AVS 入門 (2)

伊東栄典[†]

1　AVS へのデータの入力

近年，科学や工業技術の分野では流体解析，分子設計などさまざまな解析が計算機を用いて行なわれています．AVS (Application Visualization System) はこのような解析において，可視化部分を支援してくれるシステムです．AVS は可視化システムという日本語名からもわかるように，さまざまなアプリケーションから出力されるデータを，目に見える形に描画してしてくれます．AVS におけるデータ可視化の概念や，九州大学大型計算機センターにおける AVS の利用開始方法については，文献 [1] で説明いたしました．本稿では AVS 入力するデータについての解説を行ないます．

AVS に可視化をさせるには，可視化させたいデータを AVS に入力しなければなりません．測定器からのデータや解析計算結果データを用いて，AVS が画像を作成するからです．データの測定や解析作業などは研究者や利用者が自分で行なう必要があります．この測定や解析した手持ちのデータを AVS に入力するためには，2 つの方法があります．

(1) 手持ちのデータを AVS の標準フォーマットに変換する．
(2) 手持ちのデータを読み込むことができるようなモジュールを作成する．

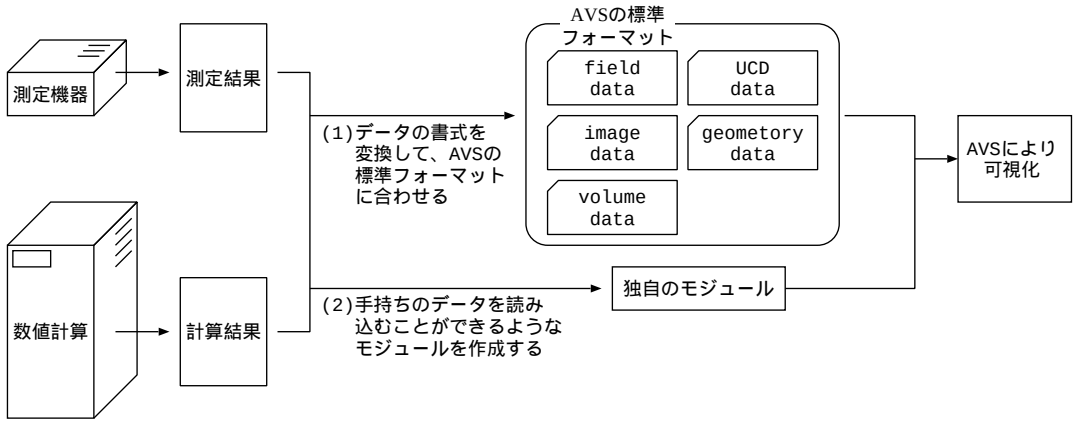

図 1: AVS へのデータ入力

今回は (1) の AVS の標準フォーマットについて説明します．

2　AVS へ入力できる標準のデータ ・フォーマット

AVS が入力できる標準データフォーマットには以下の 5 つがあります．

1. フィールド ・データ (**Field Data**)
   AVS で最も頻繁に利用されるフォーマットです．流体解析などで用いられる差分格子のような規則正しく並んでいるデータに利用します．各格子点のデータ成分数，次元数等には特に制限がありません．

[†]九州大学大型計算機センター 研究開発部　E-mail:itou@cc.kyushu-u.ac.jp

フィールド ・データの読み込みは [read field] モジュールを用いて行ないます .

2. イメージ ・データ (**Image Data**)
3. ボリューム ・データ (**Volume Data**)
   イメージおよびボリューム ・データは , フィールド ・データの一種です . 使用頻度が高いため , フォーマットが定義されています . [read image], [read volume] モジュールはこれらのフォーマットで記述されているデータを読み , フィールド ・データに変換する機能を持つモジュールです .
4. **UCD データ (Unstructured Cell Data) :** 非構造格子データ
   非構造格子型のデータ ・フォーマットを表わし , 各セルと節点 (node) から構成されます . このフォーマットは , 有限要素法の解析結果のデータ等を表現するために利用されます . UCD フォーマットのデータは [read ucd] モジュールを用いて AVS に取り込む事ができます .
5. ジオメトリ ・データ (**Geometry Data**)
   3 次元の幾何データを表現するフォーマットです . フィールド , UCD データを 3 次元的に可視化する場合 , マッパー ・モジュールを通して , このジオメトリ ・データ型に変換します . また , 著名な他のアプリケーションの出力ファイルをジオメトリ ・データに変換するモジュールも用意されています .

次に各データ ・フォーマットについて説明します . ジオメトリ ・データについては次号で説明する予定です .

2.1 フィールド ・データ (Field Data)

フィールド ・データは , AVS で最も頻繁に利用されるフォーマットです . フィールド ・データはファイルに記述されます . フィールド ・データが入っているファイルの名前は必ず “.fld” という拡張子で終了していなければなりません . フィールド ・データは , ヘッダー部 , データ部 , 座標情報部の 3 つの部分から成ります . これらのデータをファイルに記述する場合 , 図 2 に示すように 1 つのファイルにデータを格納するネイティブ ・フィールド入力形式と , 3 つのファイルに格納するデータ変換入力形式があります . データ部のみを変更する場合などに対応が容易であるため , データ変換入力形式の方が扱い易い形式になっています .

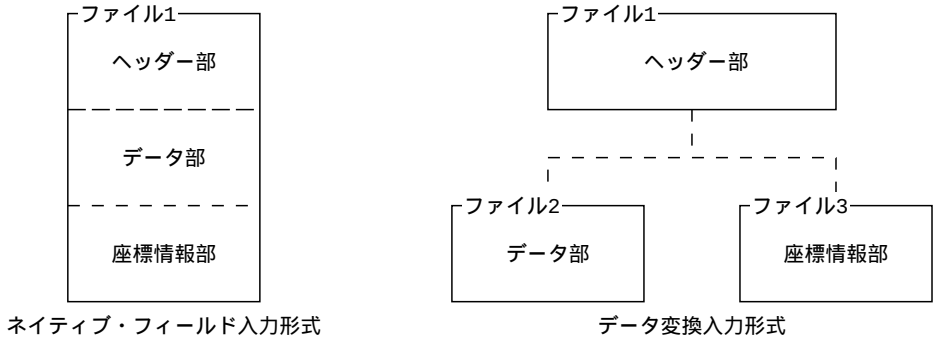

図 2: 各入力形式の概要

まず , ヘッダー部について説明し , その後にネイティブ ・フィールド入力形式およびデータ変換入力形式について説明します . データ変換入力形式だけ知りたい方は , ヘッダー部の説明 (**2.2** 節) とデータ変換入力形式の説明 (**2.4** 節) を参考にして下さい .

2.2 フィールド ・データのヘッダー部

ヘッダー部は，図 3 に示すように，コメントや次元数，次元サイズがどのような値も持つのかという情報を，アスキー文字で記述します．値は [項目名 = 値] という形で記述されます．

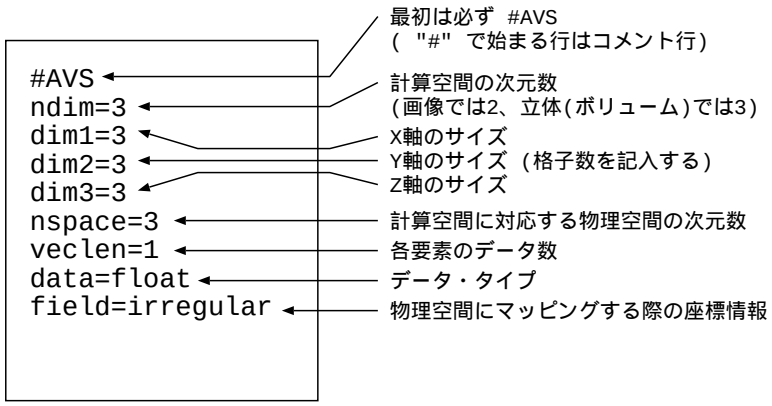

図 3: ヘッダー部

ヘッダー部における項目の記述について説明します．

(1) `# AVS`
必ずこの記述から始まります．”#”で始まる行は注釈行として扱われます．ファイル内に説明などを記述しておきたい場合に用います．

(2) `ndim` :計算空間
計算空間の次元数を表わします．2 次元のデータなら，`ndim=2` ですし，3 次元空間ならば `ndim=3` になります．

(3) `dim`N :各軸の次元数を表わします．ここで，軸の数は計算空間 (`ndim`) の値に依存します．`ndim=3` の場合，`dim1`,`dim2`,`dim3` まで記述します．

(4) `nspace` :物理空間
データを座標軸に配置するために必要となる座標の次元数を表わします．3 次元空間を表現する場合は，`nspace=3` になります．

(5) `veclen` :ベクトル長
各点で持つ要素数を表わします．

(6) `data` :データ型
ここには以下に示す 5 つの基本データ型のうち，いずれかを記述します．

`byte`	:	1byte
`short`	:	整数型, 2byte
`integer`	:	整数型, 4byte
`float`	:	単精度実数型, 4byte
`double`	:	倍精度実数型, 8byte

(7) `field` :フィールド
次の 3 つから選択して記述します．

uniform	:	直交等間隔格子．各格子の間隔が等しいため，座標情報部を記述する必要はありません．
reactilinear	:	直交不等間隔格子．3 次元の場合，(dim1 + dim2 + dim3) 個のデータを持ちます．
irregular	:	非直交不等間隔格子．制約なしのデータ．3 次元の場合，(dim1 × dim2 × dim3) 個のデータを持ちます．

計算空間ndim, 物理空間nspace, ベクトル長veclen の関係を図 4 の例で示します．ここでは，薄膜上の解析を例に考えます．薄膜は 3 次元空間に存在するので，物理空間はnspace=3 となります．しかし解析の計算は膜という平面上 (2 次元空間) で考えるので，計算空間の値はndim=2 となります．薄膜上の各点で温度と圧力の 2 つの要素を考慮する場合，veclen=2 となります．

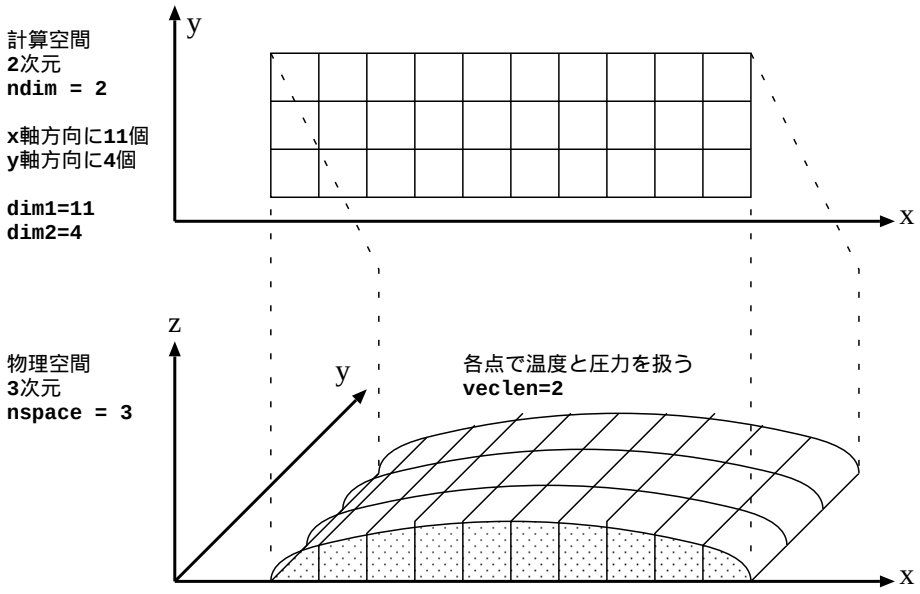

図 4: 計算空間，物理空間，ベクトル長

2 次元空間の場合における直交等間隔格子 (uniform), 直交不等間隔格子 (rectilinear), 非直交不等間隔格子 (irregular) の概念を図 5 に示します．

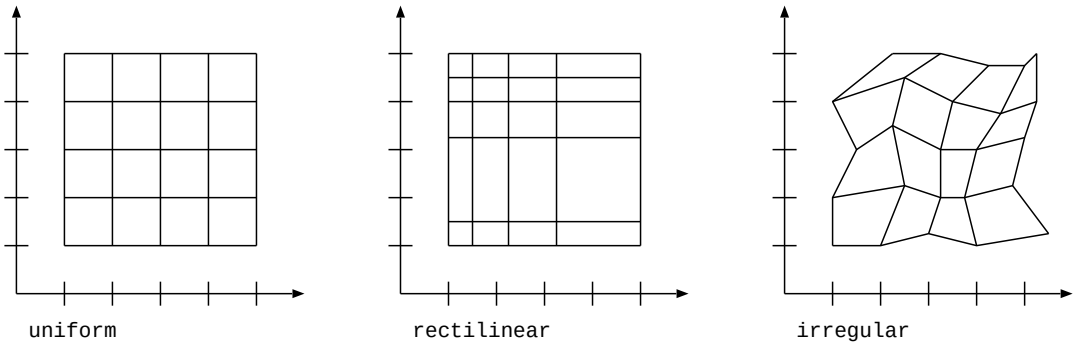

図 5: uniform, rectilinear, irregular

2.3 ネイティブ ・フィールド入力形式

ネイティブ ・フィールド入力形式は，AVS フィールド ・データの内部フォーマットの形式で作成されたデータ型です．この形式の構造は図 6 に示しているように，ヘッダー部，データ部，座標情報部を 1 つのファイル内に記述します．ヘッダー部はアスキー文字で記述しますが，データ部と座標情報部はバイナリ形式で記述します．ファイルの名前には必ず “.fld” で終る拡張子が付いていなければなりません．

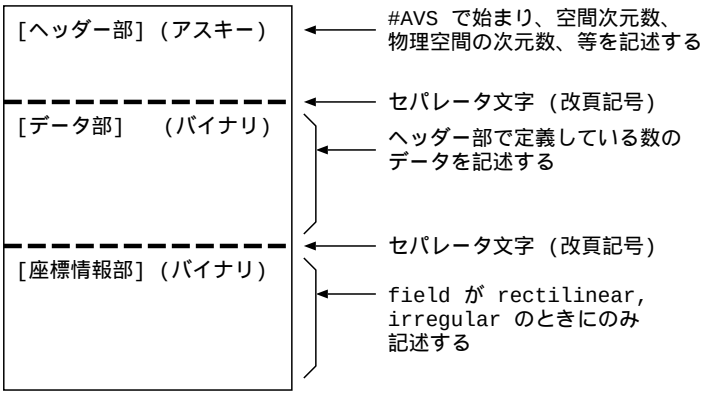

図 6: ネイティブ ・フィールド入力形式のフォーマット

2.3.1 データ部におけるデータの順序

ネイティブ ・フィールド入力形式におけるデータ部について説明します．データ部には，各点でのデータがバイナリ形式で記述されます．データ部の要素数はヘッダー部の定義に依存しており，以下の式で表わされる数のデータが記述されます．

$$\text{データ部の要素数} = \texttt{veclen} \times \texttt{dim1} \times \texttt{dim2} \cdots \times \texttt{dim}N$$

データは図 7 のような順序で記述します．

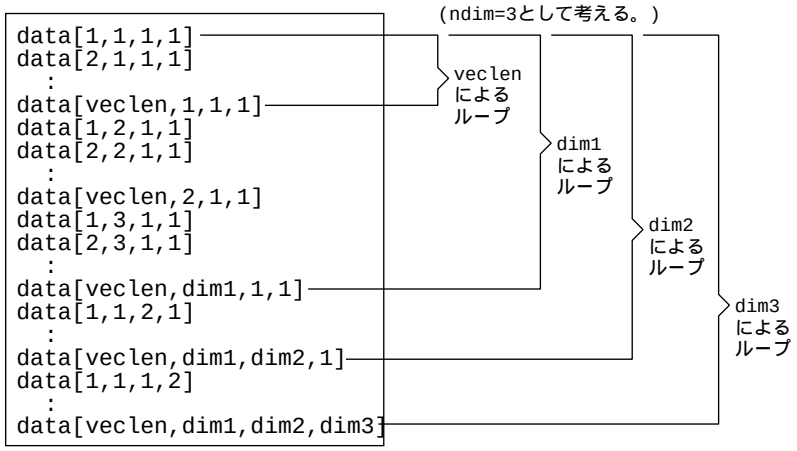

図 7: データの記述順序

記述されるデータの順番を，Fortran と C 言語のプログラムで用いる配列で考えると，以下のようになります．

- Fortran での配列表現
  `DIMENSION DATA(veclen, dim1, dim2, dim3)`
- C での配列表現
  `float data[dim3][dim2][dim1][veclen];`

2.3.2 座標情報部の説明

座標情報部は，ヘッダー部における field の値が rectilinear および irregular のときのみに記述します．
計算空間が 3 次元 (nspace=3) である場合を考えます．nspace=3 ですのでdim1,dim2,dim3 の値を指定します．

- rectilinear の場合は直交しているので X 軸方向にdim1 個の座標，Y 軸方向にdim2 個の座標，Z 軸方向にdim3 個の座標，全体でdim1+dim2+dim3 個の座標情報を記述します．
- irregular の場合は全ての点について座標を指定する必要があるため，dim1×dim2×dim3 個の座標情報を記述します．

座標情報部を記述する際は，座標を記述する順番に注意して下さい．図 8 に示しているように，最初に X 座標，次に Y 座標，最期に Z 座標，というような順番で記述します．

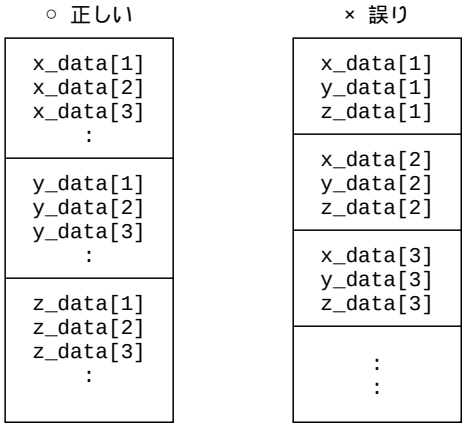

図 8: 座標情報部におけるデータの並び

2.4 データ変換入力形式

一般ユーザが AVS にデータを取り込む際に最も利用しやすいデータ変換入力形式について説明します．この形式では，ヘッダー部，データ部，座標情報部はそれぞれ別のファイル記述されます．座標はそのままでデータだけを変更して可視化するような場合に使いやすい形式です．図 9 に，データ変換入力形式の概要を示します．ヘッダー部が格納されているファイルの名前には，必ず “.fld” で終る拡張子が付いていなければなりません．

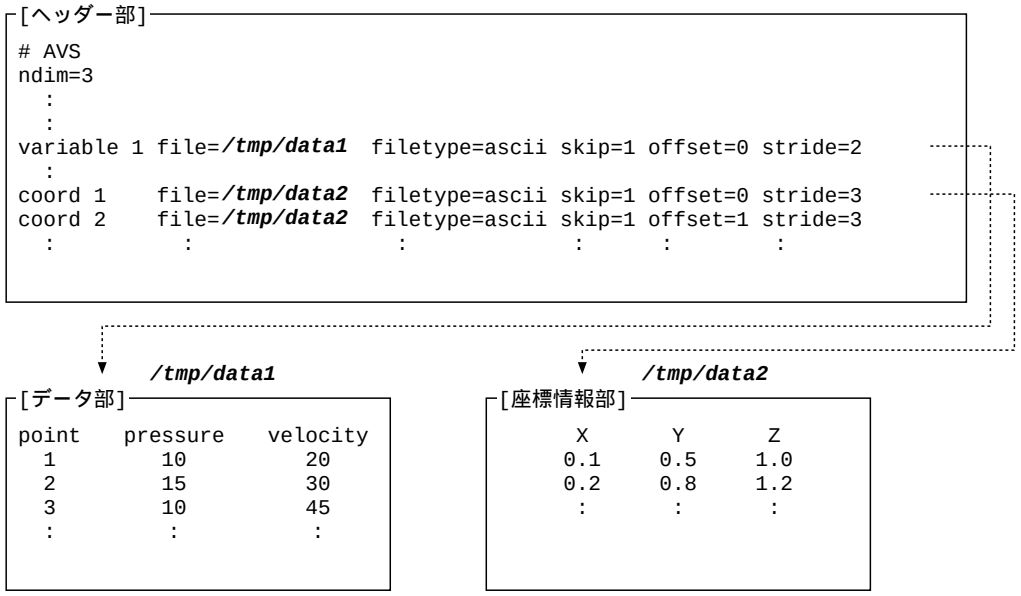

図 9: データ変換入力形式

図 9 に示しているように，データ部と座標情報部の内容に応じて，ヘッダー部に様々な指定を記述することができます．データ変換入力形式のヘッダー部には，2.2 節で説明した項目の他に，variable とcoord という項目が増えています．variable 項目はデータ部の情報を表現しており，veclen 項目で指定する数と同じ行数のvariable 項目が必要です．coord 項目は座標情報部についての情報を表現しています．nspace 項目で指定した数と同じ行数のcoord 項目が必要です．

図 9 に記述しているように 2 つの項目には，複数の副項目があります．それぞれの副項目について以下で説明します．

- variable 項目の副項目 :(n, file, filetype, skip, offset, stride)

  n : 何番目の副項目であるかを表現します．1 からveclen までの値を取ります．

  file : ファイル名を絶対パスで記述します．
  図 9 の例では “/tmp/data1” というファイルを指定しています．

  filetype : ファイルのタイプが，アスキーかバイナリであるのかを指定します．
  アスキー (文字ファイル) の場合 : “filetype=ascii”
  バイナリの場合 : “filetype=binary”

skip : 最初に読み飛ばす量を設定します．
filetype=ascii の場合 : 最初の何行を飛ばして，データを読みこむのかを指定します．
filetype=binary の場合 : 最初の何バイトを飛ばして，データを読みこむのかを指定します．
(※ デフォルトは 0 になっています．)

offset : “filetype=ascii” の場合にのみ記述します．データが記述されている項目を何個読み飛ばすかを指定します．
(※ デフォルトは 0 になっています．)

stride : 途中で読み飛ばす量を指定します．
filetype=ascii の場合 : 何項目ごとにデータを読み込むのかを指定します．
filetype=binary の場合 : データサイズ (byte,short,float,double,integer)$\times s$ ごとにデータを読み込むように指定します．ここでstride=s とします．
(※ デフォルトは 1 になっています．)

- coord 項目の副項目 :(n, file, filetype, skip, offset, stride)
  それぞれの意味はvariable の場合と同じです．

具体例を用いて副項目の説明を行ないます．図 10 に示しているファイルが，データ部を表わしているとします．AVS での可視化に，pressure の値だけを用いる場合は，variable 項目を以下のように設定します．

```
variable 1 file=/tmp/data1 filetype=ascii skip=1 offset=1 stride=2
```

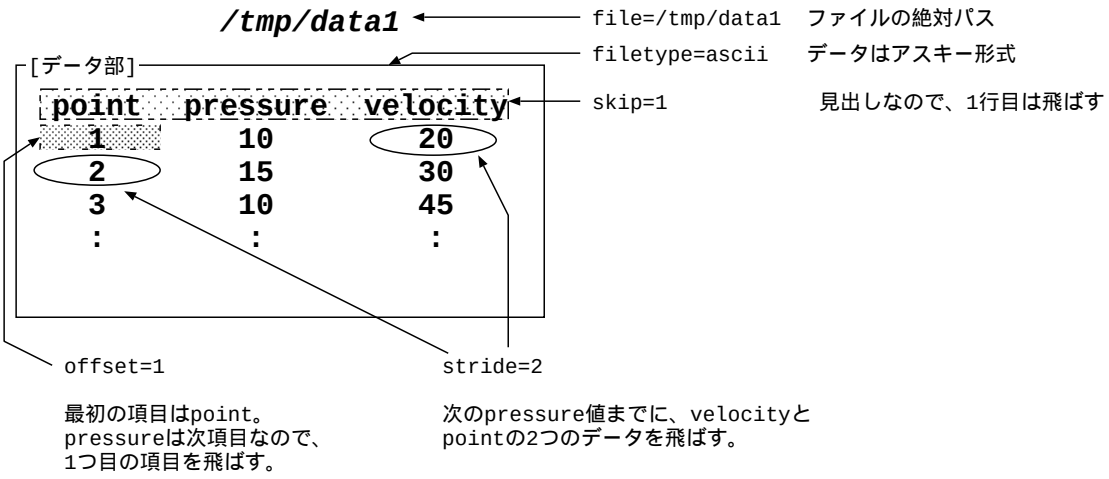

図 10: variable 項目の説明

2.5 フィールド ・データの利用例

フィールド ・データを用いた可視化の簡単な例を示します [2] . まず , 図 11,14,15 で示しているような内容を持つファイルを作成します . それぞれのファイル名はsampl.fld,data1,data2 とします . sampl.fld の内容に記述してあるように , ファイルdata1 とdata2 は /tmp/ ディレクトリに置きます .

次に , AVS を起動し図 12 に示しているようなネットワークを作成します . 作成したフィールド ・データを AVS のネットワークへ読み込む動作は , [read field] モジュールによって行ないます . ネットワーク ・エディタ上で [read field] モジュールを選択すると , 左側にフィールド ・データを選択できるようになります . [New dir] ボタンや [New file] ボタンを押して作成したsample.fld ファイルを指定しましょう . ファイルを指定すると , ネットワークにsample.fld ファイルのデータが流れ込みます . すると図 2.5 で示したような図が表示されます .

```
# AVS
ndim=3
dim1=9
dim2=2
dim3=2
nspace=3
veclen=1
data=float
field=irregular
variable 1 file=/tmp/data1 filetype=ascii skip=1 offset=1 stride=2
coord 1 file=/tmp/data2 filetype=ascii skip=1 offset=1 stride=4
coord 2 file=/tmp/data2 filetype=ascii skip=1 offset=2 stride=4
coord 3 file=/tmp/data2 filetype=ascii skip=1 offset=3 stride=4
```

図 11: sample.fld

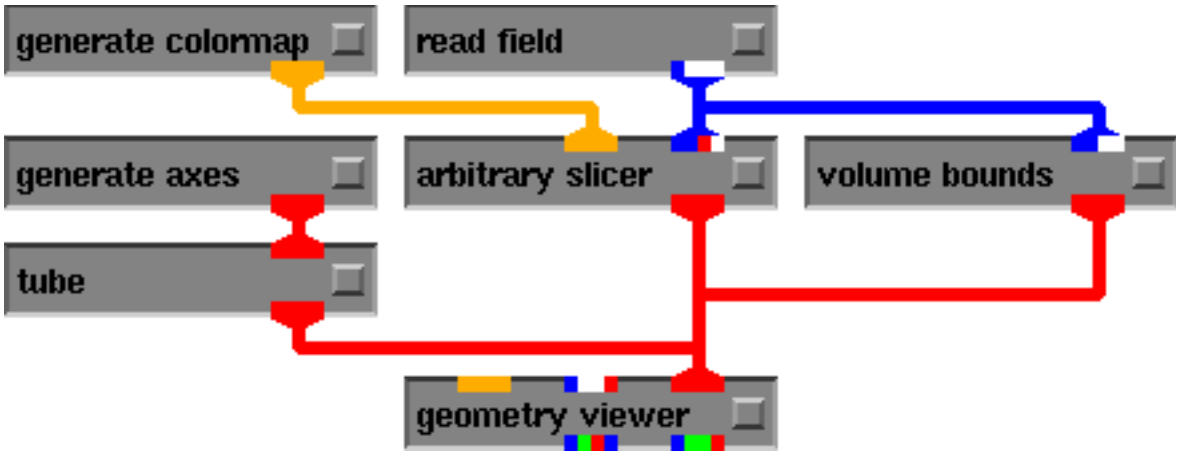

図 12: ネットワークの例

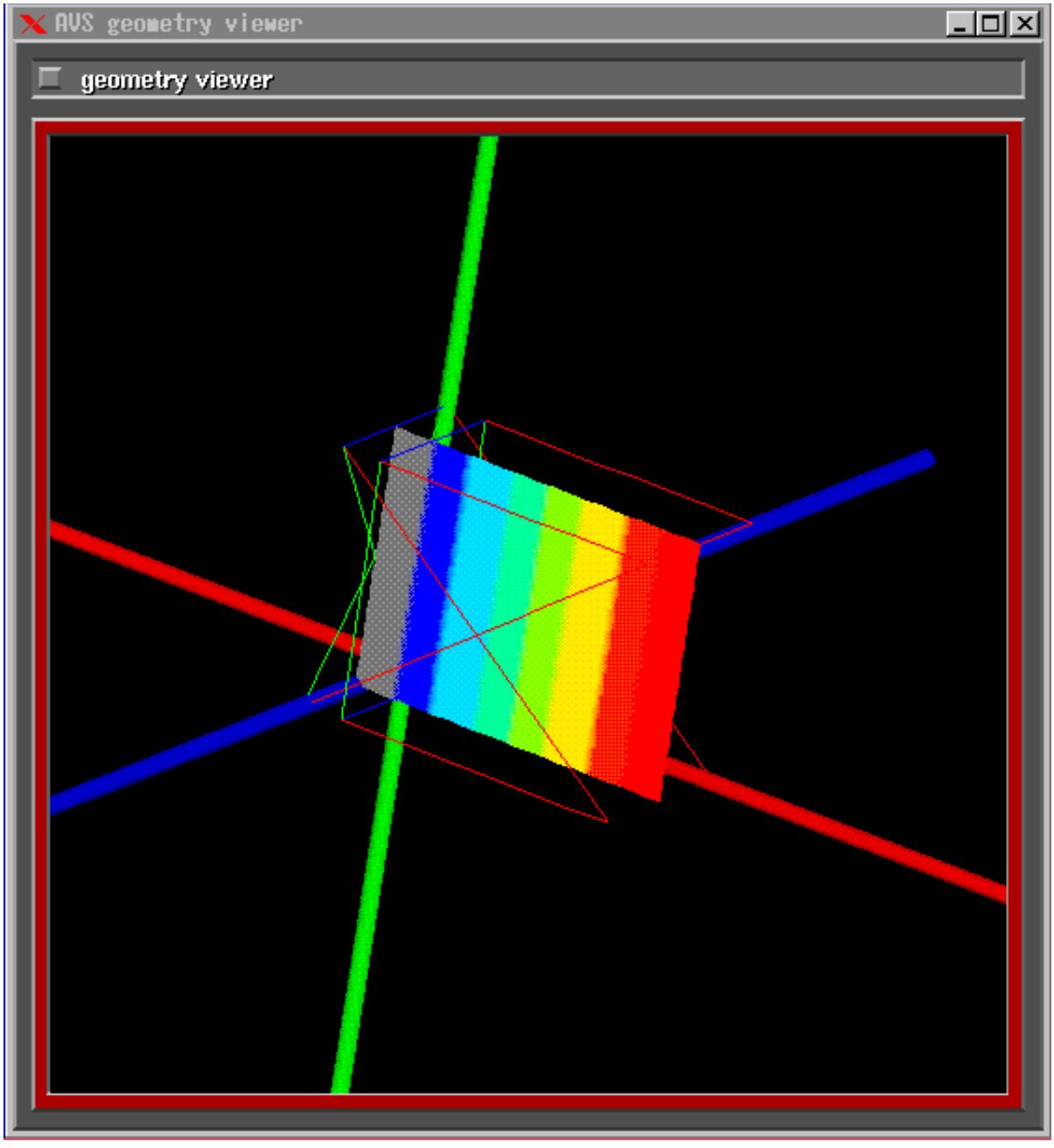

図 13: 表示例

point	data
1	0.0e+00
2	5.0e+01
3	1.0e+02
4	1.5e+02
5	2.0e+02
6	2.5e+02
7	3.0e+02
8	3.5e+02
9	4.0e+02
10	0.0e+00
11	5.0e+01
12	1.0e+02
13	1.5e+02
14	2.0e+02
15	2.5e+02
16	3.0e+02
17	3.5e+02
18	4.0e+02
19	0.0e+00
20	5.0e+01
21	1.0e+02
22	1.5e+02
23	2.0e+02
24	2.5e+02
25	3.0e+02
26	3.5e+02
27	4.0e+02
28	0.0e+00
29	5.0e+01
30	1.0e+02
31	1.5e+02
32	2.0e+02
33	2.5e+02
34	3.0e+02
35	3.5e+02
36	4.0e+02

図 14: data1

point	X	Y	Z
1	0.0	0.0	0.0
2	0.5	0.0	0.0
3	1.0	0.0	0.0
4	1.5	0.0	0.0
5	2.0	0.0	0.0
6	2.5	0.0	0.0
7	3.0	0.0	0.0
8	3.5	0.0	0.0
9	4.0	0.0	0.0
10	0.0	3.0	0.0
11	0.5	3.0	0.0
12	1.0	3.0	0.0
13	1.5	3.0	0.0
14	2.0	3.0	0.0
15	2.5	3.0	0.0
16	3.0	3.0	0.0
17	3.5	3.0	0.0
18	4.0	3.0	0.0
19	0.0	0.0	2.0
20	0.5	0.0	2.0
21	1.0	0.0	2.0
22	1.5	0.0	2.0
23	2.0	0.0	2.0
24	2.5	0.0	2.0
25	3.0	0.0	2.0
26	3.5	0.0	2.0
27	4.0	0.0	2.0
28	0.0	3.0	2.0
29	0.5	3.0	2.0
30	1.0	3.0	2.0
31	1.5	3.0	2.0
32	2.0	3.0	2.0
33	2.5	3.0	2.0
34	3.0	3.0	2.0
35	3.5	3.0	2.0
36	4.0	3.0	2.0

図 15: data2

3　イメージ ・データ (Image Data)

イメージ ・データとボリューム ・データは，前述したフィールド ・データの特別な場合です．2 次元および 3 次元のフィールド ・データは，画像と立体を表現するものとして非常によく用いられるため，この 2 つだけ特別に定義されています．

イメージ ・データを格納しているファイルの名前には，必ず “.x” で終る拡張子が付いていなければなりません．イメージ ・データのファイルは図 16 に示しているようなフォーマットを持っています．イメージ ・データではデータは 4byte 毎に区切って利用され，各 4byte のデータはバイナリ形式で表現されています．

最初の 4 バイトは画像の X 方向 (水平方向) の大きさ (ピクセル数) を表現し，次の 4 バイトは Y 方向 (垂直方向) の大きさ (ピクセル数) を表現しています．その後に各ピクセルのデータが 4byte おきに並んでいます．1 ピクセルは 4byte で表現され，透明度，赤，緑，青のデータが 1 バイト (0 ~ 255 の値) で収められています．

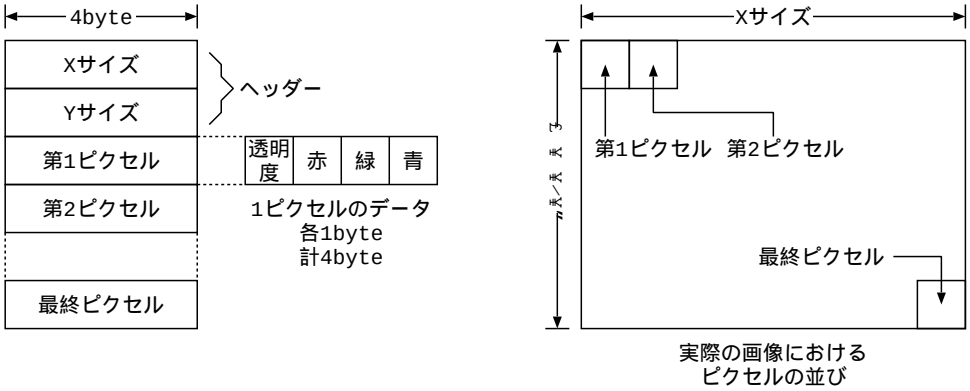

図 16: イメージ ・データ

4　ボリューム ・データ (Volume Data)

ボリューム ・データは，3 次元のフィールド ・データと同じです．3 次元立体の可視化は頻繁に行なわれるため，特別にデータ型として定義されています．ボリューム ・データは MRI などの医療画像の可視化によく用いられます．ボリューム ・データを格納しているファイルの名前には，必ず “.dat” で終る拡張しが付いていなければなりません．拡張子が異なっている場合，“read volume” モジュールからデータを読みこむ際にファイル名が表示されません．

ボリューム ・データのファイルは図 17 に示しているようなフォーマットを持っています．ボリューム ・データのファイルは，1byte バイナリ形式データが 1 塊のデータとして表現されます．最初の 3 バイトが X,Y,Z 方向の大きさを表現します．その後に各ボクセルのデータが 1byte おきに並んでいます．ボクセル (boxel) とは正規化された 3 次元のスカラーデータのことです．ボクセルという粒を集めて立体を構成していると考えて下さい．X,Y,Z 方向の大きさは 1 バイトで表現されているので，$255 \times 255 \times 255$ 個以上のボクセルを持つデータは，ボリューム ・データ形式では表現できません．これ以上のボクセルを持つデータを扱いたい場合には，フィールド ・データを使用する必要があります．

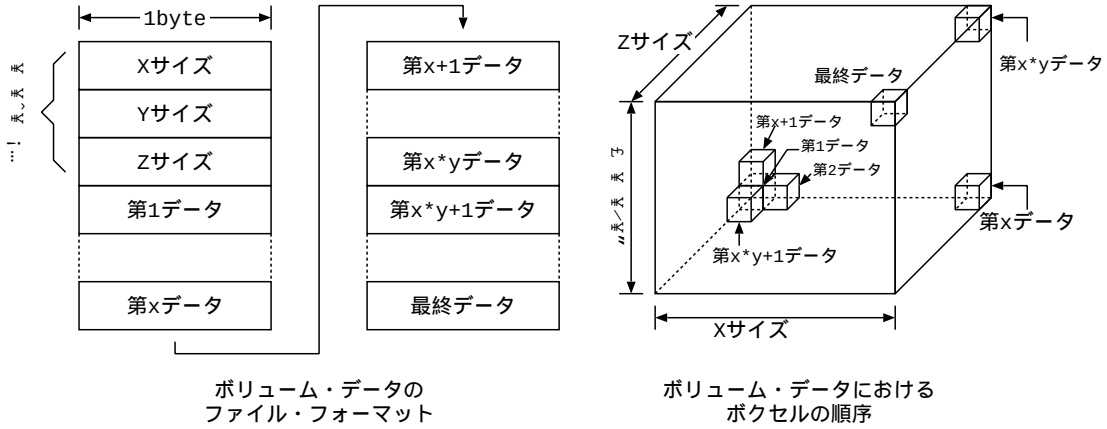

図 17: ボリューム ・データ

5 UCD データ (Unstructured Cell Data : 非構造格子データ)

UCD データは非構造格子型のデータ型を表現し , 主に有限要素法などを用いる構造解析や計算流体力学の解析結果を表現するために用いられます . UCD データを格納しているファイルの名前には , 必ず “.inp” で終る拡張子が必要です . 拡張子が異なっている場合 , [read ucd] モジュールからデータを読みこむ際にファイル名が表示されません .

5.1 UCD データのデータ構造

UCD データは図 18 に示しているような , 階層構造を持っています . 階層は上からストラクチャー (Structure 構造体) , セル (Cell 要素) , ノード (Node 節点) という順になっています . データの定義は , ストラクチャー , セル , ノードの各階層ごとに設定でき , それぞれの成分はスカラー又はベクトルで表現します .

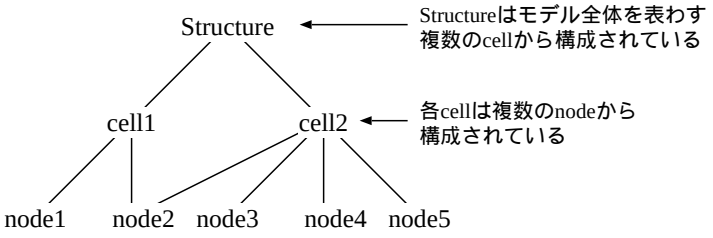

図 18: UCD のデータ構造

セルには図 19 で示しているような 8 種類の型があります . それぞれのセルを構成する節点には番号が付いています . 番号の順序は , 図 20 のように面の法線を考えて付けます .

5.2 UCD ファイル ・フォーマット

UCD データはアスキー形式でもバイナリ形式でも記述することができます . どちらも [read ucd] モジュールを用いて読み込むことが可能です . ただしアスキー形式はバイナリ形式に比べて AVS に読み込む際の速

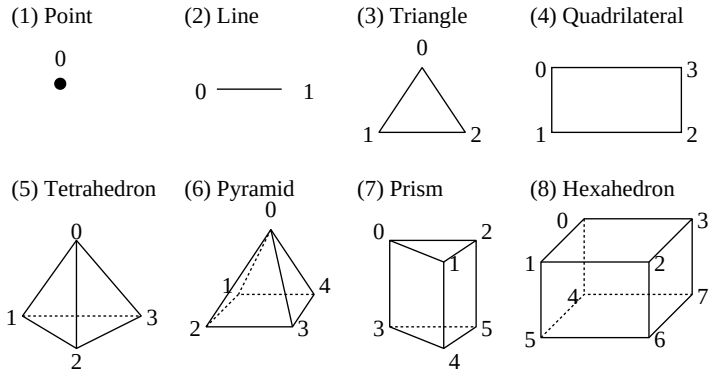

図 19: セルの種類

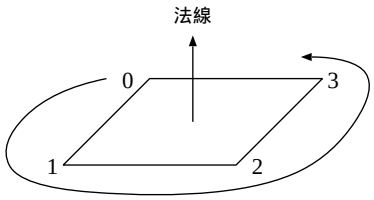

図 20: 節点番号の順序

度が遅くなります．アスキー形式で記述した UCD データは，一度 [read ucd] モジュールで読みこみ，次に [write ucd] モジュールで出力，保存するとバイナリ形式に変換されますので，同じデータ複数回使用する場合は，アスキー形式からバイナリ形式へ変換すると良いでしょう．この節では説明のしやすさの観点から，アスキー形式で記述した UCD データについて説明します．

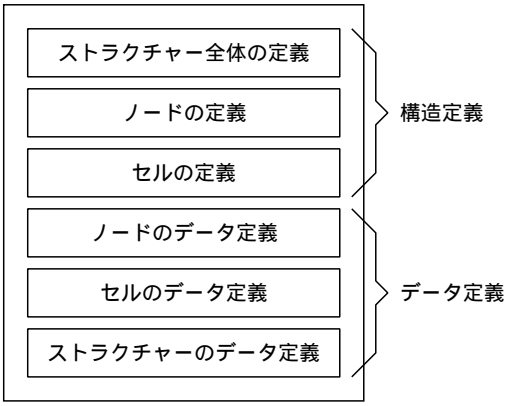

図 21: UCD ファイル ・フォーマット

UCD データを記述するファイルの内部フォーマットは，図 21 に示しているように 6 つの部分から成っています．最初の 3 つの構造定義部分は必ず記述しなければなりません．データ定義部分は必要に応じて記述します．それぞれの記述方法については図 22 を参考にして下さい．

- 図 22 におけるセルタイプは次のように省略して記述することも可能です．

Point	=	pt	Line	=	line
Triangle	=	tri	Quadrilateral	=	quad
Tetrahedron	=	et	Pyramid	=	pyr
Prism	=	prism	Hexahedron	=	hex

- 図 22 における各ノード (セル) が持つデータ成分数とベクトル長の関係は次に示す例のようになります．

```
(ノード数)(セル数)(各ノードのデータ数)(各セルのデータ数)(モデルのデータ数)

(ノードID) (X) (Y) (Z)
    :

(セルID) (材料) (セルタイプ) (セルを構成するノードの継がり)
    :

(ノードのデータ成分数) (成分1のベクトル長) (成分2のベクトル長) ・・・
(各ノード成分のラベル),(単位)
    :
(ノードID 1) (ノードデータ) (ノードデータ) (ノードデータ) ・・・・
    :

(セルのデータ成分数) (成分1のベクトル長) (成分2のベクトル長) ・・・
(各セル成分のラベル),(単位)
    :
(セルID 1) (セルデータ) (セルデータ) (セルデータ) ・・・・
    :

 (モデルのデータ成分数) (成分1のベクトル長) (成分2のベクトル長) ・・・
 (各モデル成分のラベル),(単位)
     :
 (モデルID 1) (モデルデータ) (モデルデータ) (モデルデータ) ・・・・
     :
```

図 22: UCD ファイル，アスキー ・ファイル ・フォーマット

	データ成分数	ベクトル長
温度 (T)	1	1
圧力 (P)	1	1
速度 (V_x, V_y, V_z)	1	3

- 図 22 におけるノード成分の単位は省略可能です．ただし省略する場合でもカンマ “,” は記述する必要があります．

6 おわりに

今回は可視化作業をするために AVS へ入力されるデータについて説明しました．AVS へデータを入力するには 2 つの方法があります．利用者のデータを標準フォーマット形式で作成して AVS に可視化させる方法と，利用者独自フォーマットのデータを解釈できる入力モジュールを作成する方法です．今回説明したのは AVS が標準で入力できるデータ ・フォーマットです．

標準データ ・フォーマットには，フィールド ・データ，イメージ ・データ，ボリューム ・データ，UCD データ，ジオメトリ ・データの 5 種類があります．これらのデータ ・フォーマットのうち，最初の 4 つについて説明しました．これらのいずれかの形式でデータを記述すれば，そのデータを AVS で可視化させることが可能です．

次号では，ジオメトリ ・データの説明や，標準データ ・フォーマットを使用した例を説明したいと考えています．モジュールの作成方法や，アニメーションの使用法も説明していきたいと思います．AVS にはさまざまな利用方法がありますので，是非利用してみて下さい．九州大学大型計算機センターの WWW ページからも，AVS についての説明を見ることができるようにしています．こちらも参考にしてください．インターネット上には AVS 利用者のメイリングリストもありますので，電子メールが利用できる方はメイリングリストに参加してみても良いでしょう．

http://www.cc.kyushu-u.ac.jp/system/avs.html

参考文献の中に，インターネット上にある AVS に関連する WWW ページの URL を示しておきます．インターネットと WWW が利用可能な方はこちらも参考にして下さい．

参考文献

[1] “AVS 入門 (1)”
九州大学大型計算機センター広報 Vol.30 No.3, pp.225-241, 1997.

[2] “AVS 入門教育テキスト 第 2 版”
富士通長野システムエンジニアリング, 1995.

[3] “京都大学大型計算機センター 利用の手引　– AVS 編 –”
京都大学大型計算機センター広報編集委員会, 1997.

[4] “AVS ユーザーズ ・ガイド (ダイジェスト) 第 3 版,”
クボタコンピューター株式会社, 1992.

[5] “AVS ハンドブック – For The Better Visualization Environment–”,
クボタコンピューター株式会社, 1993.

[6] “AVS ハンドブック Vol.2 – For The Better Visualization Environment –”,
クボタコンピューター株式会社, 1993.

[7] “AVS ハンドブック Vol.3 – For The Better Visualization Environment – ,”
クボタコンピューター株式会社, 1993.

[8] “AVS User’s Guide, Release 4,”
ADVANCED VISUAL SYSTEMS Inc., 1992.

[9] “AVS Developters’s Guide, Release 4,”
ADVANCED VISUAL SYSTEMS Inc., 1992.

[10] “AVS Module Reference, Release 5,”
ADVANCED VISUAL SYSTEMS Inc., 1993.

[11] “ Advanced Visual Systems 社 WWW ページ, ”
http://www.avs.com/

[12] “京都大学大型計算機センター AVS サービス WWW ページ,”
http://www.kudpc.kyoto-u.ac.jp/Services/AVS/