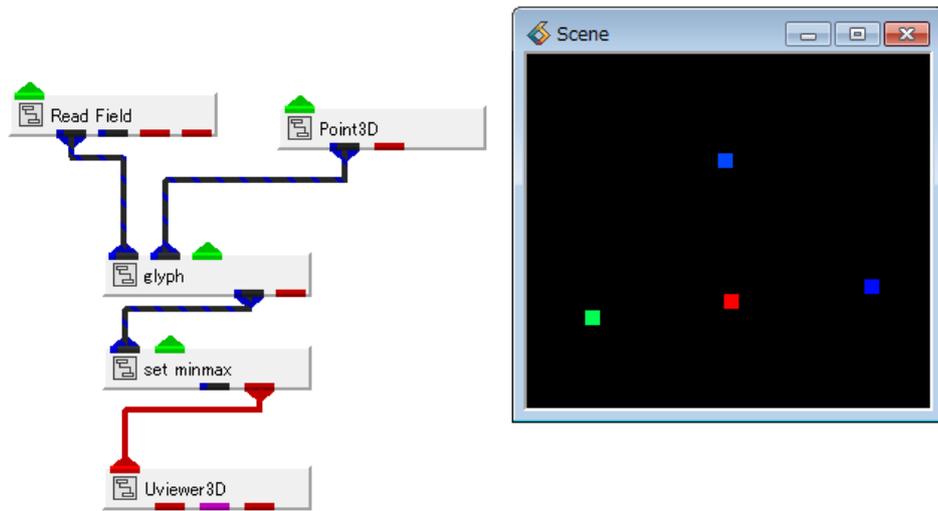


図 54 Point3D モジュールによる点表示

Point3D モジュールはピクセル（点）を作成します。  
デフォルトでは1ピクセルですが、そのサイズを変更することもできます。

重要) ただし、ハードウェアレンダラの場合のみ、変更可能です。

このサイズの変更は、オブジェクト・エディターの Properties、Point/Line メニューにある Line Thickness で設定できます。

カレント・オブジェクトを選択した後、オブジェクト・エディターを開きます。  
以下のメニューを順に選択します。

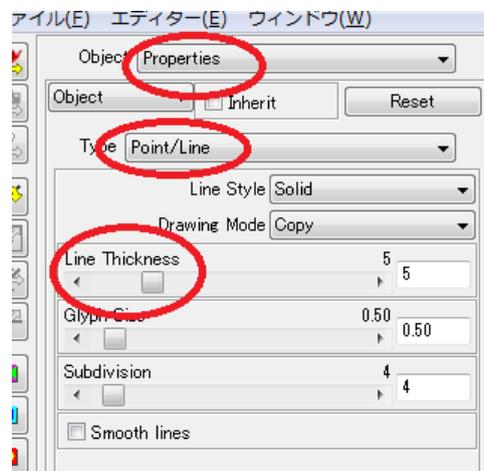


図 55 Line Thickness パラメーター

### 3.6 スティックの表示（チューブ形状）

分子間の結合を示すスティック（ボンド）は、チューブ形状で表示することができます。

下図の例では、先の set\_radius モジュールによる球の表示に tube モジュールを追加しています。

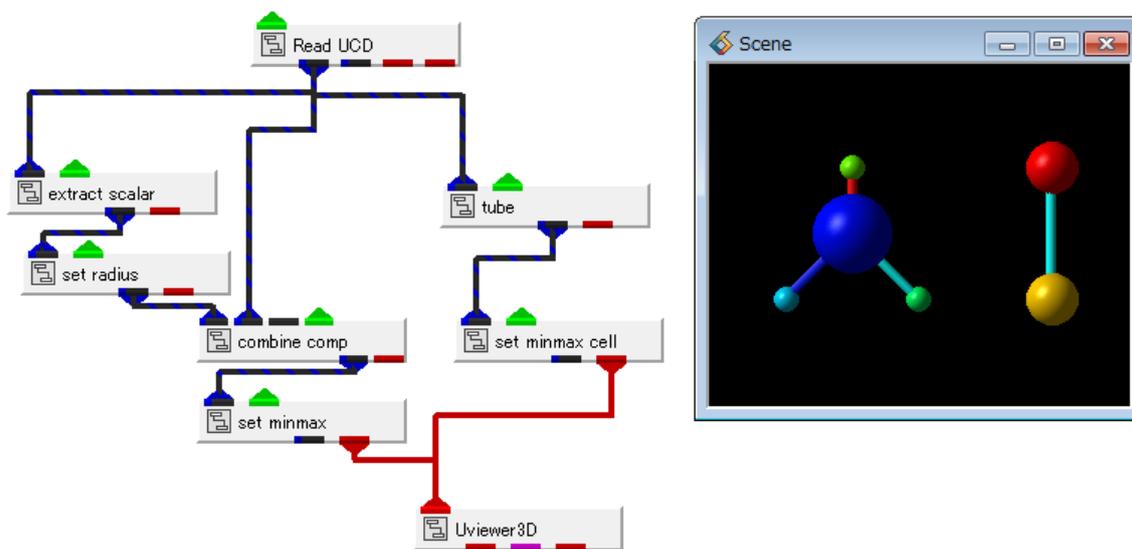


図 56 tube モジュールの追加

#### ・ tube モジュール

tube モジュールは、Line セルを結合し、且つ、太さを持ったチューブ形状で表現できるモジュールです。

パラメーターを以下に示します。

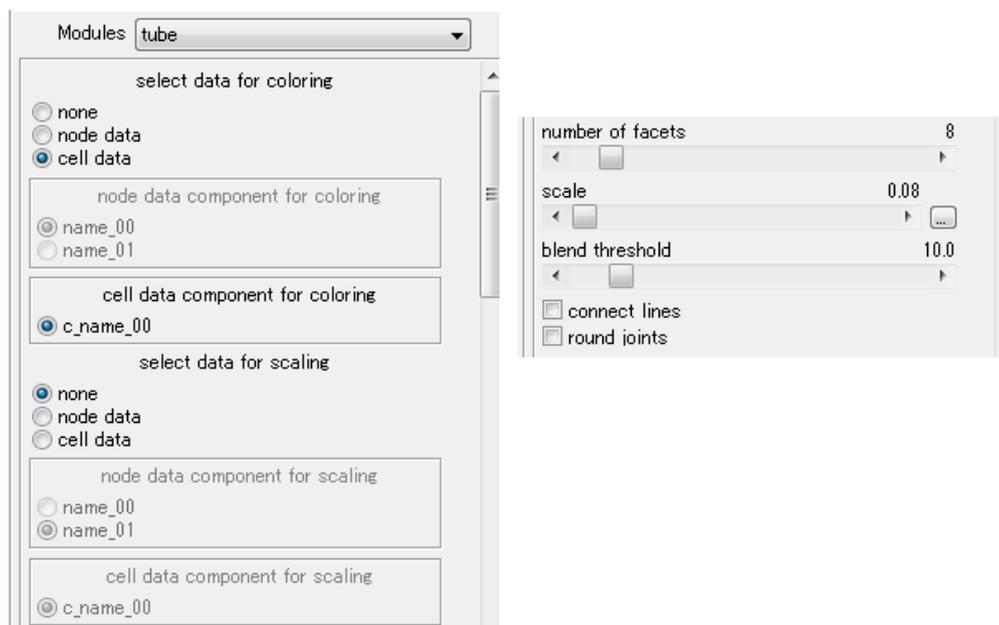


図 57 tube モジュールのパラメーター

色を変更するデータ成分の選択、太さ（スケール）を変更するデータ成分の選択があります。まず、いずれも（select data for coloring と select data for scaling）、none（選択なし）か node\_data（ノードデータ）か cell\_data（セルデータ）か、から選びます。

ノードデータの場合、Line の両端のノード上の値を使って、色や太さが変わります。両端のノードの値が異なる場合、色はその両端の色を使ったグラデーションに、太さは、円錐形となります。

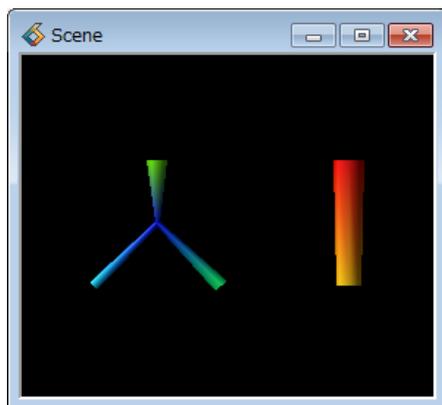


図 58 ノードデータで色と太さを変更した例

セルデータの場合、Line セルの中心に設定されたセルデータを使って色と太さを変更します。セルデータは、Line に対して1つの値となりますので、単色、また、その値に応じた円筒となります。

「図 57 tube モジュールのパラメーターの例では、セルデータの `c_name_00` 成分で色づけしています。

太さは `none` で一定としています。

また、`number of facets` スライダーは、チューブ形状の円筒方向の分割数を調整するパラメーターです。

太さの倍率は `scale` スライダーで調整できます。

その他はこの分子結合の表現では特に利用しません。

#### • `set_minmax_cell` モジュール

`set_minmax_cell` モジュールは、セルデータの `min/max` 値とカラーの対応に利用しています。

「図 57 tube モジュールのパラメーターの例では、セルデータで色づけしているため、この `set_minmax_cell` モジュールを利用しています。

ノードデータで色づけしている場合には、`set_minmax` モジュールを利用してください。

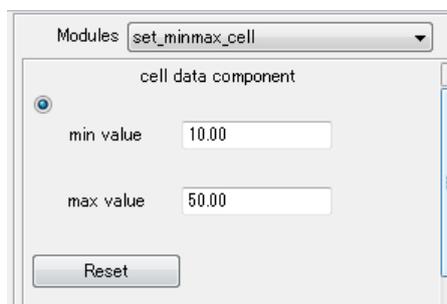


図 59 `set_minmax_cell` モジュール

データの選択成分名が表示されていませんが、tube モジュールで選択したセルデータの `min/max` 値が表示されます。

数値と色の対応の設定を行うことができます。

なお、単色で表示したい場合は、tube モジュールの赤ポートを直接ビューワーに接続してください。tube モジュールの `select data for coloring` のパラメーターを `none` に設定します。

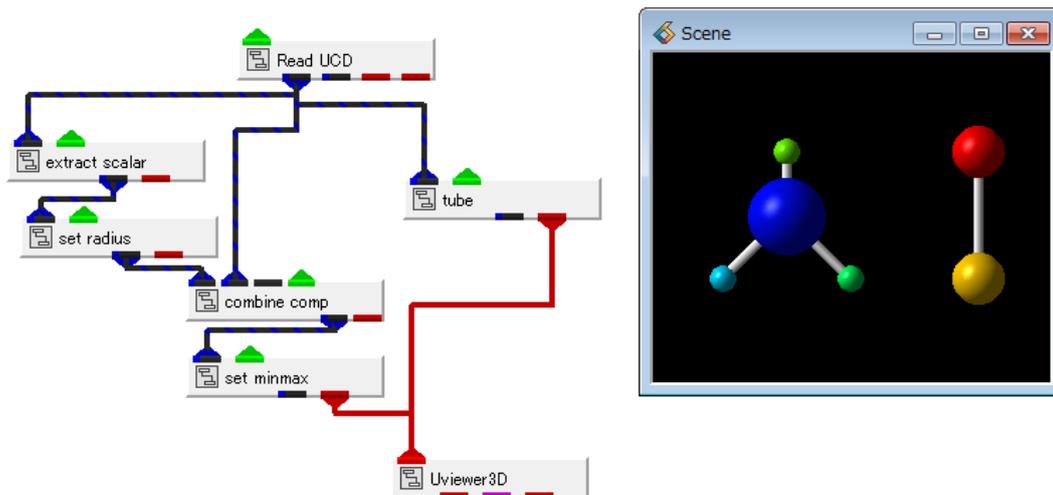


図 60 チューブ表示（単色）

単色の色設定は、オブジェクト・エディターで行います。「第6章 色、各属性の変更」をご参照ください。

### 3.7 スティックの表示（ライン形状）

分子間の結合を示すスティック（ボンド）を単なるラインで表示する方法です。

チューブではポリゴン数が多くなってしまいパフォーマンスが悪いといった場合には、ラインによる表示も検討してみてください。

まず、以下の図はセルデータで色づけしたい場合のネットワークです。

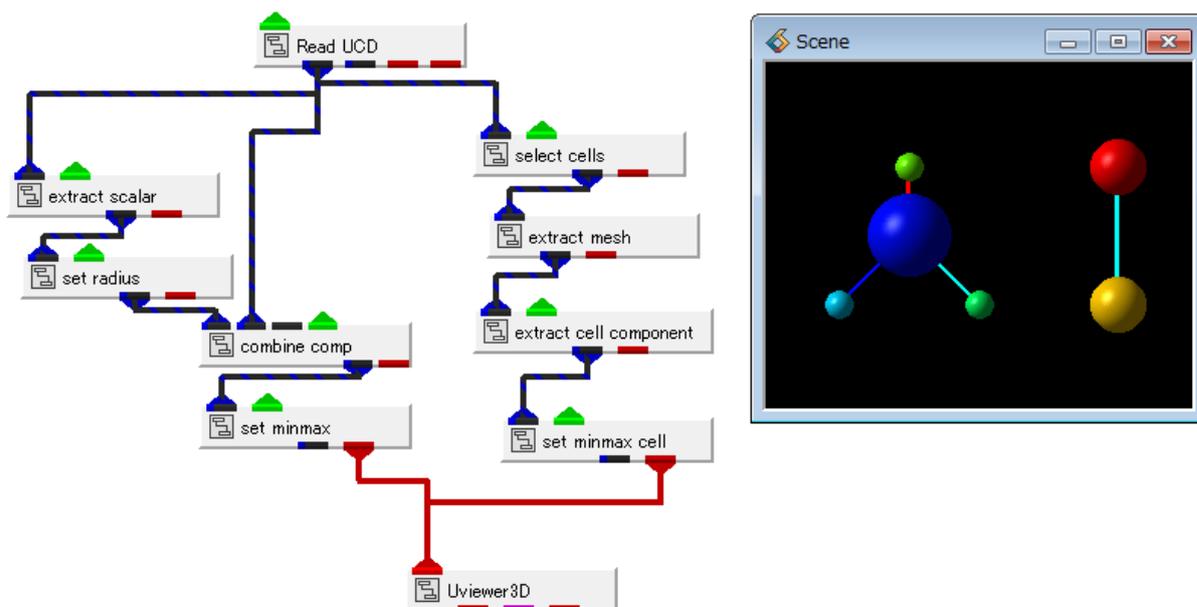


図 61 スティックのライン表示（セルデータで色づけ）

ノードデータで色づけしたい場合は、以下のように接続します。

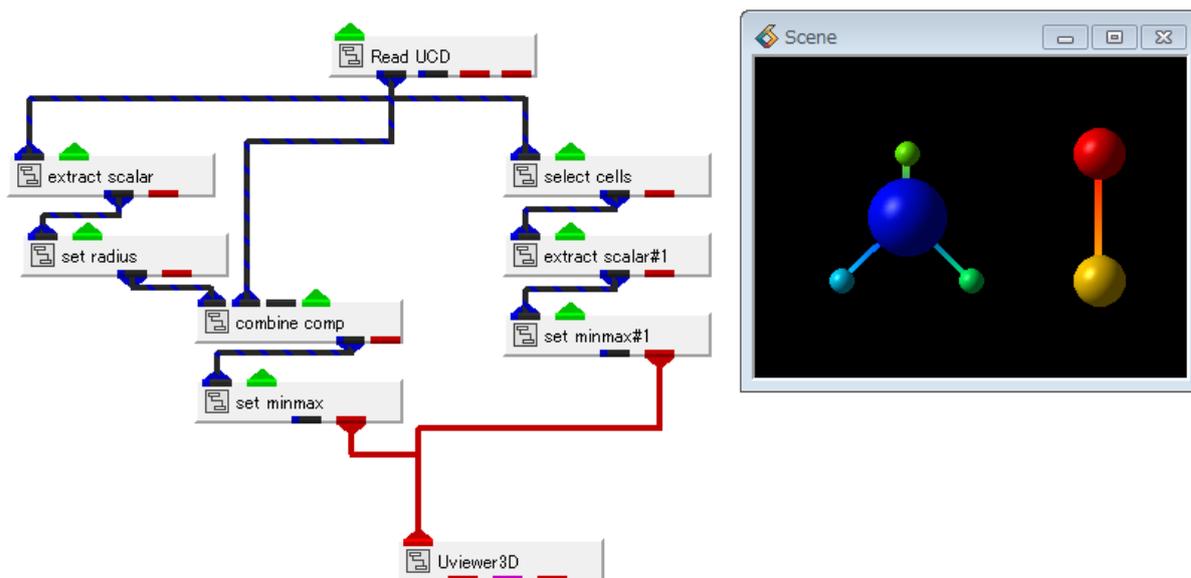


図 62 スティックのライン表示（ノードデータで色づけ）

・ `select_cells` モジュール

`select_cells` モジュールは、UCD（や Lammps）データで定義されている Point セル、Line セルのセルの選択を行うモジュールです。

上図の例では、いずれも、Line セルを選択し、処理対象を Line セルのみとしています。

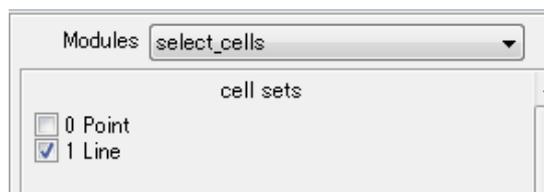


図 63 `select_cells` モジュール

・ `extract_mesh` モジュール

ノードデータを扱いたくない場合に利用できるモジュールです。

デフォルトでは AVS/Express は、ノードデータに値が設定されている場合、その値を色づけの対象として利用します。

「図 61 スティックのライン表示（セルデータで色づけ）」の例では、ノードデータがあるとセルデータよりも優先されてしまうために、このモジュールを間に挟むことで、ノードデータを削除しています。

※ セルデータを削除するモジュールは現在ありません。

単色にするには、カラーマップを使って、1色に設定してください。

「第6章 色、各属性の変更」をご参照ください。

もしくは `tube` モジュールで色なしによる表示を行います。

・ `extrace_cell_component` モジュールと `set_minmax_cell` モジュール

色づけしたいセルデータの選択とセルデータ用の数値と色の対応を設定するモジュールです。

• `extract_scalar#1` モジュールと `set_minmax#1` モジュール

AVS/Express では、同じ名前のモジュールには # (と数値) が自動的に割り振られます。

「図 62 スティックのライン表示 (ノードデータで色づけ) の例では、球の表示で

同じモジュールを利用しているため、#1 の名前となっています。

色づけしたいノードデータの選択と数値と色の対応を設定します。

点表示 (「図 54 Point3D モジュールによる点表示) で述べた点のピクセルサイズの変更 (「図 55 Line Thickness パラメーター) と同様に、同じ方法で、ラインの線幅を変更できます。

カレント・オブジェクトを `set_minmax_cell` か `set_minmax#1` のラインを作成しているオブジェクトとして選択します。線幅は同じ `Line Thickness` プロパティで設定できます。操作後は忘れずに Top に選択を戻すようにしてください。

※ 線幅はソフトウェアレンダラでも変更できます。

## 第4章 データの抽出

この章では、データの抽出関連について説明します。物理量（や ID）の大きさで、データの一部分を抽出したり、空間上のある範囲から範囲まで抽出するといったことが可能です。

### 4.1 空間に対する抽出

挟み込みやカット、ある半径内など、空間に対して設定した条件でデータを抽出します。

まず、空間的な位置からデータ抽出を行うには、crop で始まる名前のモジュールを利用できます。

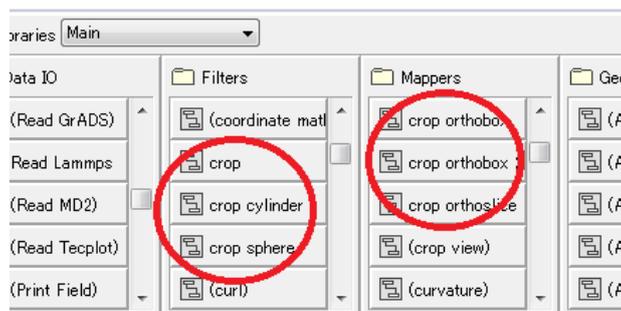


図 64 crop 関連モジュール

代表的なモジュールをいくつか紹介します。

モジュールのパラメーターの詳細など、モジュールリファレンスをあわせてご参照ください。

#### ・crop\_orthoslice モジュール

このモジュールは、ある幅を持った XYZ 方向のスライス断面を使った抽出を行います。スライスしたいデータの最初に、このモジュールを挿入します。図の例では、読み込んだ LAMMPS データ全体に対して、スライスで抽出した後、球やチューブを作成しています。

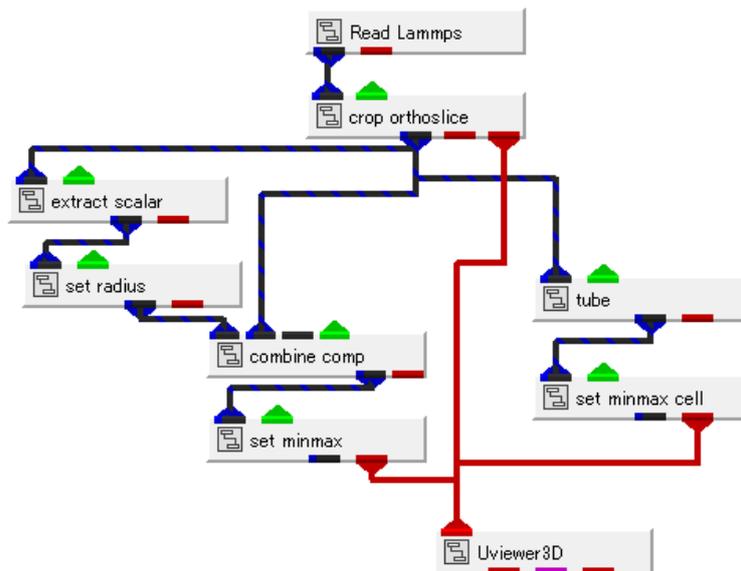


図 65 crop\_orthoslice モジュール

crop\_orthoslice モジュールの一番右の出力ポート（赤）を直接ビューワーに接続すると、その範囲を示す枠線を表示します。

抽出前と抽出の様子を下図に示します。

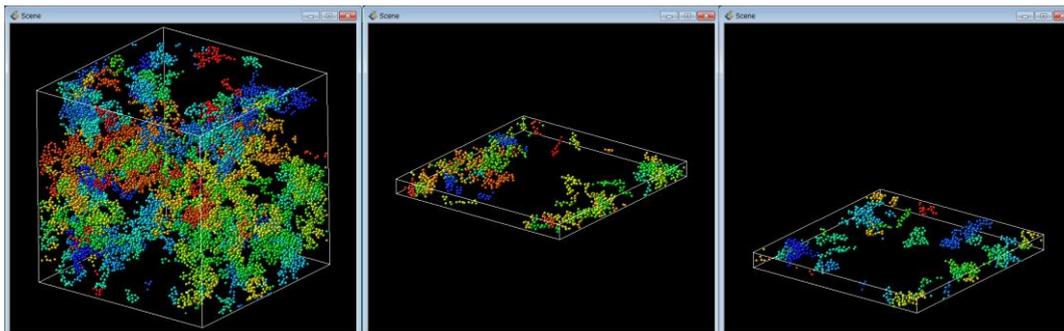


図 66 crop\_orthoslice モジュールによる抽出

パラメーターの詳細は省略しますが、XYZ の断面方向を選択し、Width パラメーターで、割合 (Relative) か絶対値 (Absolute) でスライス幅を指定します。

Move にある position スライダーでその幅を保持したまま、その位置を移動できます。また、Animation を使って、自動的に increments に指定した量ずつ、移動させることもできます。

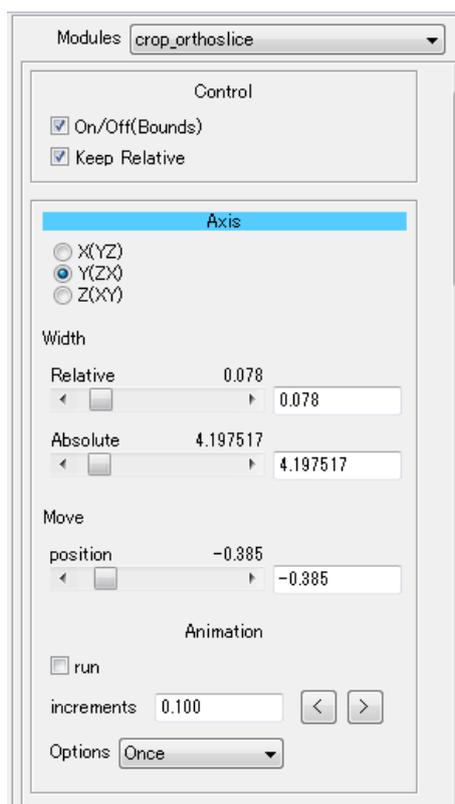


図 67 crop\_orthoslice モジュールのパラメーター

• `crop_orthobox_2way` モジュール

`crop_orthoslice` モジュールは幅を持ったスライス断面で抽出を行いました。

`crop_orthobox` モジュールは、XYZ の各方向に対して範囲を指定することで、抽出します。

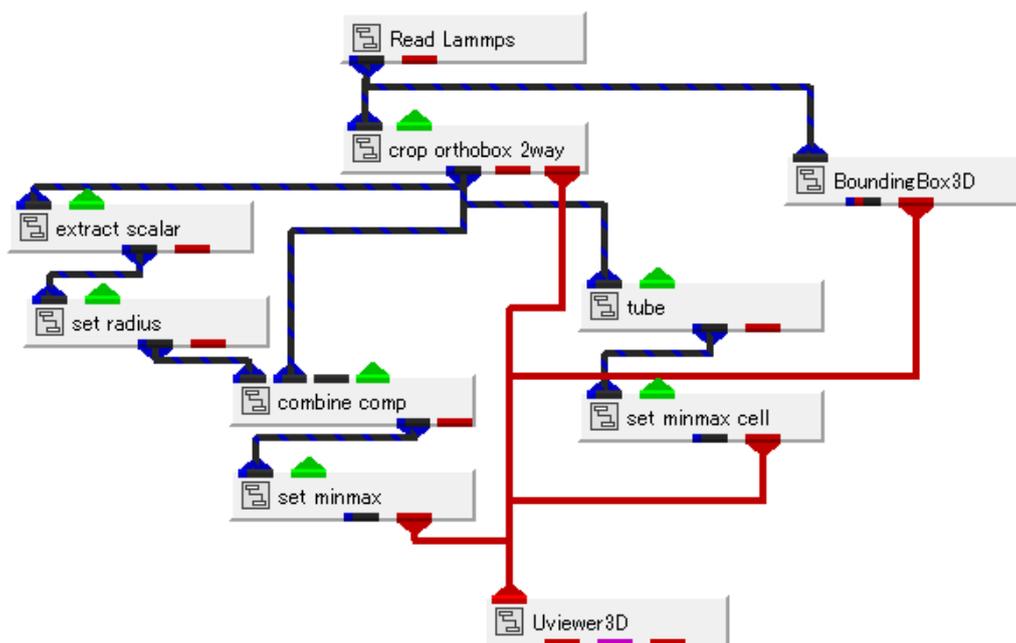


図 68 `crop_orthobox_2way` モジュール

「図 65 `crop_orthoslice` モジュールと同様、このモジュールを挿入した前後でデータの抽出が

行われます。

このモジュールは、`crop_orthoslice` モジュールの機能を含んでいます。スライスだけではなく、XYZ の各方向に対して範囲を設定できます。

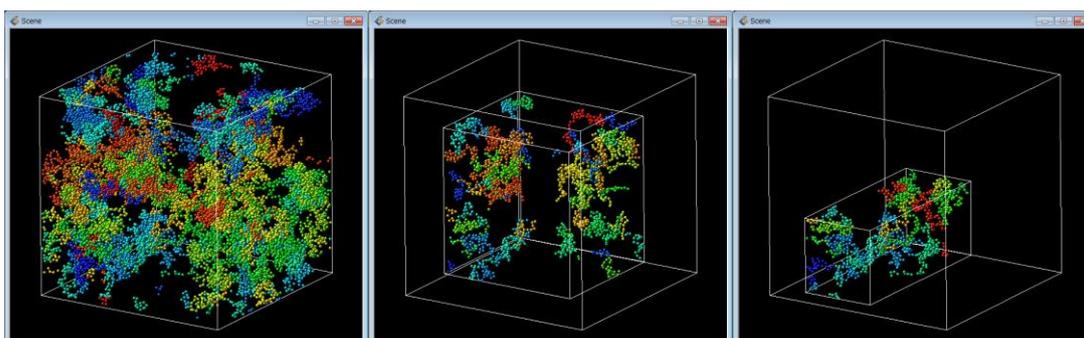


図 69 `crop_orthobox_2way` モジュールによる抽出

`crop_orthobox_2way` モジュールには、XYZ 方向のサイズや位置を指定するパラメーターがありますので、適宜、操作してみてください。

また、In/Out のパラメーターを使って、内側か外側かを選択することもできます。

- **BoundingBox3D** モジュール (補足)

また、上記のネットワークには、**BoundingBox3D** モジュールを追加しています。元データの範囲の枠線を表示するために利用しています。

その他、BOX 形状で抽出を行うモジュールとして、以下のモジュールもあります。

- **crop\_orthobox** モジュール

**crop\_orthobox\_2way** モジュール同様、XYZ の範囲の絞込みで抽出できます。

- **crop\_box** モジュール

**crop\_orthobox\_2way** モジュールや **crop\_orthobox** モジュールと同様、ボックス形状で抽出を行います。

XYZ 固定ではなく、任意にボックスを回転、移動を行うことができます。

- **crop\_sphere**、**crop\_cylinder** モジュール

球形状や円筒形状で抽出するモジュールです。

使い方は、これまでの例と同じです。データの読み込み後、このモジュールを接続します。

パラメーターで、球の中心と半径、また、円筒の場合は2点(両端)と半径を指定することで、球形状や円筒形状で抽出できます。

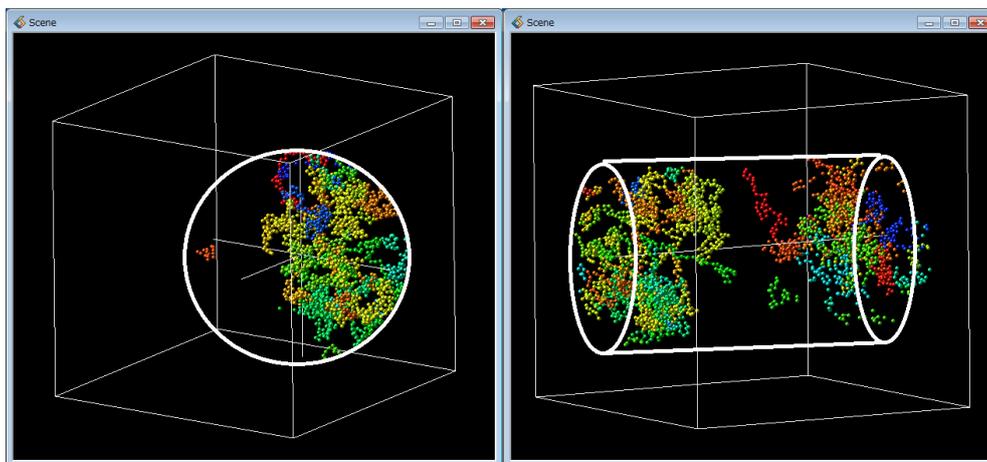


図 70 球や円筒による抽出

※ 図の白い線は説明のために追記したもので、AVS 上では表示されません。

・ crop\_area\_box モジュール

crop\_area\_box モジュールは、対話的に画面操作で選択した領域のデータを抽出できるモジュールです。

以下のように接続します。

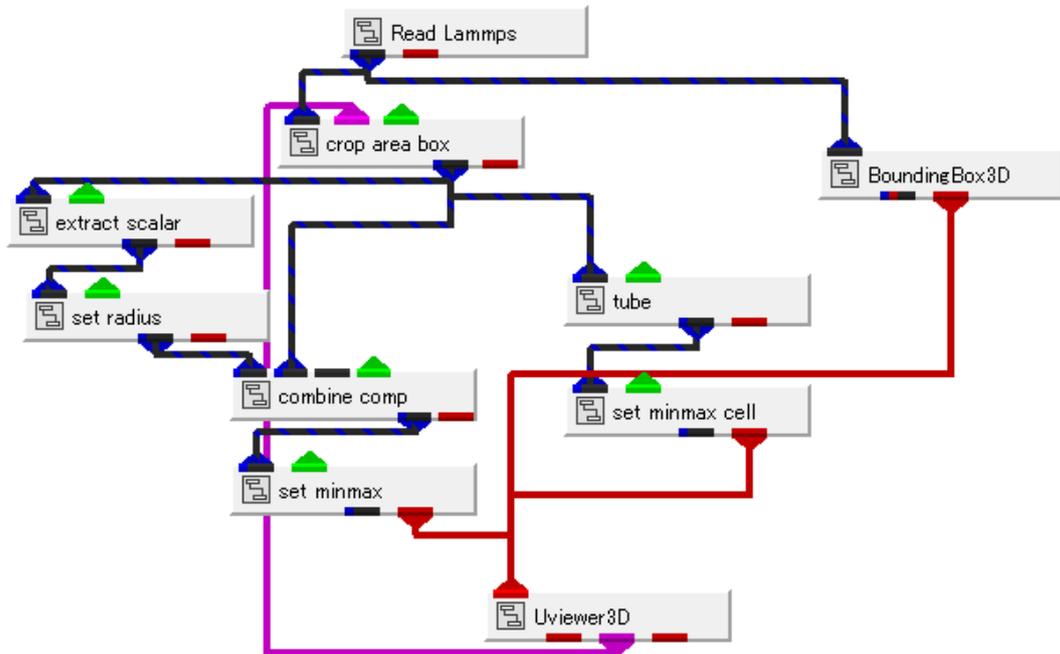


図 71 crop\_area\_box モジュール

先述のモジュールの接続に加え、ビューワー・モジュール（Uviewer3D）とピンクのポートを接続している点に着目してください。

操作方法は以下の通りです。

まず、画面上でマウス右ボタンを押したまま、四角の領域を選択するように、マウスを動かし、放します。

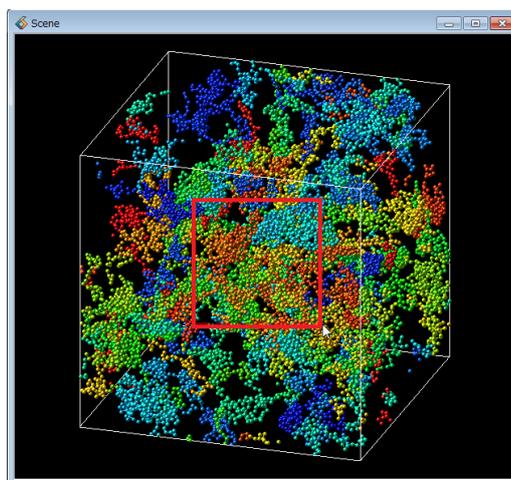


図 72 crop\_area\_box モジュールの領域選択

画面上で赤い線が表示されます。

次に `crop_area_box` モジュールのパラメーターの `Crop` ボタンをクリックします。

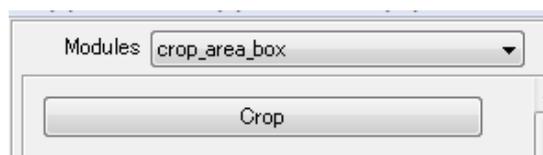


図 73 Crop の実行

選択した領域を画面に垂直な方向にボックスを作り、その間に含まれるデータを抽出します。

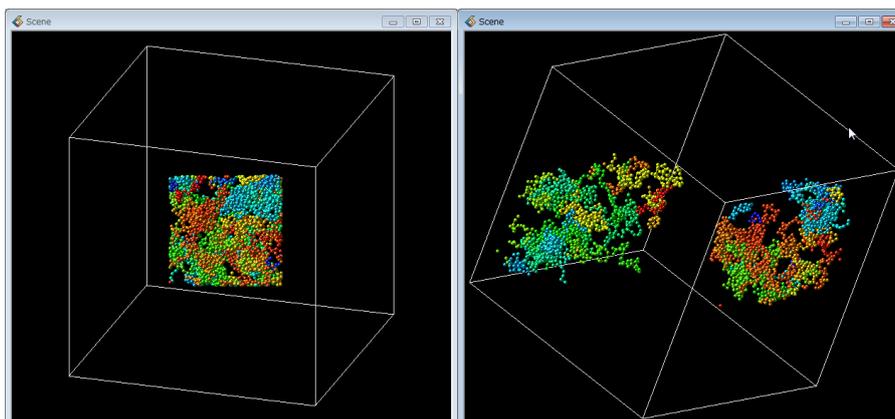


図 74 領域抽出の結果（右図は回転した様子）

## 4.2 空間に対する抽出（カット）

また、空間に対する抽出方法として、ある方向からカットし、片側を残す方法もあります。  
cut で始まる名前のモジュールを利用できます。

- cut\_plane モジュール

cut\_plane モジュールは、任意の断面で片側をカットできるモジュールです。

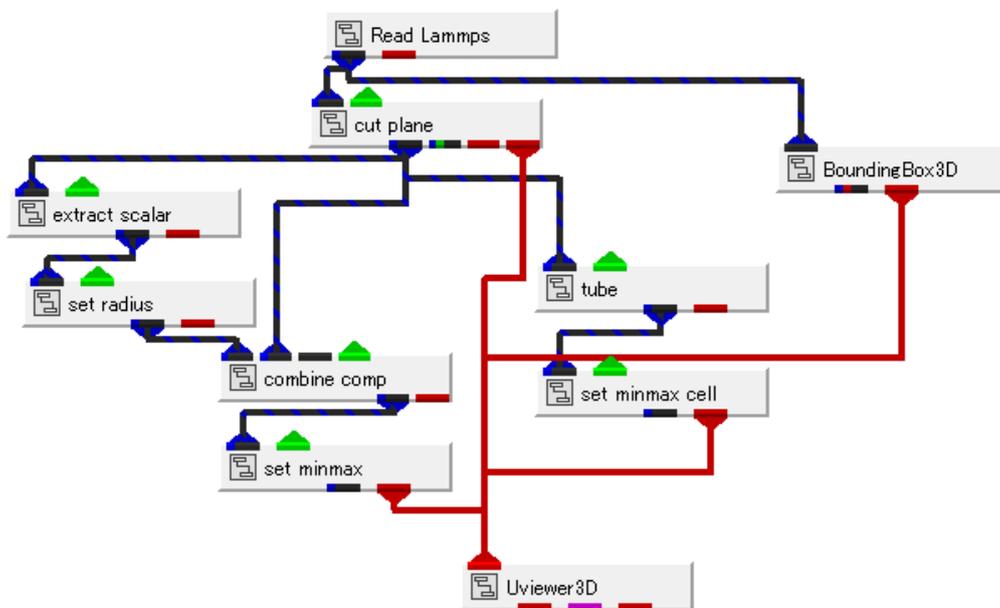


図 75 cut\_plane モジュール

結果を以下に示します。

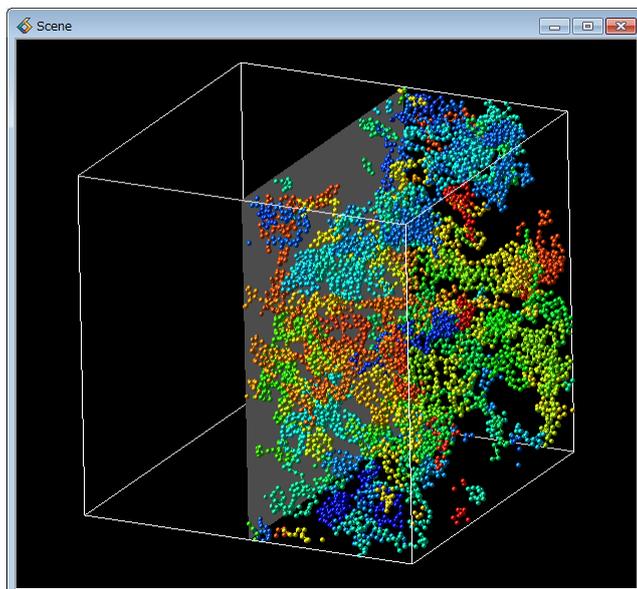


図 76 cut\_plane モジュールによるカット

cut\_plane モジュールの右端の出力ポート（赤色）を接続していると、その平面位置が表示されます。

この断面位置を cut\_plane モジュールのパラメーターで操作し、ある断面の片側をカットすることができます。

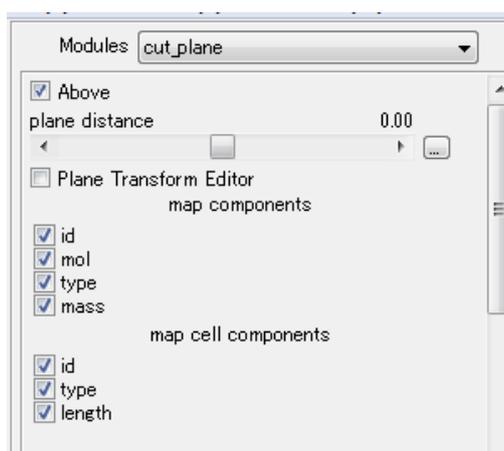


図 77 cut\_plane モジュールのパラメーター

まず、map component と map cell components では、そのカット後にも出力するデータを選択します。

すべての成分にチェックします。

断面の位置は、plane distance スライダーで移動できます。

さらに、Plane Transform Editor にチェックすると、回転などを行う設定パネルが開きます。

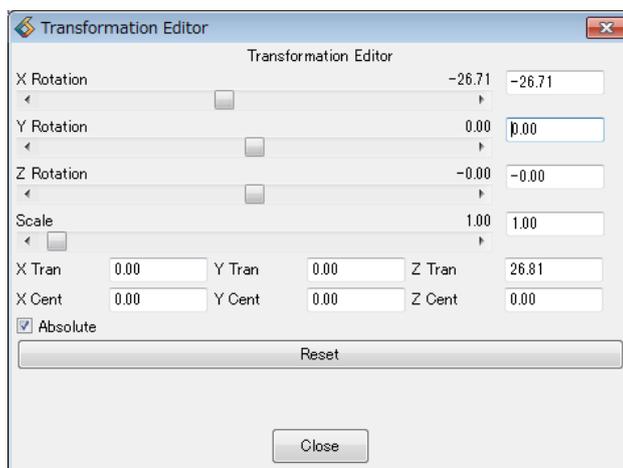


図 78 Transoformation Editor

この面の位置決めは、XYZ の Cent で指定される原点に対して、回転や移動を行います。そのため、面の位置指定が難しいかもしれません。

次に示す cut\_arbitplane モジュールでは、3点を通る断面、1点と法線ベクトルで作成される断面などを作成できます。

なお、Above のチェックは、断面のどちら側をカットするかを選択を切り替えることができるパラメーターです。

• cut\_arbitplane モジュール

3点を通る断面、1点と法線ベクトルで作成される断面など、簡単な方法で断面指定が行えるモジュールです。

さらに、このモジュールは、画面上の表示をピックしながら、面の位置を設定できる機能を持っています。

そこで、そのインタラクションのために、Uviewer3D モジュールの赤ポートとピンクポートの2つのポートを cut\_arbitplane モジュールの入力ポートに接続してください。

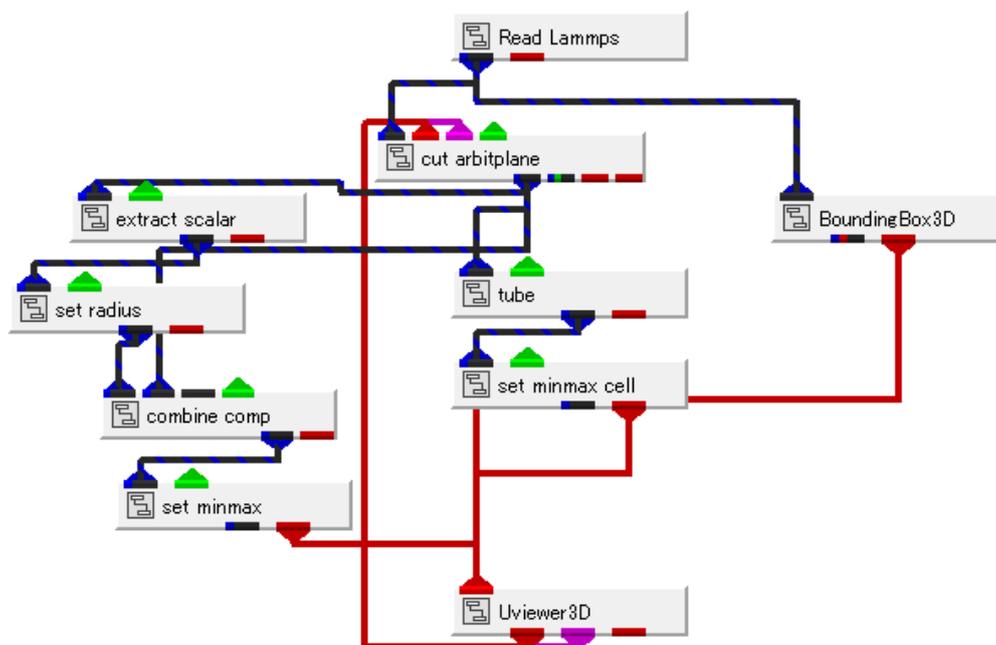


図 79 cut\_arbitplane モジュール

このモジュールのパラメーターを以下に示します。

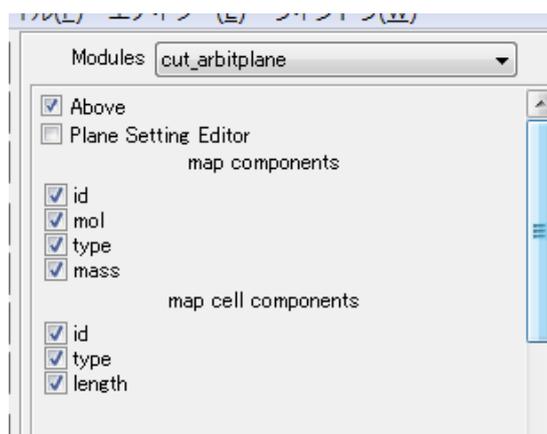


図 80 cut\_arbitplane モジュールパラメーター

まず、map component と map cell components では、そのカット後にも出力するデータを選択します。

すべての成分にチェックします。

次に **Plane Setting Editor** にチェックしてみてください。  
以下のパネルが開きます。

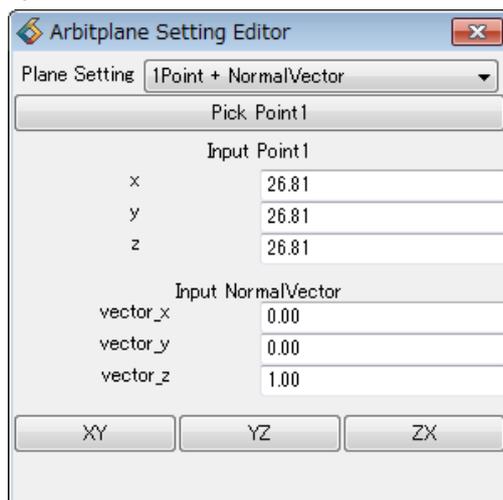


図 81 Arbitplane Setting Editor

4つの方法で断面の位置を設定できます。  
**Plane Setting** のメニューから、いずれかを選択します。

- 1Point + NormaVector : 1点と法線ベクトル
- 2Point + Direction : 2点と XYZ のいずれかの方向
- 1Point + RotationAngle : 1点と XYZ 軸に対する回転角度
- 3Point : 3点指定（3点を通る断面）

※ 特に 3Point を選んだ際に、デフォルトの位置によりデータが存在せずに、以下のエラーが出る場合があります。

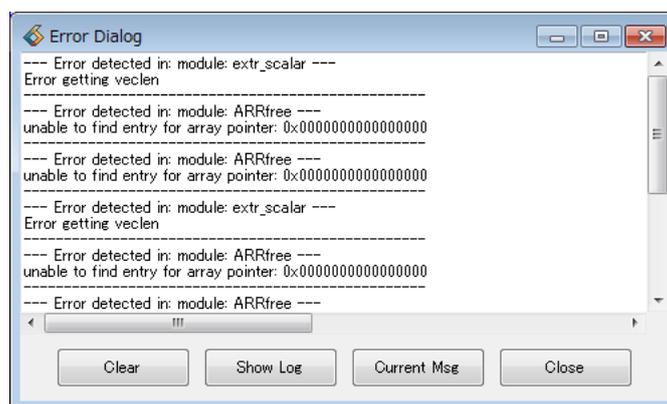


図 82 断面指定時のエラー

断面の位置によって、データが存在せずにエラーになっていますので、まずは、無視して、次の断面指定を行ってみてください。

ここでは、3点を画面上でピックして設定する方法について紹介します。

- 1) まず、Plane Setting から 3Point を選びます。
- 2) 次に、Pick Point1 のボタンをクリックします（色が変わります）。
- 3) 画面に表示されているオブジェクトを “Ctrl キー” を押しながら、マウス左でクリックします。  
例えば、下図では、枠線のある角をピックしています。

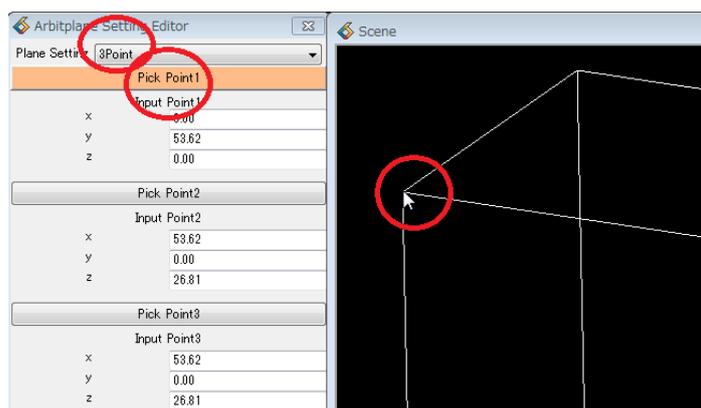


図 83 画面上のピック

ピック値が取得できると、パラメーターの画面の xyz に数値が反映されます。

- 4) 同じ操作で2点目を選びます。  
Pick Point2 ボタンを押し、画面上で Ctrl キー+マウス左ボタンでクリックします。  
下図では、反対側の角をピックしています。

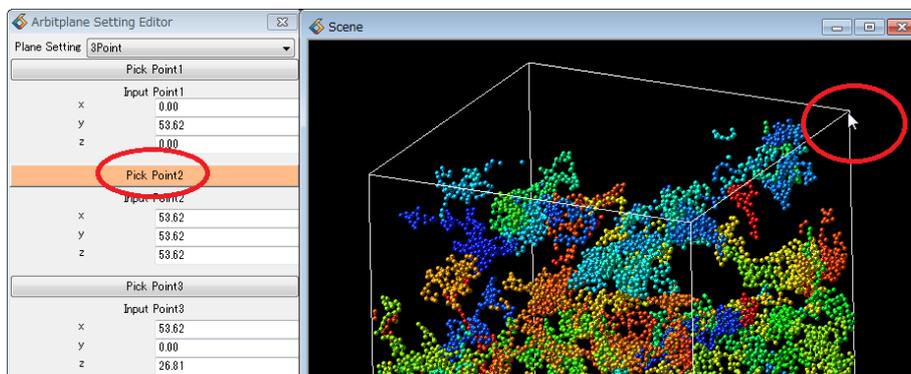


図 84 画面上のピック2点目

- 5) さらに、3点目を選びます。  
下図の例では、説明のため、赤線で三角形をドキュメント上で追記していますが、この3点を通る断面が設定され、データがカットされています。

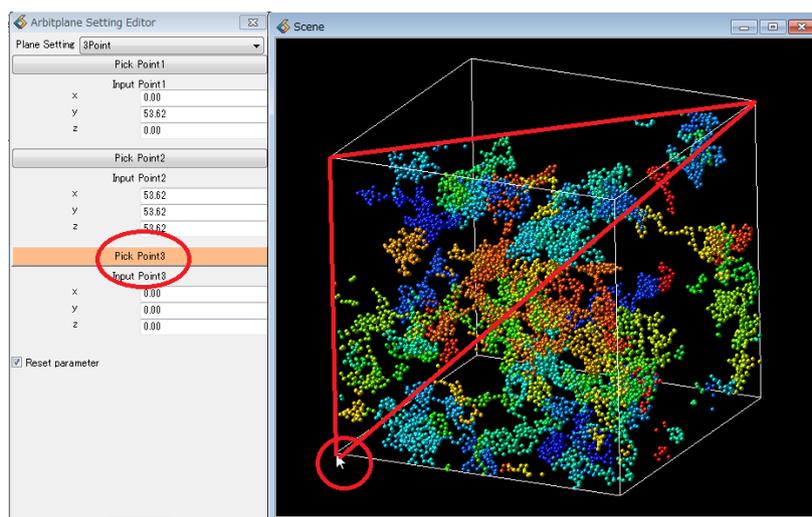


図 85 画面上のピック 3 点目

下図は、モジュールのパラメーターにある Above のチェックをオフにし（カットを反対側に設定）、上から見た様子を示しています。

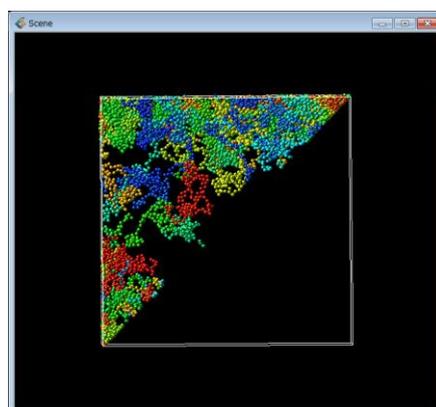


図 86 断面カット（上から見た図）

断面の設定パネルに **Reset parameter** というチェックがあります。通常、データの読み替えなどを行うと、この断面の設定パラメーターはリセットされるようになっています。  
 V ファイルとして作業状態を保存する場合や、時系列を進めた際などに、パラメーターを保持したままにするには、チェックを外すようにしてください。

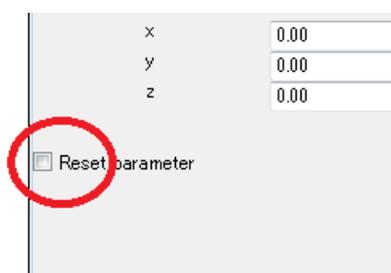


図 87 パラメーターの保持

### 4.3 データ値を利用した抽出

ある ID のデータを抽出、ある大きさ以上や以下など、物性値を使ったデータの抽出を行います。

#### • threshold\_all モジュール

まず、ノードデータに対するしきい値処理を行うモジュールです。

以下のネットワークをご参照ください（説明のため、tube モジュールによるスティックの表示は削除しています）。

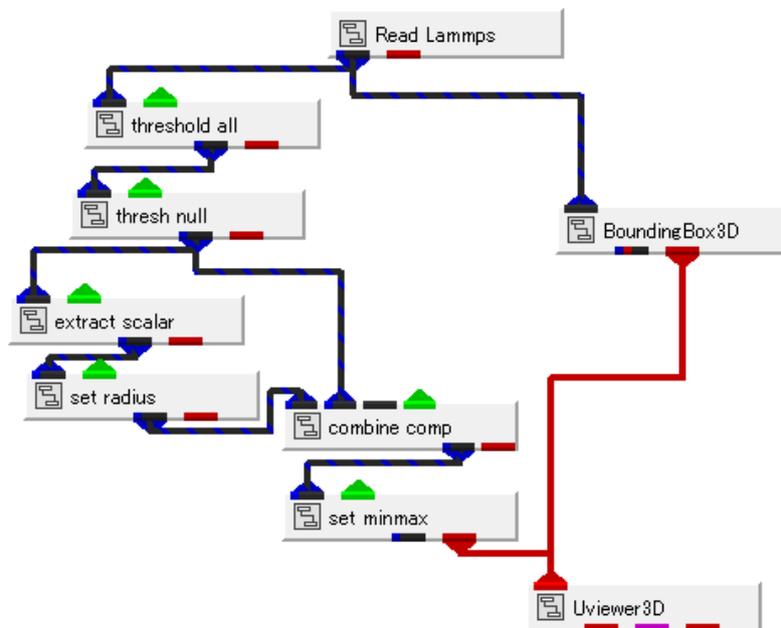


図 88 しきい値処理（ノードデータ）

threshold\_all モジュールのパラメーターを以下に示します。

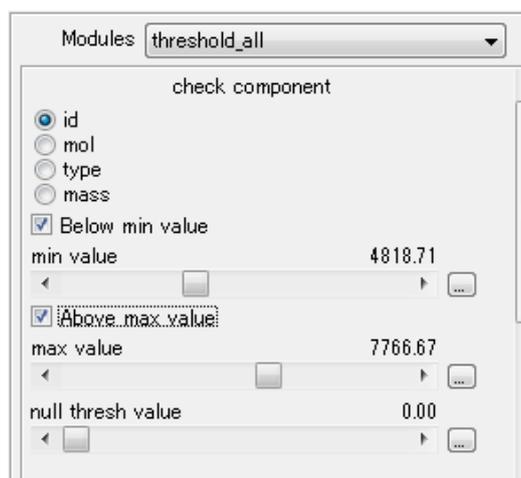


図 89 threshold\_all モジュールパラメーター

このモジュールは、選択した check component の成分値の min/max を指定することで、範囲抽出できるモジュールです。

上図の例では、**Below min value** と **Above max value** にチェックし、それぞれに、  
適当な値を設定しています。

この範囲内の **id** 値（ノードデータ）を持った球のみが表示対象となります。

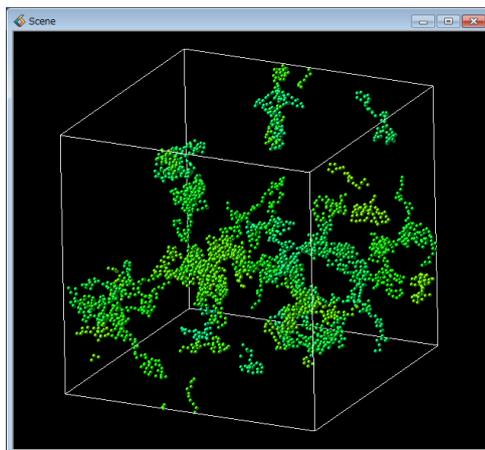


図 90 ノードデータによる抽出

色の設定は、**combine\_comp** モジュールで行っていた点を思い出してください。  
**extract\_scalar** モジュールは球の半径を変えるためのデータ成分の抽出で、  
色づけ用のデータ成分は **combine\_comp** モジュールの **data\_components2** で  
指定しています。

また、その色（赤青）と数値の対応は、**set\_minmax** モジュールで行っています。

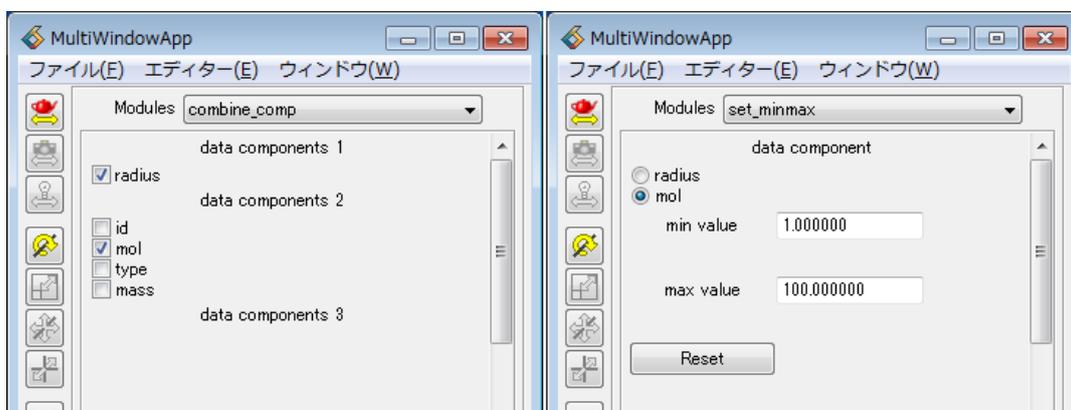


図 91 色づけの設定

この例では、**id** でデータを抽出し、その抽出された球を、**mol** 値で色づけしています。  
抽出によって、**1** や **100** の値が存在しなくなったため、ほとんどが緑色となっています。

抽出後のデータに対して青から赤の変化を付けたい場合は、**set\_minmax** モジュールの  
**min/max** で調整します。

次の図のように、**min** から **max** の範囲を狭めると、色の変化を見ることができます。

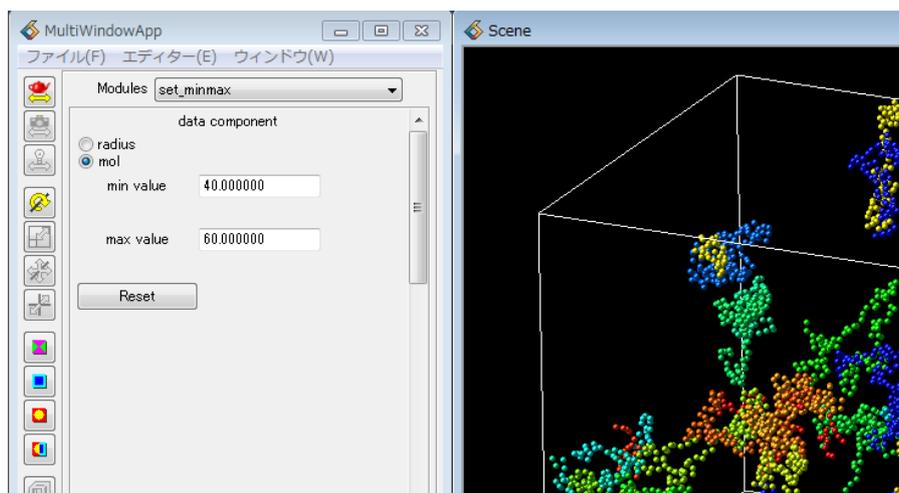


図 92 set\_minmax モジュールの調整

・ thresh\_null モジュール

「図 88 しきい値処理（ノードデータ）のネットワークでは、threshold\_all モジュールの後ろに thresh\_null モジュールを接続しています。

AVS/Express では、ノードに null 値（任意の値）を設定（null 値を利用するというフラグをオン）することで、そのノードを非表示として扱うことができます。

例えば、図 88のネットワークでは、threshold\_all モジュールの出力には、null フラグがオンに

設定され、抽出範囲外に null 値（threshold\_all モジュールのパラメーターで指定した値）が設定されています。

この例では、thresh\_null モジュールがなくても、null 値は非表示として扱われるため、同じ表示となります。

thresh\_null モジュールは、null 値の設定されているノードを削除するモジュールです。モジュールの中には、null 値を反映しないモジュールもあります。

そこで、このように null 値を設定した後、そのノードそのものを削除しています。

パラメーターで、null 値を設定したデータ成分（この例では id）を選択してください。

※ threshold モジュールと threshold\_all モジュール

2つの同じ機能を持ったモジュールがあります。

threshold モジュールは、抽出範囲を指定する成分（check component）と、その抽出後に出力する成分（threshold component）を指定します。

threshold モジュールの出力は、その出力対象の1成分のみとなります。

一方、threshold\_all モジュールは、抽出範囲を指定する成分も含め、抽出後も、すべてのデータ成分が出力されます。

メモリ消費を抑えたい、1成分しか扱わない場合などには、threshold モジュールが便利な場合もあります。

• threshold\_cell\_all モジュール

セルデータ（Line セル上に設定したデータ）に対しても同様に、しきい値処理を行うモジュールがあります。

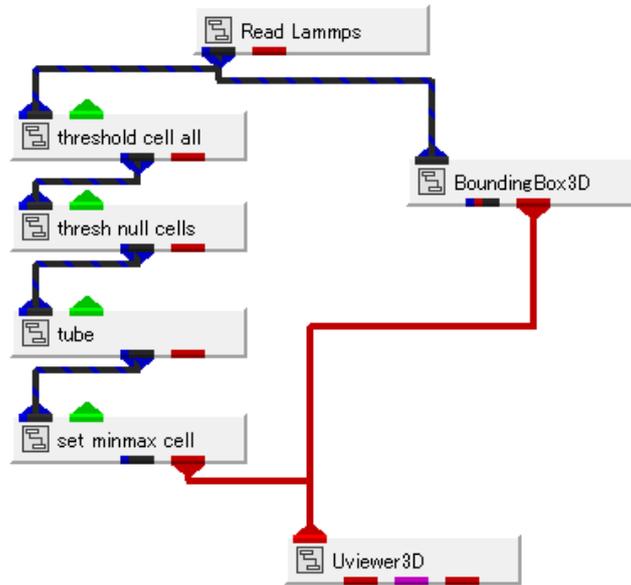


図 93 threshold\_cell\_all モジュール

tube モジュールの表示に対して、先の例と同様、threshold\_cell\_all モジュールと threshold\_null\_cells モジュールを接続しています。パラメーターも同じように、min/max の範囲を指定します。

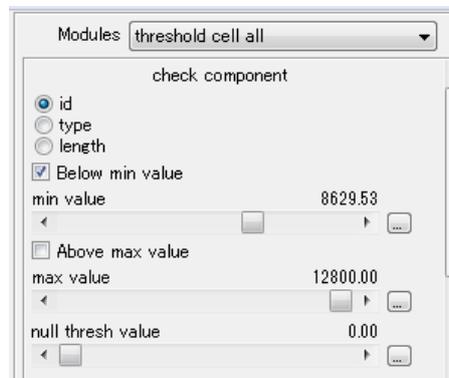


図 94 threshold\_cell\_all モジュールパラメーター

以下に抽出された様子を示します。

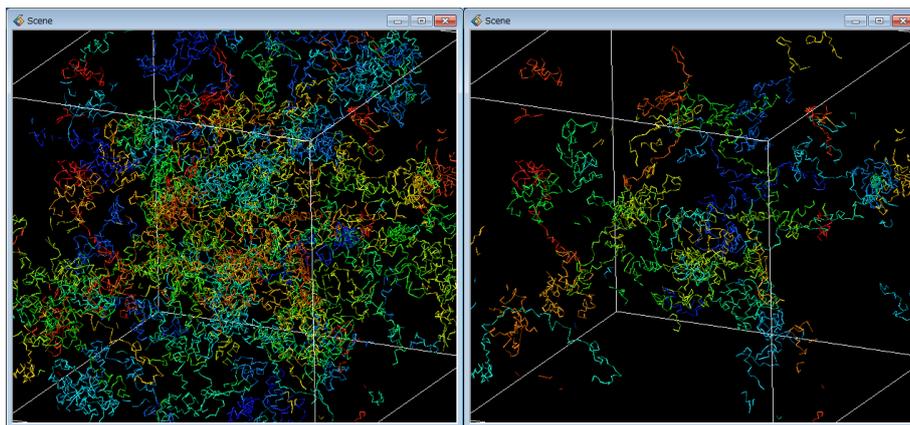


図 95 セルデータによる抽出

ノードデータと同様、`thresh_null_cells` モジュールで `null` 値が設定されたセルを削除しています。

パラメーターで、`threshold_cell_all` モジュールで抽出したデータ成分を選択してください。

※ `Read_Lammps` モジュールでは、モジュール内部で、粒子間のボンドの長さを計算し、`length` データをセルデータに設定しています。

例えば、ある長さ以下のセルは削除するという事も可能です。

以下のネットワークは、上記2つのネットワークを一緒に利用した例です。

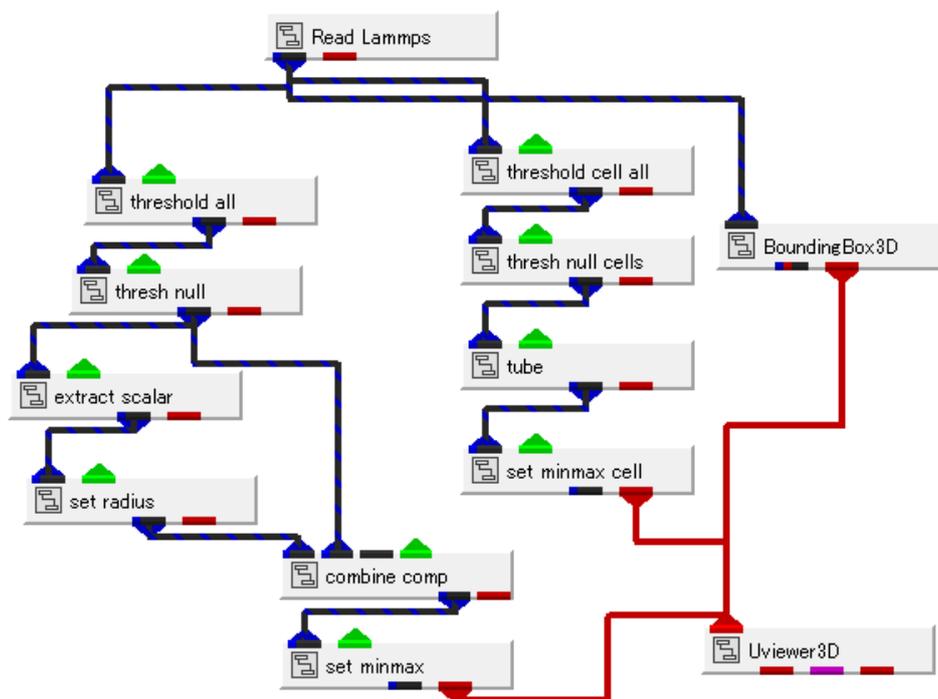


図 96 データ値による抽出

## 第5章 データの可視化（メッシュデータ）

電荷密度分布など、メッシュデータ（ボリュームデータ）を可視化する方法について説明します。

※ メッシュデータの可視化については、分野は異なりますが、断面コンター図や等値面など、気象分野に関するハンドブック「気象データの3次元可視化」でも触れていますので、あわせてご参照ください。

### 5.1 データの準備

電荷密度などのメッシュデータに対して、その空間内の断面のコンター図や等値面を作成できます。AVS/Express では、配列で表現できるメッシュデータは、フィールドデータとして定義することで扱うことができます。

まず、座標値として、以下の3つのタイプをサポートしています。図は2次元の配列データ（I, Jの配列）を表現していますが、1次元、もしくは、3次元の配列を扱うことができます。

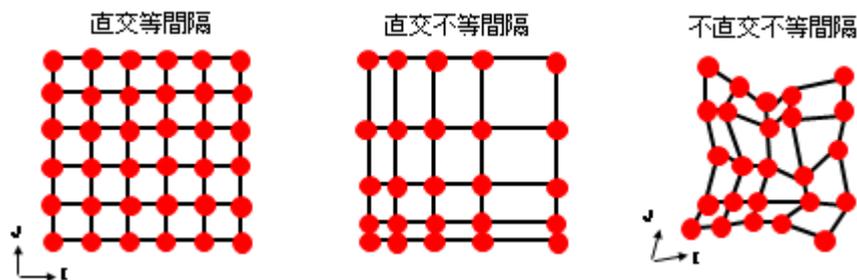


図 97 メッシュデータの座標

この各配列に密度や温度など、物理量を定義できます。

フィールドデータの詳細については、ユーザズガイドをご参照ください。

### 5.2 断面コンターによる表示

3次元のメッシュデータに対して、そのメッシュ断面を作成するには、以下のモジュールを利用できます。

- orthoslice モジュール
- slice\_orthoplane モジュール

ネットワークを以下に示します。

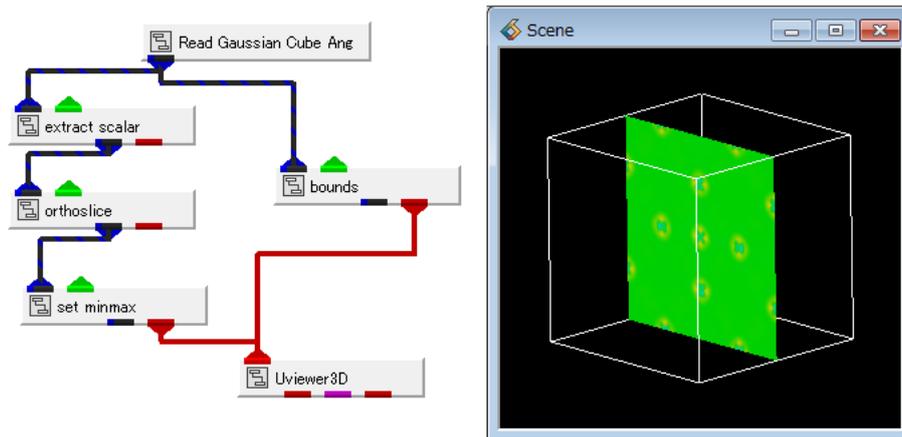


図 98 メッシュ断面の表示

この例では、Gaussian の Cube ファイルを読み込んでいます。

bounds モジュールは、メッシュの外形線を表示するモジュールです。

extract\_scalar モジュールで選択した成分に対して、orthoslice モジュールでメッシュ断面を作成しています。IJK のいずれかの方向とその配列の位置をパラメーターで指定します。

set\_minmax モジュールは、色と数値の対応を設定するために利用しています。ボール&スティックの表示で述べた「図 35 set\_minmax モジュール」などをあわせてご参照ください。

orthoslice モジュールは、メッシュの断面（配列の位置）を作成します。一方、slice\_orthoplane モジュールは、XYZ 方向の指定した座標値で断面を作成できるモジュールです。メッシュ断面ではなく、ある座標位置に断面を作成したい場合には、上図の orthoslice モジュールの代わりに、slice\_orthoplane モジュールを利用します。

その他、等高線を作成する isoline モジュールもあります。上図では、塗りつぶしの面コンター図を作成していますが、isoline モジュールを利用すると、線コンターを作成することができます。

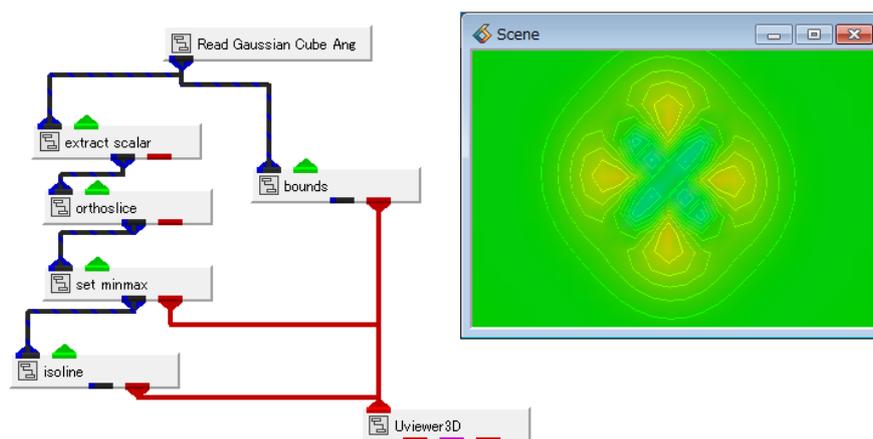


図 99 isoline モジュール

線コンターだけにしたい場合は、set\_minmax モジュールからビューワーに接続している赤ポートを切断してください。

※ 「第 6 章 色、各属性の変更」にあるライトの調整やジッター機能などについてもあわせてご参照ください。

### 5.3 等値面表示

等値面は、空間内の同じ値を面で結ぶことで、どのようにその値が分布しているかを見ることができる方法です。

- isosurface モジュール
- isosurface\_nest モジュール

以下のように、選択したデータ成分に対して、等値面を作成します。パラメーターにある iso\_level を使って、その等値面を作成するレベル値を指定できます。

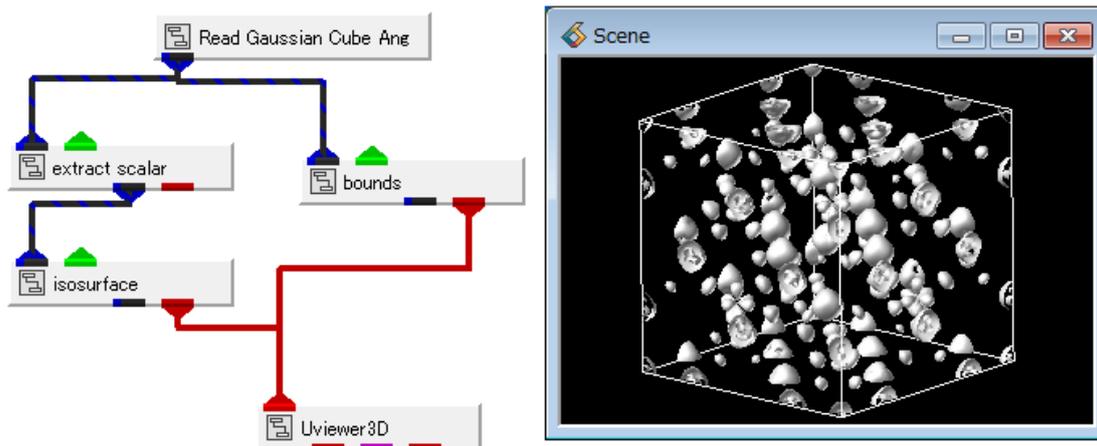


図 100 等値面

パラメーターにある map components をオフにすると、単色で表現できます。等値面を作成している成分、また、異なる成分での色づけも可能です。色づけを行う場合、必要に応じて、set\_minmax モジュールを下に接続し、数値と色の分布を調整してみてください。

### 5.4 サンプル

その他、以下のライブラリに、Gaussian の cube ファイルを例に、コンター図や等値面などを作成するサンプルがあります。あわせてご参照ください。

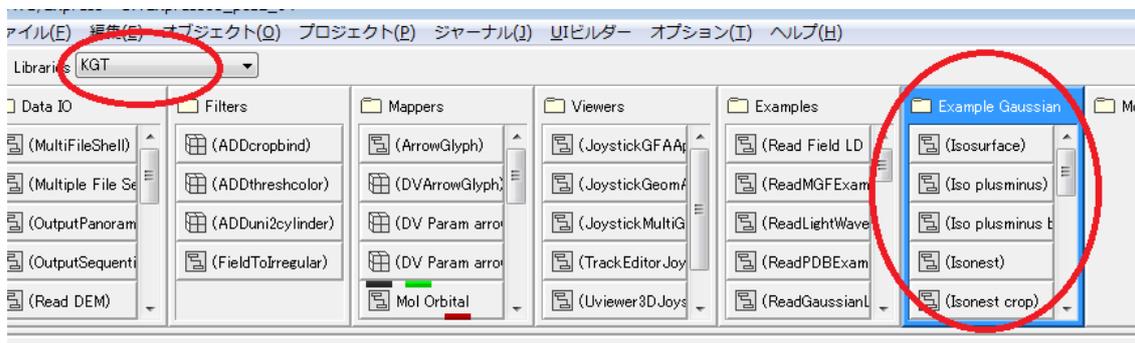


図 101 メッシュデータの可視化サンプル

## 5.5 ボール&スティックとの重ね合わせ

座標空間が同じであれば、メッシュデータの可視化と「第3章 データの可視化（ボール&スティック）」に述べたボール&スティックの可視化とは同時に同じビューの中に表示することができます。

下図は Gaussian の Cube と Log ファイルをそれぞれ読み込み、合成している例を示していますが、Read\_Field モジュールで読み込んだメッシュデータ、Read\_UCD モジュールで読み込んだボール&スティックのデータの場合も、同様に、2つのリーダー・モジュールで読み込んだデータに対して、それぞれ可視化を行い、1つのビューワーで合成できます。

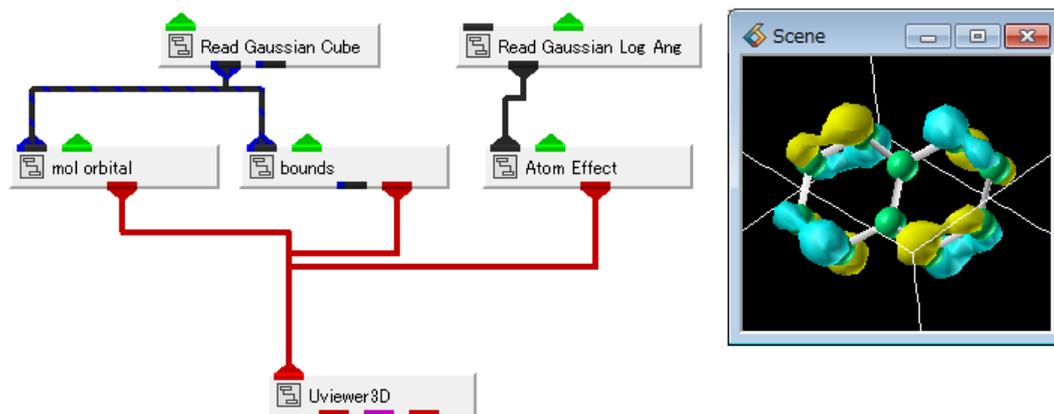


図 102 メッシュとボール&スティックの表示

## 第6章 色、各属性の変更

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。

### 6.1 カラーマップの色の調整

まず、「3.1 ソフトウェア球の表示 (UCD) 「図 35 set\_minmax モジュール」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) ColormapEditor モジュール

このモジュールは、ビューワーに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るモジュールで、その色情報を扱うことができます。

通常のモジュールと異なり、可視化しているモジュールの赤ポートに接続し、以下のように利用します。

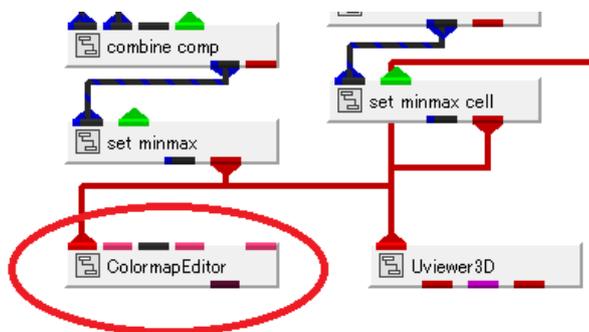


図 103 ColormapEditor モジュール

まず、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。プリセットを選択し、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)。

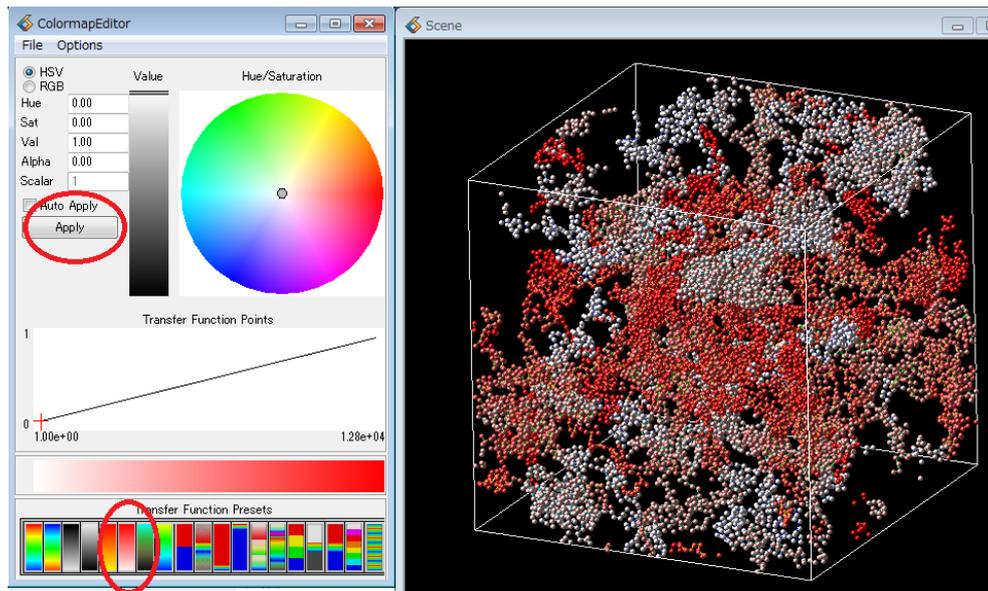


図 104 プリセットを使った変更

図の例では、白から赤のカラーマップを選んでいます。  
デフォルトの青から赤のカラーから、白から赤に変化するカラーマップで表現できます。

このカラーマップの分布は、自分で作成することができます。例えば、上図の白から赤を表示している状態で、カラーマップファイルに保存してみてください。

ColormapEditor の File メニューから Save を選択し、ファイル（拡張子.cmp）に保存します。

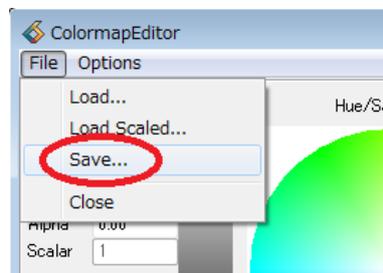


図 105 カラーマップの保存

保存したファイルは、アスキーファイルです。  
メモ帳など、適当なエディターで開いてみてください。

```
2 1 12800          → この数値はご利用のデータによって異なります。
0.000000 0.000000 1.000000 1.000000 1.000000
255.000000 1.000000 1.000000 0.000000 0.000000
```

フォーマットについては後述しますが、まずは、以下の変更を行ってください。

```
3 1 12800          → 先頭の数値を 2 から 3 へ変更
0.000000 0.000000 0.000000 0.000000 1.000000      → 青
127.500000 0.500000 1.000000 1.000000 1.000000    → 白
```

255.000000 1.000000 1.000000 0.000000 0.000000 → 赤

先頭の数値を3に増やし、2行目の後ろから2つ目、3つ目の値を 0.0 に変更します。

※ この後ろの3つが RGB (0 ~ 1) を示しています。  
この2行目 (1つ目) の設定を青にしています。

次に2つ目 (3行目) の中間色を追加し、白に設定しています。  
先頭を 0 ~ 255 の中間の 127.5 に設定し、2カラム目は 0.5 としています。

※ 詳細は後述します。

3つ目 (4行目) 元と同じままで、赤の設定です。

青から白、白から赤に変わるように書き換えています。この書き換えた後の .cmp ファイルを読み込み、適用します。

ColormapEditor の File メニューから Load を選び、作成した .cmp ファイルを指定します。

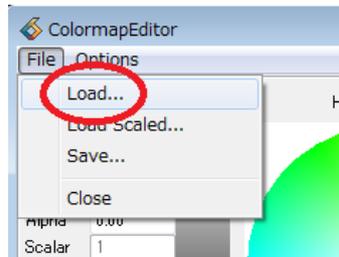


図 106 カラーマップの読み込み

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)。

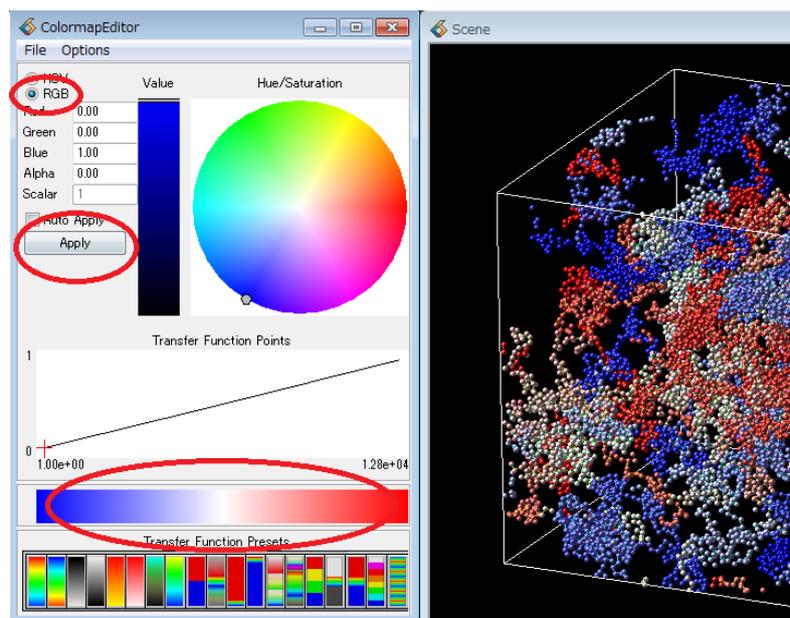


図 107 青白赤のカラーマップ

ColormapEditor のカラーバーが設定した色になっていることを確認し、Apply ボタンをクリックします。青から白、さらに赤に変化する色で色づけできます。

## 2) カラーマップファイルのフォーマット

カラーマップファイルのフォーマットを以下に示します。

再度、先の青から白、白から赤のカラーマップファイルを確認してみてください。

```
3 1 12800
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
```

### ・ 1 行目

まず、1 行目には、カラーマップの制御点の数と、数値の min/max の範囲を記述します。

```
3 1 12800
```

先の例では、3 つの色を設定する制御点があることを示しています。

また、このカラーマップで扱う数値データの範囲が 1 ~ 12800 になっています。

### ・ 2 行目以降

2 行目以降に、カラーマップの制御点に対する設定を記述します。

まず、1 行目の制御点数分、2 行目以降に行を定義します。

先の例では、3 つの制御点があるため、2 ~ 4 行目の 3 行があります。

```
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
```

最初のコラムに、その制御点の位置を指定します。

このコラムは 0 ~ 255 に正規化した値で指定する必要があります。

この例では、左端、中央、右端の 3 点を指定しています。

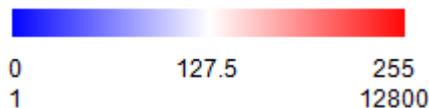


図 108 制御点と数値範囲

1 行目の数値範囲に対して、0 ~ 255 の正規化した値で制御点の位置を決めます。

```
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
```

残りのカラムは、順に、 $\alpha$ 、R、G、B で、0 ~ 1 の範囲で指定します。  
 $\alpha$  は通常の色づけでは利用しません。ボリュームレンダリング用の透明度で利用します。  
上記例では、0、0.5、1 を指定していますが、すべて 0 や 1 でも構いません。  
この  $\alpha$  値は、ColormapEditor の以下のカーブに表現されます。

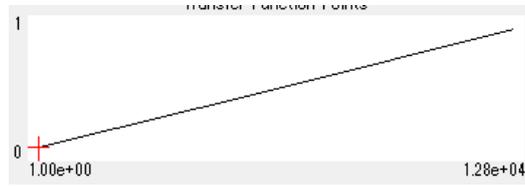


図 109  $\alpha$ カーブ

RGB はそれぞれ色の各成分で、上記例では、(0,0,1) の青、(1,1,1) の白、(1,0,0) の赤を指定しています。

### 3) ColormapEditor パネルの表示非表示

ColormapEditor パネルの以下のアイコンをクリックすると、このパネルが非表示となります。

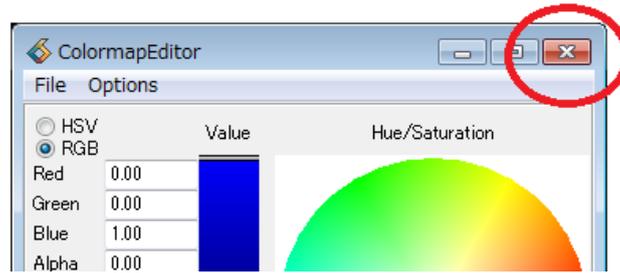


図 110 ColormapEditor パネルの非表示

非表示になったパネルを再表示するには、ネットワークエディタ上のモジュールを開き、**visible** を 1 に戻してください。ColormapEditor モジュールをダブルクリックして開く、もしくは、マウス右メニューからパラメータ表示を選びます。その中にある **visible** をさらに開き、1 を設定 (リターン) します。

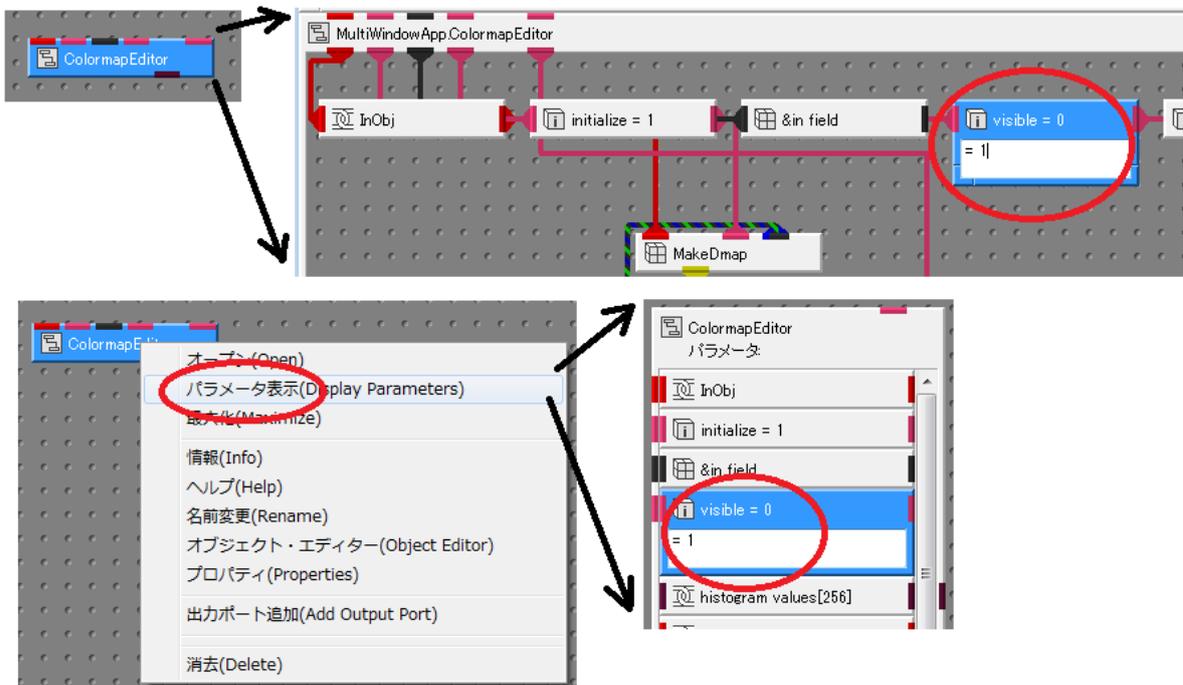


図 111 ColormapEditor モジュールの visible の設定

#### 4) 凡例の表示

色と数値の対応を示す凡例を作成するには、以下のモジュールを利用します。

- LegendHoriz
- LegendVert

以下にネットワーク例を示します。

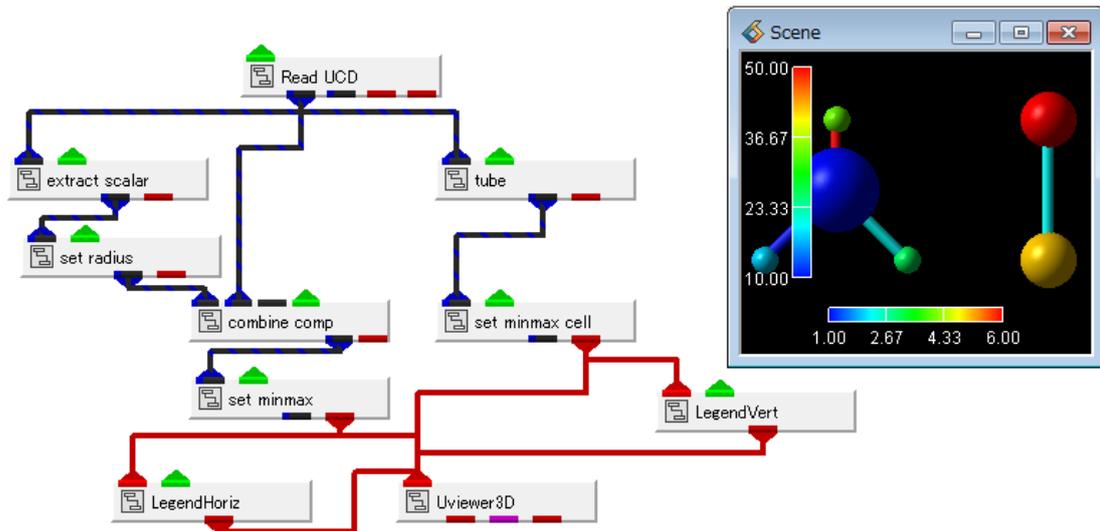


図 112 凡例表示

LegendHoriz モジュールは横向きの凡例を、LegendVert モジュールは縦向き  
の凡例を作成します。画面に表示している赤ポートを入力し、出力をビュー  
ワースと接続します。

パラメーターで位置やフォントのフォーマットなどを変更できますので、  
適宜、調整してみてください。

## 6.2 オブジェクトの属性の変更（色や反射）

オブジェクトを単色で表現する場合、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、オブジェクト・エディターを利用します。

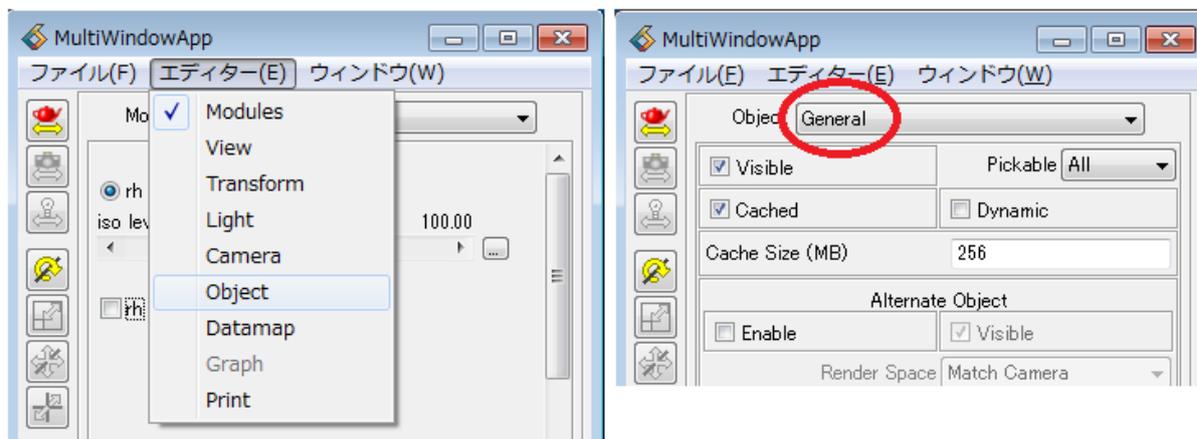


図 113 Object エディター

エディター・メニューから **Object** を選びます。

さらに、**Object** メニュー（右図）の中に、機能毎にパラメーターが分類されています。

オブジェクト・エディターを利用するには、最初に対象のオブジェクト（カレント・オブジェクト）を選択する必要があります。

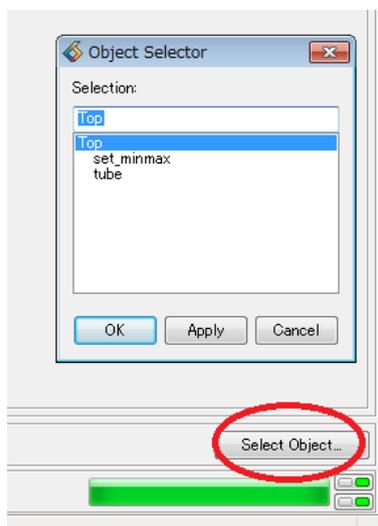


図 114 カレント・オブジェクトの選択

**Select Object** ボタンをクリックし、開いたパネルで、対象を選択します。

上図では、**tube** モジュールが単色で表現されている場合の例を示しています。**tube** を選び、**Apply** ボタンをクリックします。

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/Saturation/Value のダイアルを使って、その対象に色づけできます。

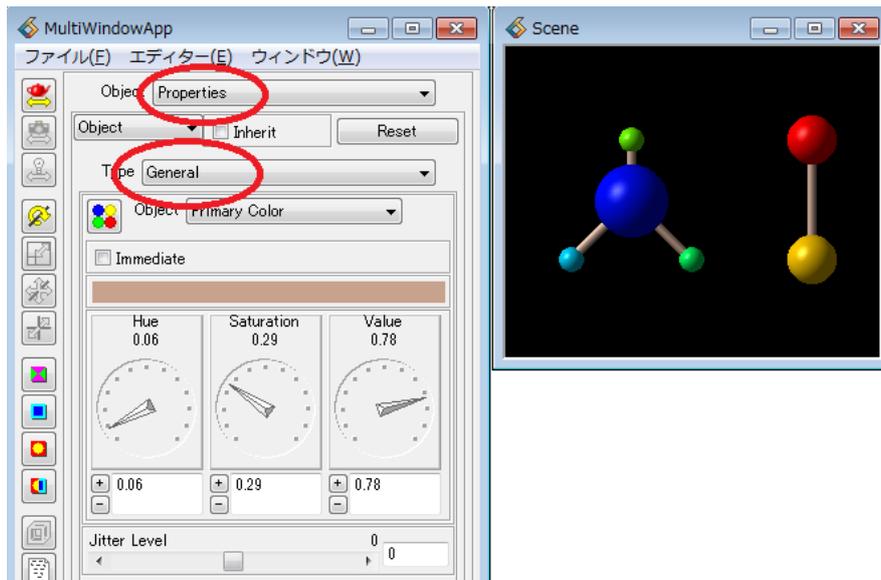


図 115 オブジェクトの色づけ

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があります。

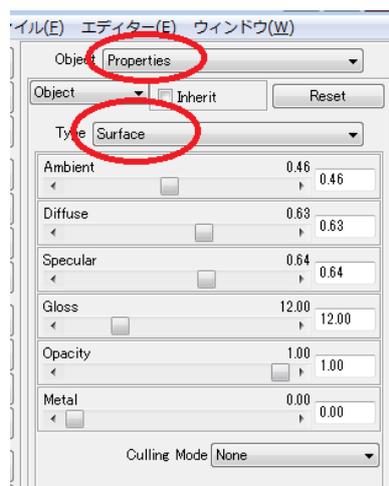


図 116 面の属性

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、以降に述べるカメラのソート機能もあわせてご参照ください。

### 6.3 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) 分割数、ラインの幅、チューブ表示

set\_radius モジュールによる球の OpenGL 表示（「3.3 ポリゴン分割球」）や「3.5 その他の形状」で述べた点表示、「3.7 スティックの表示（ライン形状）」に述べたライン表示に対して、その分割数や線幅（ピクセル幅）を設定できる機能です。

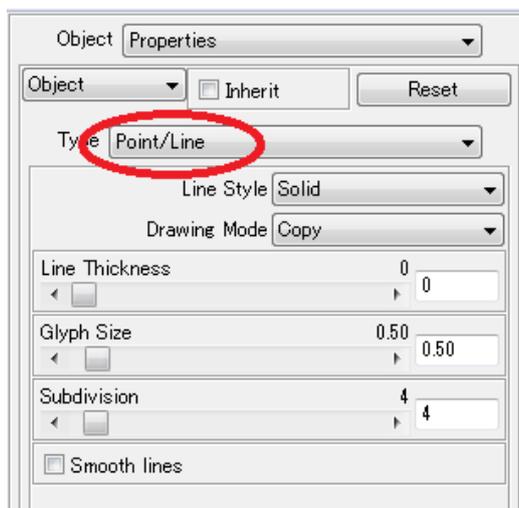


図 117 点、ラインなどの属性

また、以下のパラメーターを使うと、「3.6 スティックの表示（チューブ形状）」の代わりに、OpenGL の機能を使って、チューブ表示に変更できます。

その際は、「3.7 スティックの表示（ライン形状）」によるライン表示を行い、以下の設定と上図の Subdivision の属性で分割数を設定します。

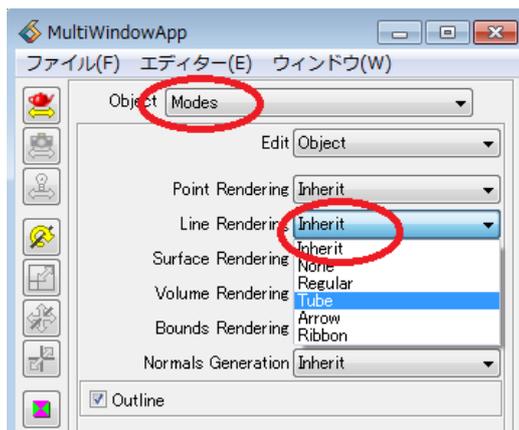


図 118 ラインのチューブ表示

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。

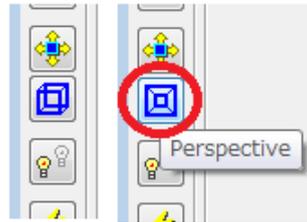


図 119 投影方法の変更

## 3) ライトの調整

デフォルトでは、ライト（照明）は、片方向からある角度で照射しています。

陰影によって暗い部分が多いような場合、双方向ライトに変更することで、見やすくなる場合があります。特に、断面コンター図などでは、面とライトの向きによっては、暗く表示されてしまいますので、そのような場合には、以下のアイコンを切り替えてみてください。



図 120 双方向ライト

このアイコンをオンにすると、デフォルトの片方向のライトに対して、反対側からのライトがオンの状態となり、両方から照射するライトとなります。

また、デフォルトでは、マウス操作は、オブジェクトの幾何変換を行うようになっています。このマウス操作をライトの回転操作に変更することができます。

下図（デフォルトでは一番上がオン）の幾何変換のモードを3つ目のライトに変更します。その後、画面上でマウス左を押したまま移動すると、ライトの向きを回転させることができます。

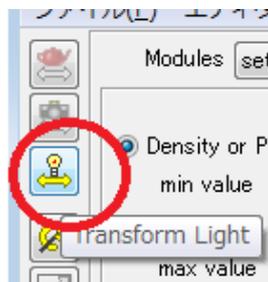


図 121 ライトの幾何変換

## 4) 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のラインを重ねるなど）、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。

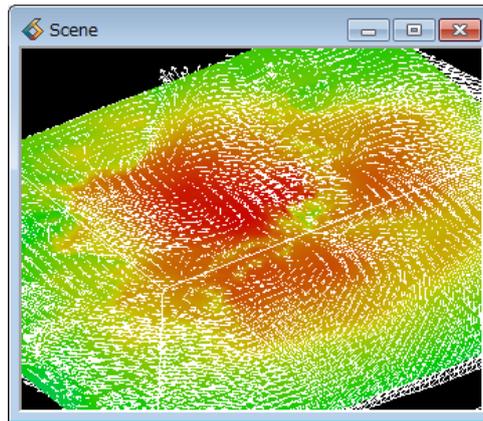


図 122 面にラインが埋もれた例

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。

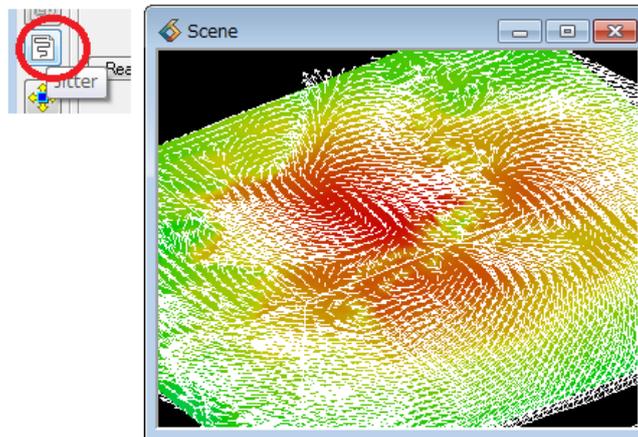


図 123 ジッター機能

##### 5) 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します。通常、AVS/Express では、半透明の表示に対して、速度優先のため、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。

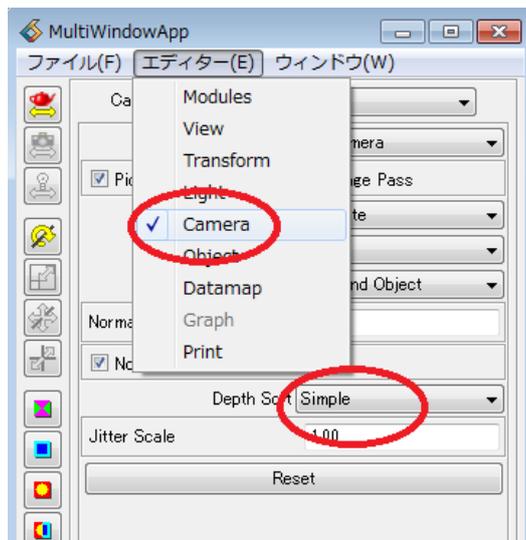


図 124 ソート機能

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。

## 第7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。

### 7.1 時系列フォーマット

まず、「第2章 データの読み込み」に述べた Field データや UCD データには、時系列フォーマットがあります。また、「2.5 一般的なフォーマット」で述べた Lammmps データの読み込みモジュールは、時系列の dump ファイルに対応しています。

Field と UCD ファイルの時系列ファイルを読み込んだ場合、その読み込みモジュールのパラメーターに、ステップ数と前後に移動するボタンなどのパラメーターが表示されます。

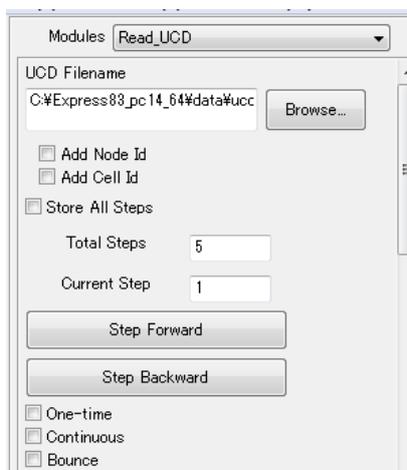


図 125 UCD 時系列ファイルの読み込み

Step Forward ボタンをクリックすると、時間ステップを進めることができます。

アニメーション等を作成したい場合は、One-time にチェックすると、全ステップの実行を自動的に1度、行います。

また、下図は Read\_Lammmps モジュールで dump ファイルを指定した場合の例です。

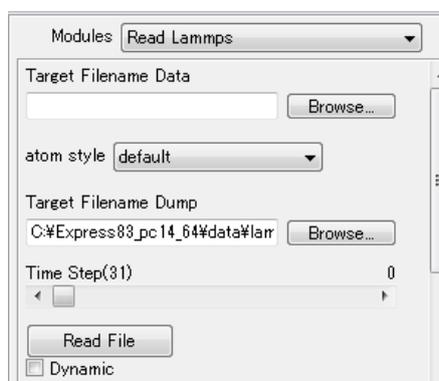


図 126 Lammmps の dump ファイルの読み込み

Read\_Lammmps モジュールの場合は、Time Step (全ステップ数) とスライダーが表示されます。スライダーで読み込みたいステップを選択し、Read File ボタンをクリックすると、そのステップ

のデータを読み込むことができます。

また、Dynamic チェックは、Read File ボタンをクリックする操作を省略することができる機能です。この Dynamic にチェックしている場合、Time Step のスライダーを動かすと同時に、データの読み込みが行われます。

Read\_Lammps モジュールには、自動で全ステップを実行する機能はありません。「7.3 Loop モジュール」で述べる Loop モジュールと一緒に利用してください。

## 7.2 Multi Files モジュール

このモジュールは、連番ファイルなどの複数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。

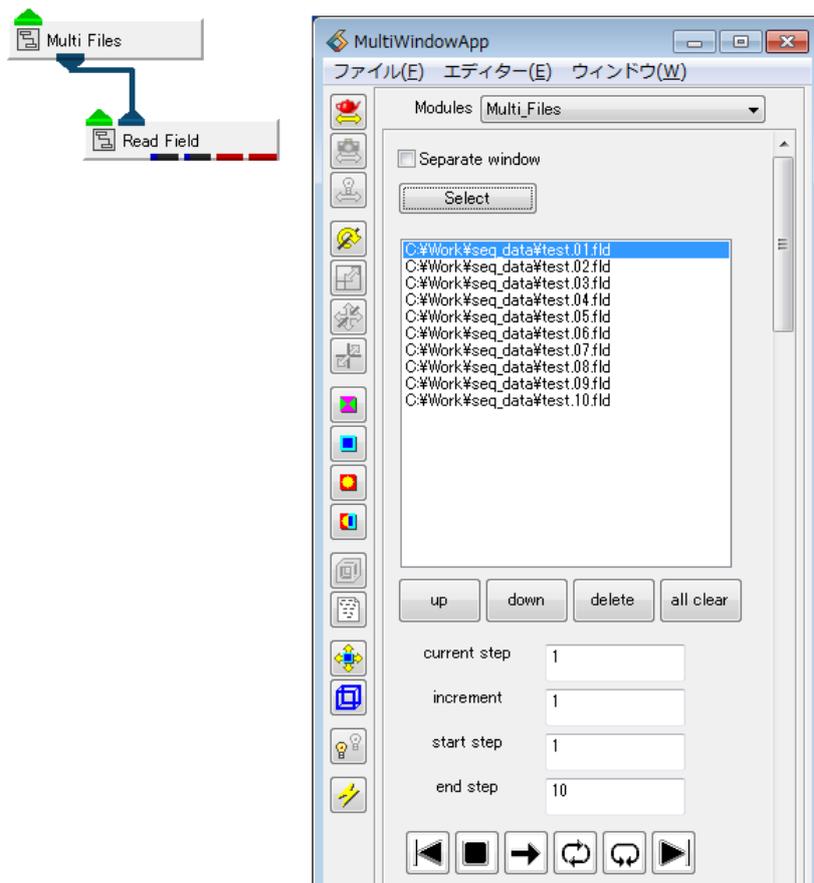


図 127 Multi\_Files モジュール

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。  
以下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を押して表示されるメニューから、パラメータ表示を選びます。

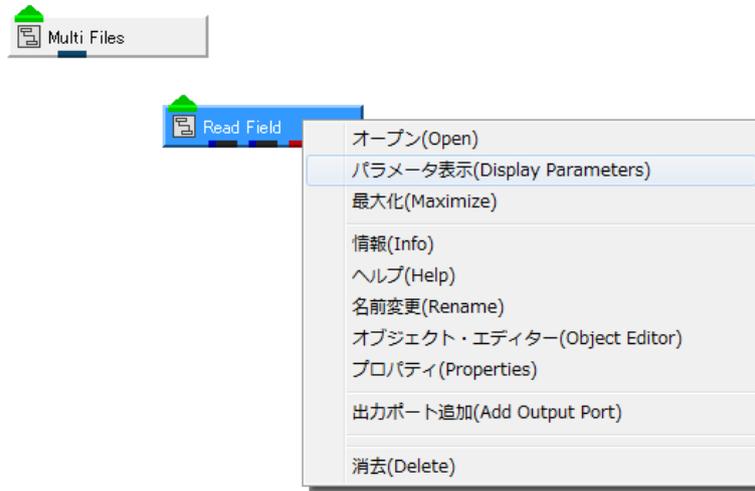


図 128 Read\_Field モジュールのパラメータ表示

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。

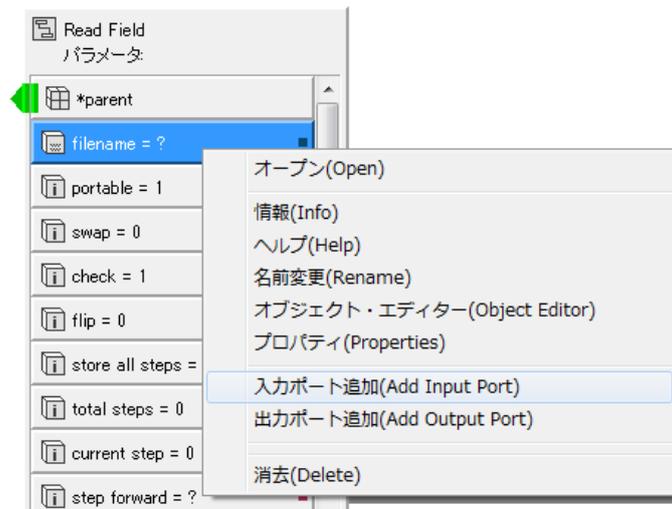


図 129 filename の入力ポート追加

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。

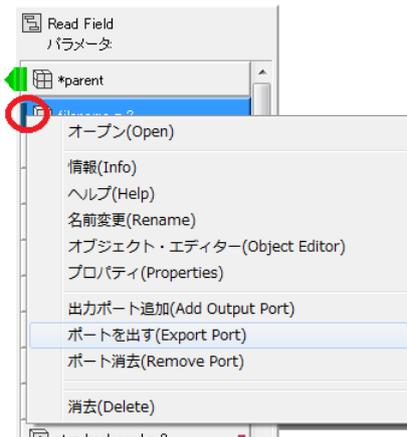


図 130 filename の入力ポートの表示

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。

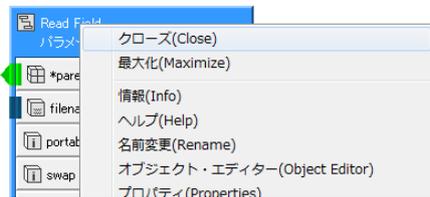


図 131 モジュールのクローズ

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。

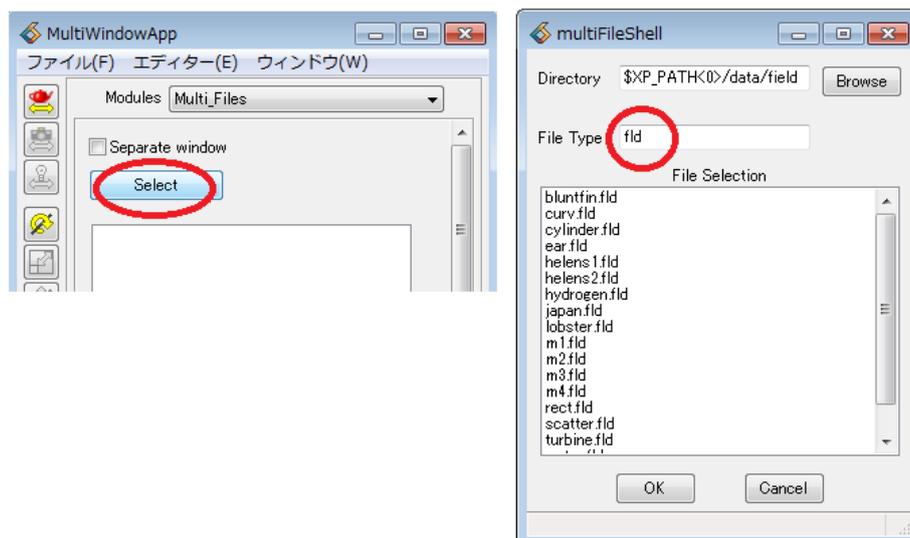


図 132 Multi\_Files モジュールの Select

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 132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。

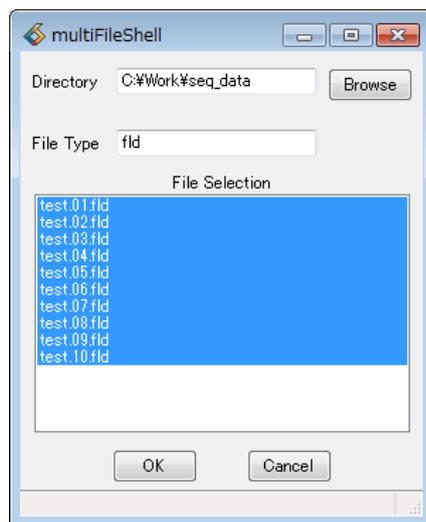


図 133 ファイルリストの選択

最初の「図 127 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。

### 7.3 Loop モジュール

Loop モジュールは、ループ処理を行い、そのカウント値を自動生成するモジュールです。Read\_Lammps モジュールの dump ファイルのように 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している場合に、そのステップ値を自動生成できます。

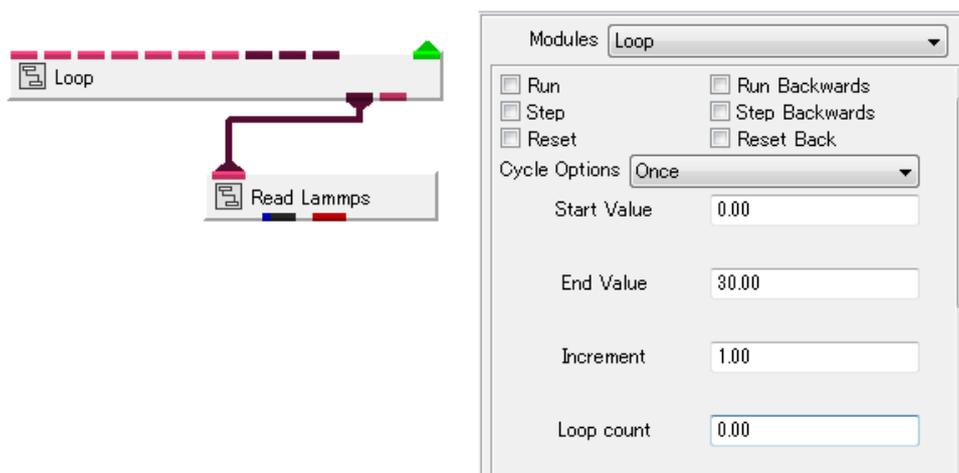


図 134 Loop モジュール

Loop モジュールは、例えば、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、上図右は、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。

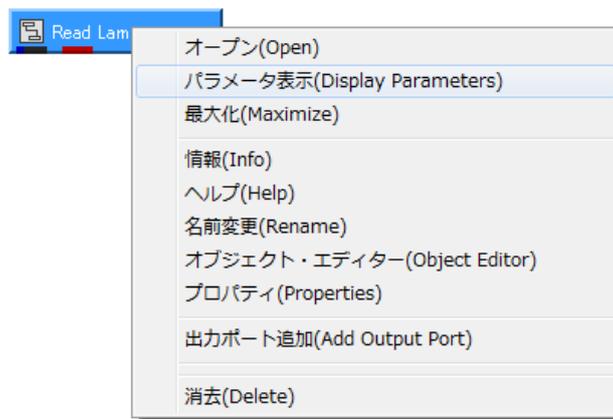


図 135 Read\_Lammps モジュールのパラメーター表示

current\_step 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。

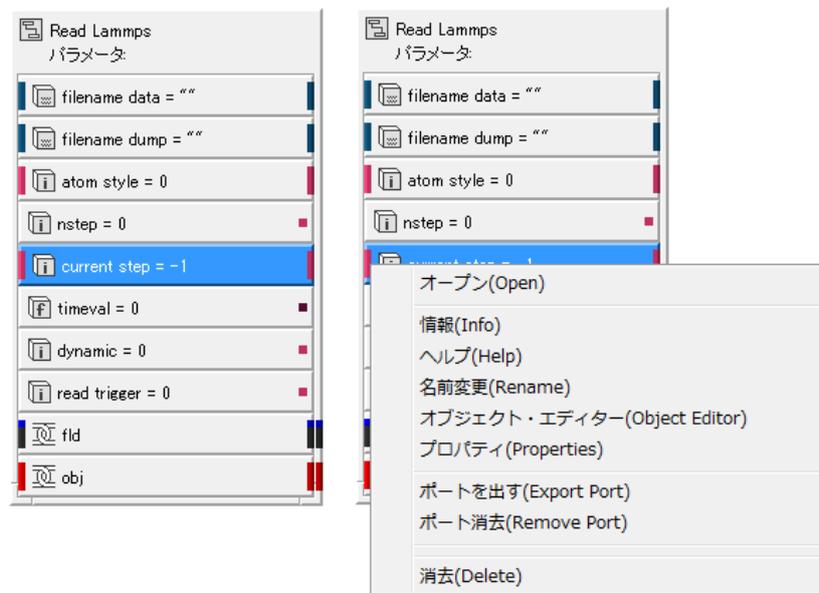


図 136 current\_step のポートの表示

「図 134 Loop モジュール」のように、Loop モジュールの count とこの current\_step を接続します。

Read\_Lammps モジュールの Time Step を確認し、そのステップ数を Loop モジュールの Start/End Value に設定します。また、Increment を 1 に設定（1 ステップずつ）します。

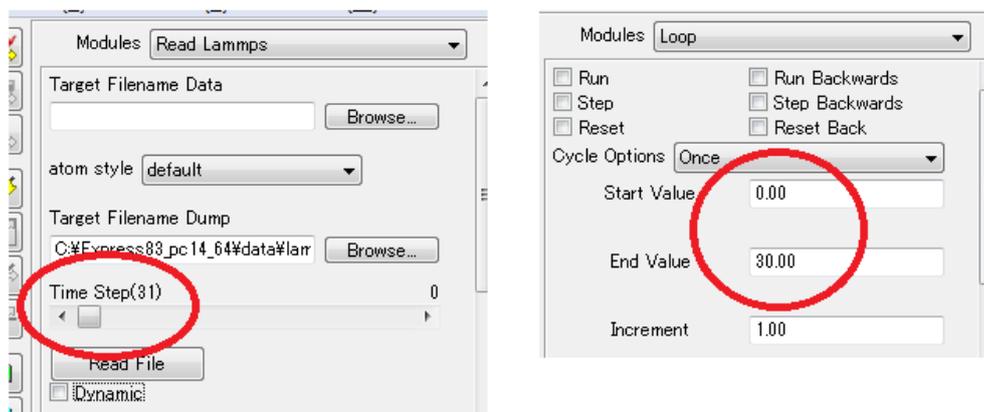


図 137 ステップ数の確認

実行するには、Read\_Lammps モジュールのパラメーター Dynamic にチェックし、その後、Loop モジュールの Run にチェックします。

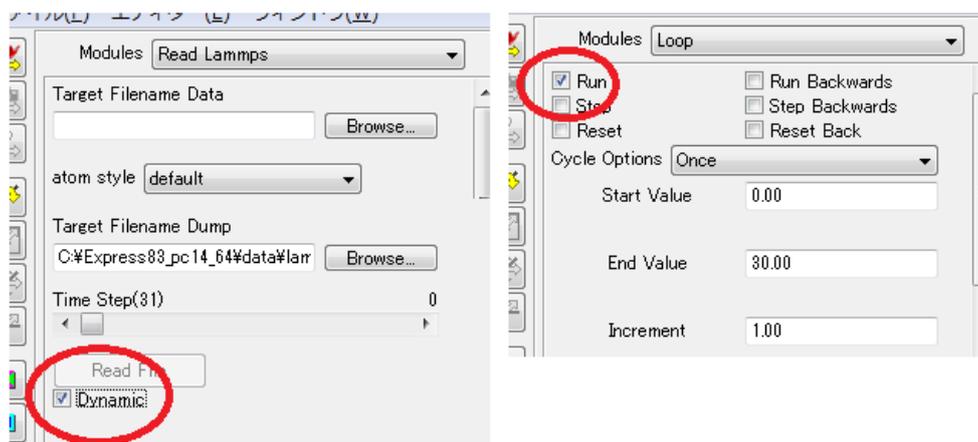


図 138 Loop の実行

※ Read\_UCD モジュールなどでも同様に、Loop モジュールと接続し、Loop のカウント値でステップを進めることができます。

Read\_UCD モジュールのパラメーターを開き、current\_step と接続します。

※ Read モジュールだけでなく、例えば、断面位置などの各モジュールのパラメーターを自動的に変更したい場合にも利用できます。

そのパラメーターを持つモジュールの中を開き、パラメーターのオブジェクトを探し、接続してみてください。

## 第8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。

### 8.1 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。

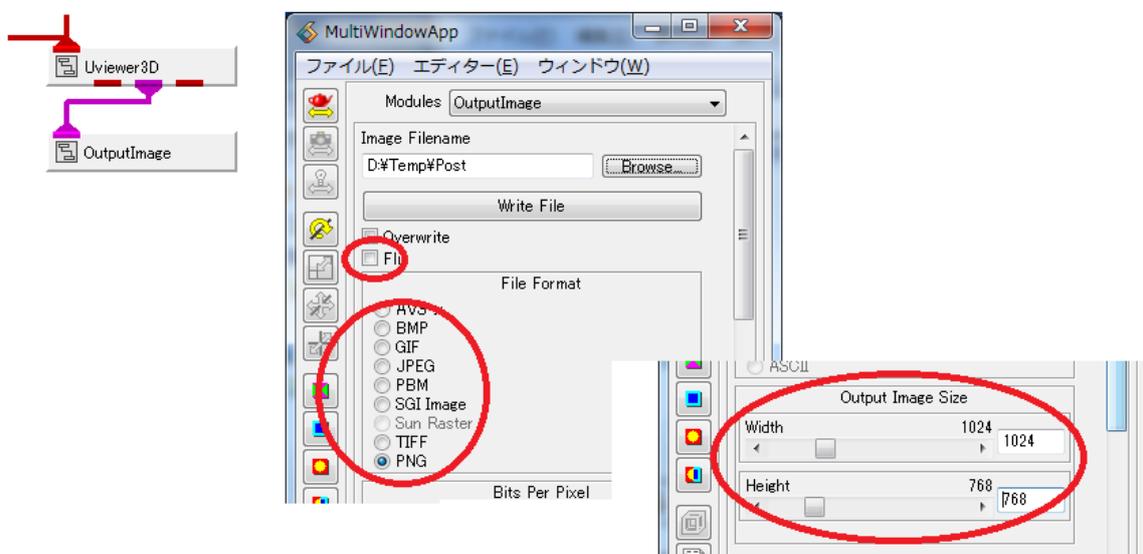


図 139 OutputImage モジュール

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。

このモジュールは画面サイズよりも大きい解像度の画像を作成できるように、ソフトウェアレンダラでレンダリングしています。

表示画面のレンダラがソフトウェアレンダラの場合は、同じ表示となりますが、ハードウェアレンダリングを行っている場合、そのレンダラの違いで出力した画像との違いが生じます。

Developer 版をご利用の場合には、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（ドットで区切ったオブジェクトを順に開きます）の値を OpenGL に変更します。

ただし、OpenGL の場合、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。

Viz 版の場合、このレンダラの変更はできません。

その他、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 140 クリップボードへのコピー

## 8.2 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。

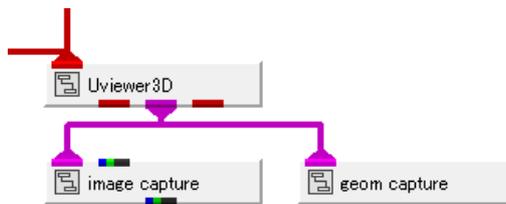


図 141 capture モジュール

### 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。

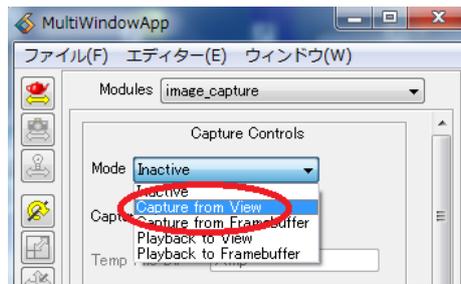


図 142 キャプチャの開始

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。

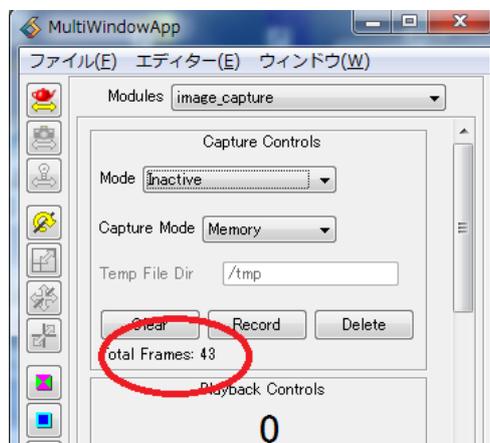


図 143 キャプチャの終了とフレーム

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。

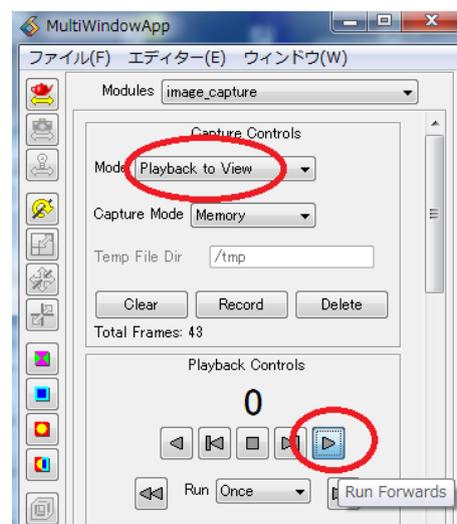


図 144 プレビュー

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。

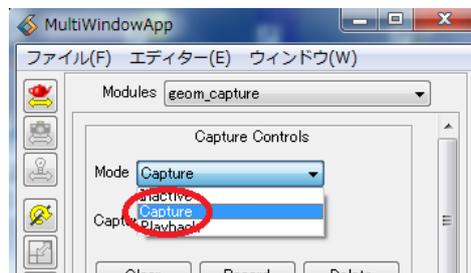


図 145 キャプチャの開始 (geom\_capture)

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）。

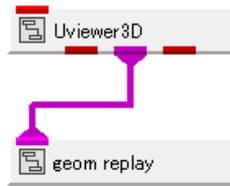


図 146 geom\_replay モジュール

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/>

### 8.3 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。

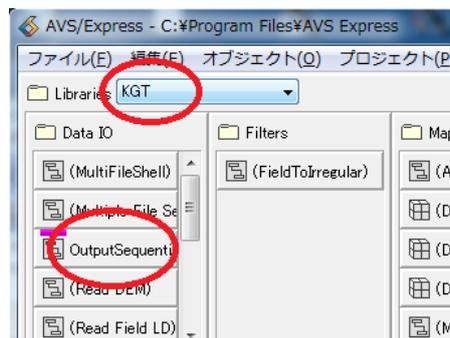


図 147 OutputSequentialImage モジュール

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。

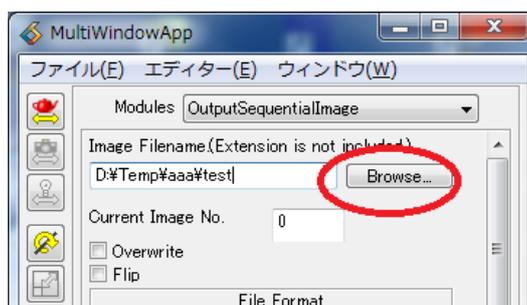


図 148 出力フォルダと名前の指定

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。

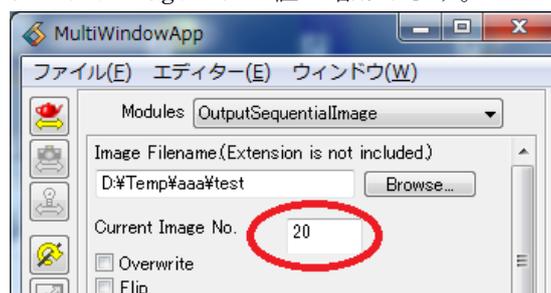
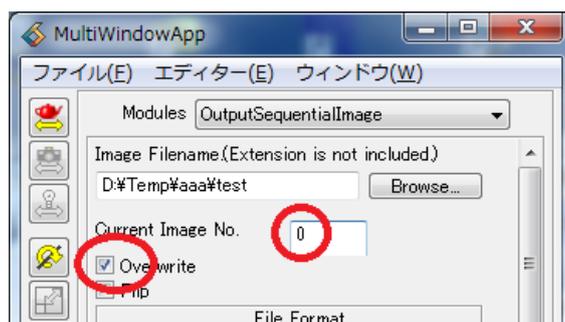


図 149 イメージ番号

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でき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、8.1 で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。

#### 8.4 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。

「8.2 動画の保存」で述べた 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。

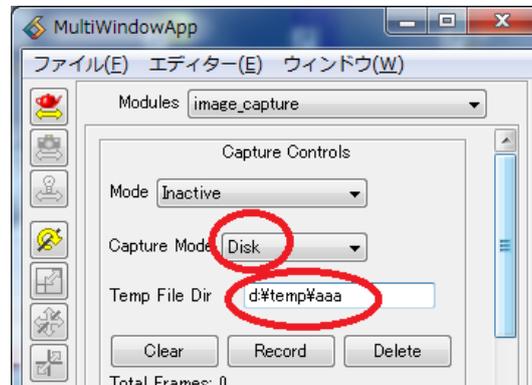


図 150 Disk モードの指定

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 による材料データの 3次元可視化

-----  
2016年7月12日 第1版

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