THE UNIVERSITY of MANCHESTER

Introductory AVS Course

MAN T&EC
The Manchester and North HPC Training and Education Centre

Course Contents

- What is AVS?
- The Network Editor
- The Geometry Viewer
- Importing Data into AVS
- Visualising Scalar Components
- Visualising Vector Components
- AVS Command Line Options and CLI
- Obtaining Hard Copy
- What is UCD?

What is AVS?

- Application Visualisation System is produced by Advanced Visual Systems.
- It is a package designed to visualise different forms of scientific data, e.g., 2D images, volumes of data and structured meshes.
- It is extensible by allowing users to write modules in C or FORTRAN and integrate them into the system.
- An advanced course will cover module writing.

THE UNIVERSITY of MANCHESTER

MAN T&EC

The Manchester and North HPC Training and Education Centre

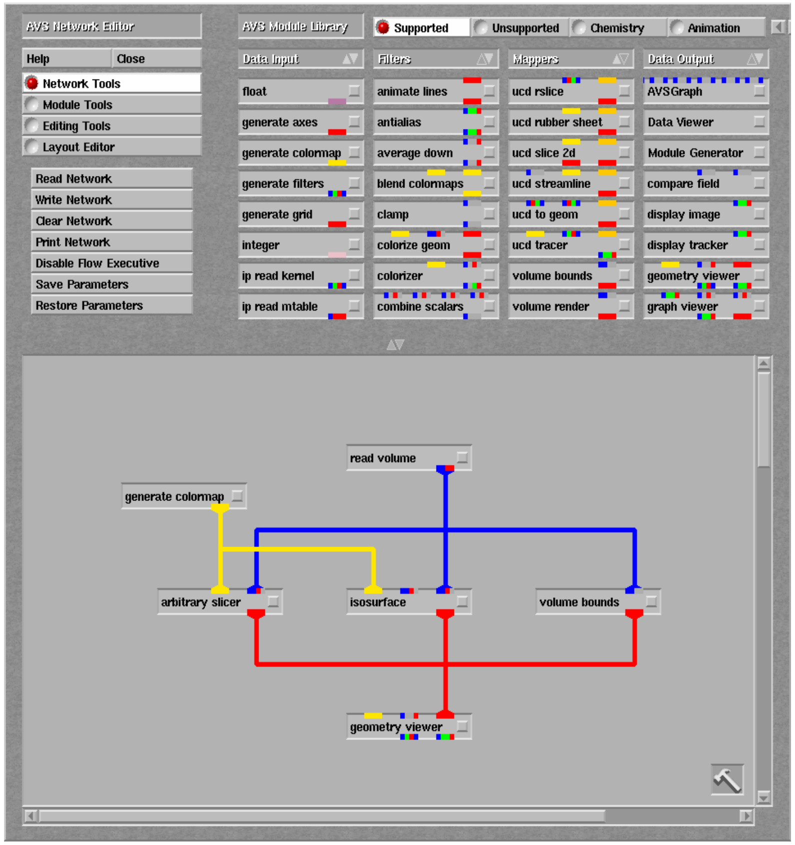
Building an application in the network editor

- A user selects modules to perform specific tasks from the palette of modules.
- These modules perform certain tasks:
  - Data input
  - Filter + processing data
  - Visualisation techniques
  - Displaying graphical results

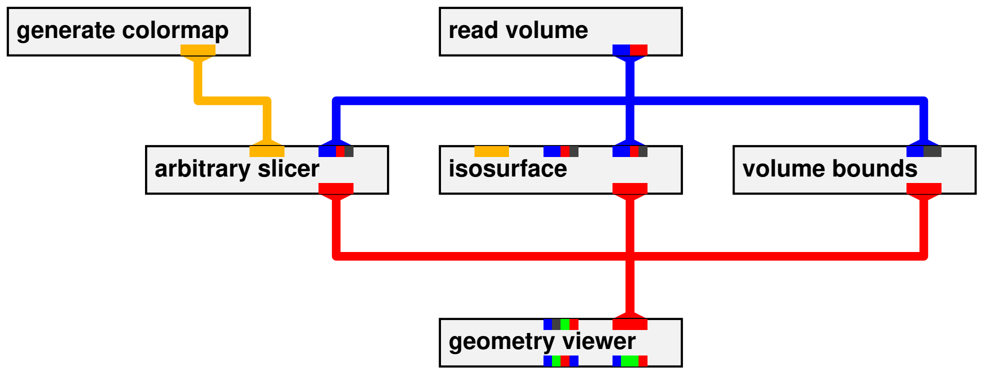
THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

The network editor



Modules and networks



THE UNIVERSITY of MANCHESTER

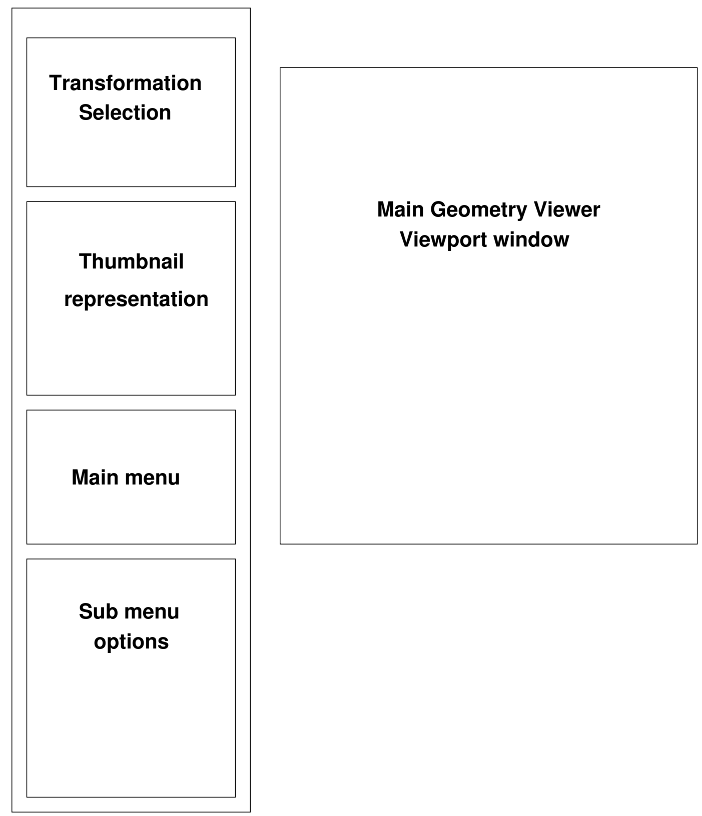
MAN T&EC
The Manchester and North HPC Training and Education Centre

Network editor exercise

What can the geometry viewer do?

- read/save objects & scenes (**`.geom`**)
- object manipulation
- change the properties of an object
- switching renderers
- altering the lights
- composite objects
- multiple view

Geometry viewer panel



Object manipulation

- left mouse button selects the object
- middle mouse button rotates the object
- right mouse button translates the object
- middle mouse button + SHIFT key scales the object
- more precise transformations can be specified by type-in widgets

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

THE UNIVERSITY of MANCHESTER

Software renderer

- All machines ranges support the software renderer
- The s/w renderer can produce many rendering capabilities e.g., transparency and texture mapping
- The software renderer allows AVS to display on an 8-bit remote display

MAN T&EC
The Manchester and North HPC Training and Education Centre

Hardware renderers

- e.g., PHIGS, GL, Dore
- the hardware renderer only provides a capability if the underlying graphics system supports it
- the hardware renderer provides the highest performance
- the user can switch between h/w and s/w renderers

Altering the lights

- the transformation panel must specify the mouse to manipulate light sources
- the “lights’’ panel must be selected
- “Show Lights” provides a vector representation of the light direction
- the light colour, type and direction can be changed

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

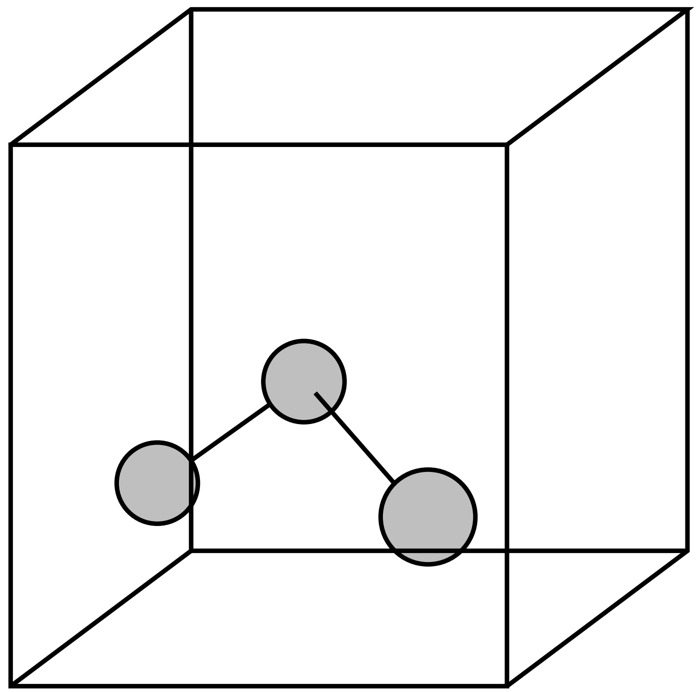
THE UNIVERSITY of MANCHESTER

Composite objects

- many scenes from AVS modules are composite objects e.g., isosurface, slice and volume bounds
- clicking the left button in an empty space selects the whole scene
- clicking on an object part selects it
- clicking in the small window cycles through the objects and scene

MAN T&EC
The Manchester and North HPC Training and Education Centre

A composite scene



Geometry viewer tutorial

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Importing data into AVS

- The overall scheme
- Describing the data
- Some examples

THE UNIVERSITY of MANCHESTER

Overall Scheme

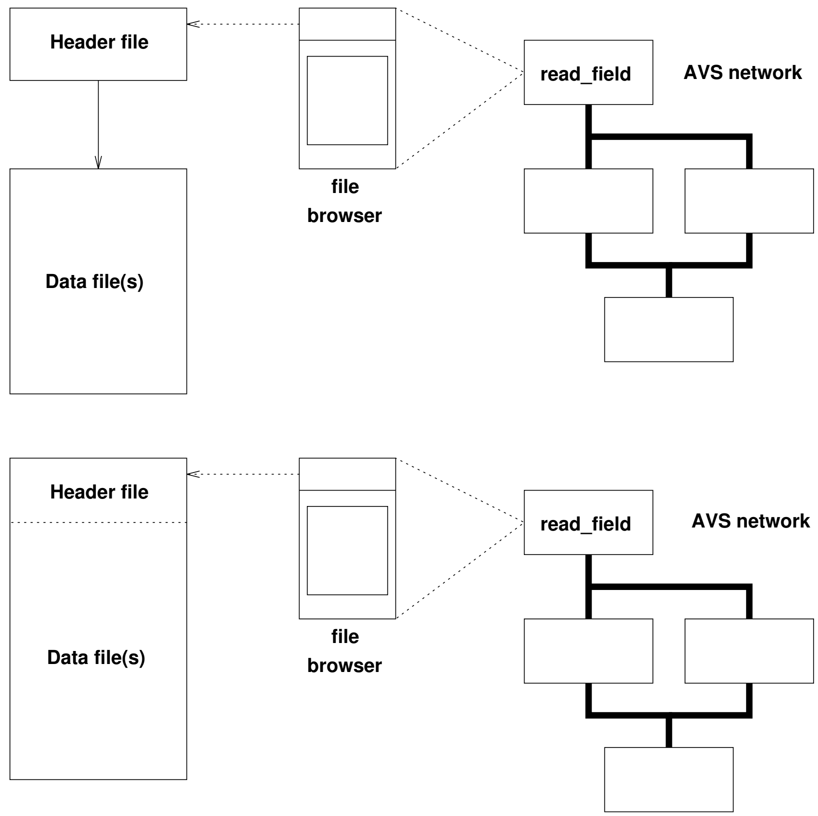
- The AVS field datatype is used to represent arrays of data in AVS.
- The data is described by an ASCII header file
- The header file can optionally contain data or reference the data file(s)
- Referencing the file(s) is a better way of working

MAN T&EC
The Manchester and North HPC Training and Education Centre

`read_field` module

- The **`read_field`** module is used to import data into an AVS network.
- The **`read_field`** module has a file browser to select the header file.

Overall Scheme



Describing the data

dimensions

- you must specify the dimensions of the data:
  - 1D: list of scattered data
  - 2D: Image
  - 3D: volume of data
- these specify the **computational** space
- you must specify the dimensions of the data as it is passed to AVS
- these specify the **coordinate** space

THE UNIVERSITY of MANCHESTER

Mapping methods

- Uniform
  - Used to import regular grids of data
- Rectilinear
  - Each data point is mapped to a physical coordinate
  - Orthogonal Axes but unequal spacing
- Irregular
  - Dimensions of computational and coordinate space may be different

MAN T&EC
The Manchester and North HPC Training and Education Centre

THE UNIVERSITY of MANCHESTER

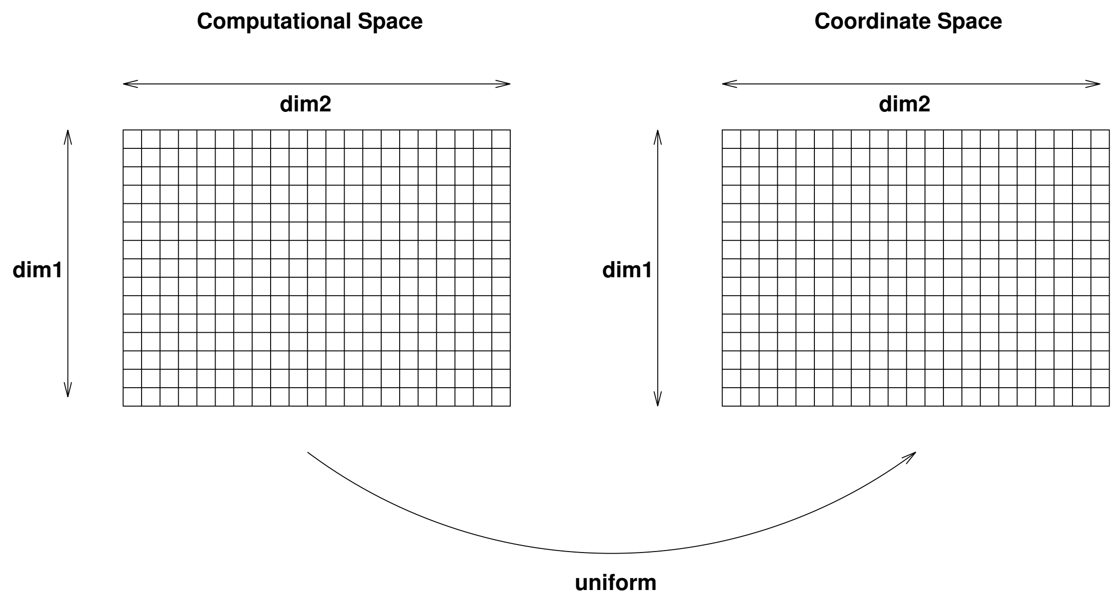
Mapping methods

UNIFORM

- The data is not transformed as it is imported (direct & implicit)
- The dimensions of the computational and coordinate space should be the same
- Used to import regular arrays of data
- No need to supply physical coordinates

MAN T&EC
The Manchester and North HPC Training and Education Centre

Uniform mapping

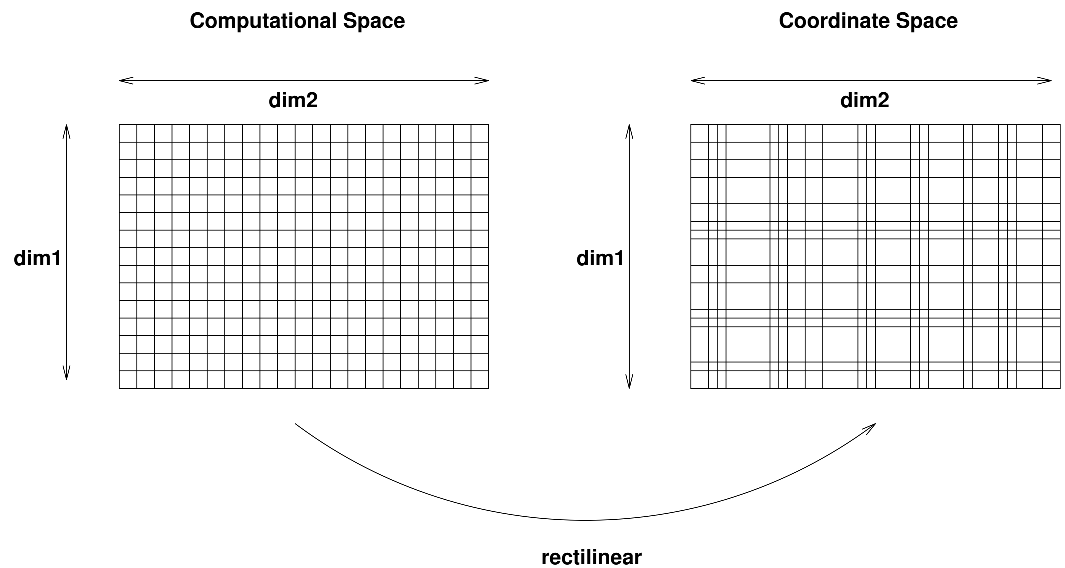


Mapping methods

RECTILINEAR

- Each dimension in the data has an explicit coordinate mapping
- The spacing along the coordinate axes can be non-uniform
- e.g., logarithmic axes
- The dimensions of the computational and coordinate space should be the same.

Rectilinear mapping

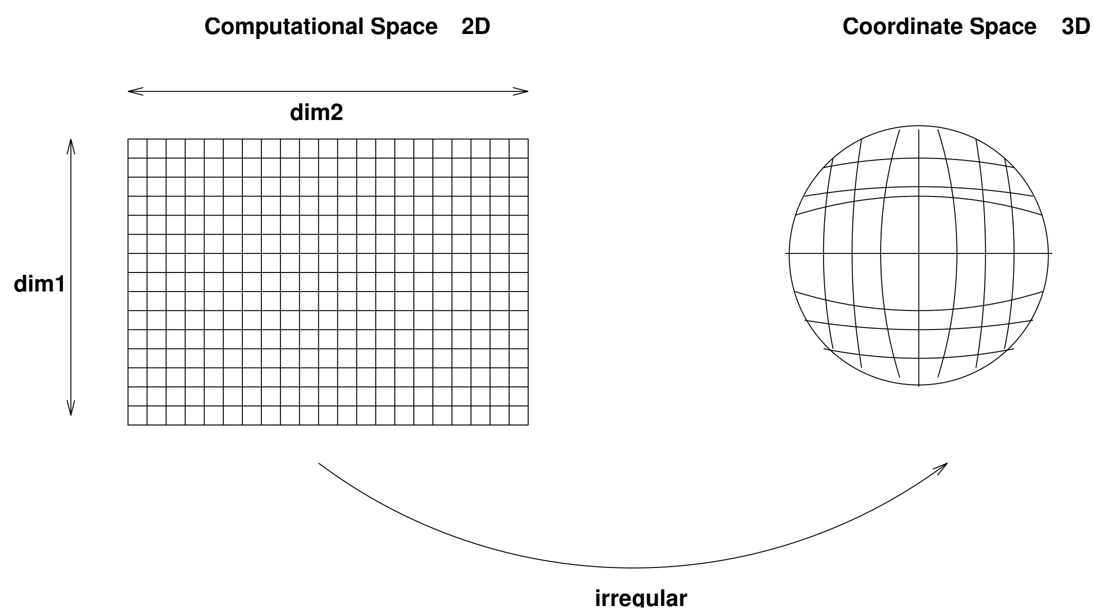


Mapping methods

IRREGULAR

- Each data element is mapped to a point in the coordinate space. Coordinates must be supplied
- The dimensions of the computational and coordinate space can be different
- e.g., 1D scattered data into 3D space
- Sometimes called Curvilinear

Irregular mapping



Describing the data elements

- The number of data components
- The datatype e.g., byte,integer,float,double
- Mapping method
- Location and layout of the data files
- A description of each data component:
  - specifying the name
  - specifying the units

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Format of the header file

- the file should have an extension of **`.fld`**
- the first line should be:

  `# AVS field file`

- the rest of the file contains a number of keywords and values

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Keywords in the header file

- **`ndim`**= number of dimensions in computational space
- **`dim1`**= specify the size of dimension 1, which can be thought of as the x dimension
- **`nspace`**= number of dimensions in coordinate space
- **`veclen`**= number of data components per element
- **`data`**= type of the data components
- **`field`**= mapping method
- **`label`**= label for each data component
- **`unit`**= units for each data component

Reading the data

- once the data structure has been defined you specify the location and format of the data file(s).
- the data file(s) can be binary or ASCII
- each data component needs a line:

```
variable N < number of format keywords >
```

- each item of coordinate information needs a line:

```
coord N < number of format keywords >
```

- Each line can reference different files

Reading the data(2)

- There are five format keywords to read the data.
  - **`file`**= location of data file
  - **`filetype`**= format of the data file, **`binary`** or **`ascii`**
  - **`skip`**: number of lines or bytes before the data is found
  - **`offset`**: number of columns to jump over before the data is read (only applicable to ASCII files)
  - **`stride:`** how many steps forward before the next item is reached
- In the case of ASCII the steps are in delimited items

Format

- all the format keywords must be on a single line
- in each of the examples a "-" indicates that the line is continued below e.g.,

```
This is a very very very very very very -
very long line.
```

- N.B., you cannot do this in the real header files though!
- the format keywords are also optional
-

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Example 1: Image data

- The dataset is a 2D image (512x256) consisting of greyscale values 0 - 255
- The data

```
0 10 0 15 200 255 ...
20 3 10 22 220 320 ...
...
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Solution for example 1

```
# AVS field file
#
ndim = 2       # Num. of computational dimensions
dim1 = 512
dim2 = 256
nspace = 2     # Num. of coordinate dimensions
veclen = 1
data = integer
field = uniform
variable 1 file=image.dat -
   filetype=ascii skip=0 stride=1
```

Example 2: Volume data

- The data consists of a 3D array of density and pressure components in a regular volume, 128^3
- The file has a line of header infomation
- The data

```
Patient: Steve 10/11/93
0 255 2 112 7 235...
5 236 6 123 8 236... ... ...
6 246 ...
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

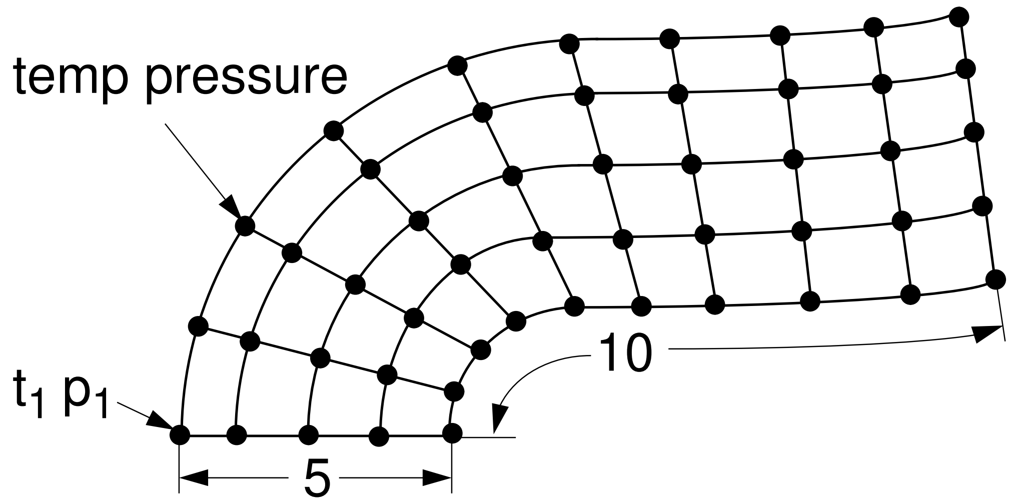
Solution for example 2

```
# AVS field file
#
ndim = 3
dim1 = 128
dim2 = 128
dim3 = 128
nspace = 3
veclen = 2
data = integer
label = denisty pressure
field = uniform
variable 1 file=volume.dat filetype= -
   ascii skip=1 offset=0 stride=1
variable 2 file=volume.dat filetype= -
   ascii skip=1 offset=1 stride=1
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Example 3: Irregular data



flow.dat

t_1 p_1 t_2 p_2 t_3 p_3 ...
...

coord.dat

x_1 y_1 x_2 y_2 x_3 y_3 ...
...

Solution to Example 3

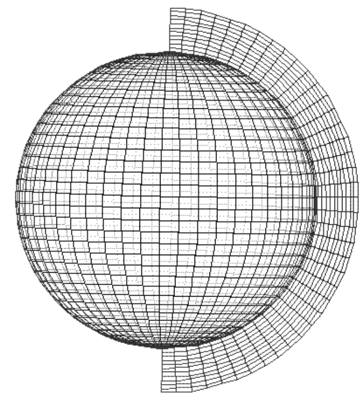
```
# AVS field file
#
ndim = 2
dim1 = 5
dim2 = 10
nspace = 2
veclen = 2
data = float
field = irregular
label = temp press
variable 1 file=flow.dat filetype=ascii -
    skip=0 offset=0 stride=2
variable 2 file=flow.dat filetype=ascii -
    skip=0 offset=1 stride=2
coord 1 file=coord.dat filetype=ascii -
    skip=0 offset=0 stride=2
coord 2 file=coord.dat filetype=ascii -
    skip=0 offset=1 stride=2
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Example 4: Climate Data

- The data is sampled on a grid of 49x40x12 around the globe
- Each grid point has:
  - Temperature (degrees kelvin)
  - Wind Velocity (M^2/sec)
  - Specific Humidity (g/kg)



- The data is contained in a number of binary files
- A separate ASCII file contains the coordinates in a spherical coordinate system

- The temperature is contained in a file called **`temp.dat.`** It consists of single floating point value at each grid point, and has 16 bytes of header information.
- Wind Velocity is in 3 files for the **`u,v`** and **`w`** components. Each file has 16 bytes of header information
- Specific humidity is in a file called **`humid.dat`**.It consists of single floating point value at each grid point, and has 16 bytes of header information.
- The coordinates are arranged as follows

```
x1 x2 x3 x4 x5 x6 x7 x8
x9 x10 .......
.
.
y1 y2 y3 y4 y5 y6 y7 y8
y9.....
.
.
z1 z2 z3 z4 z5 z6 z7 z8
z9.....
.
```

Solution to Example 4

```
# AVS field file
#
ndim = 3
dim1 = 49
dim2 = 40
dim3 = 12
nspace = 3
veclen = 5
data = float
field = irregular
label = temp humid u-comp v-comp w-comp
variable 1 file=temp.dat filetype=binary -
    skip=16 offset=0 stride=1
variable 2 file=humid.dat filetype=binary -
    skip=16 offset=0 stride=1
variable 3 file=u-velocity.dat filetype=binary -
    skip=16 offset=0 stride=1
variable 4 file=v-velocity.dat filetype=binary -
    skip=16 offset=0 stride=1
```

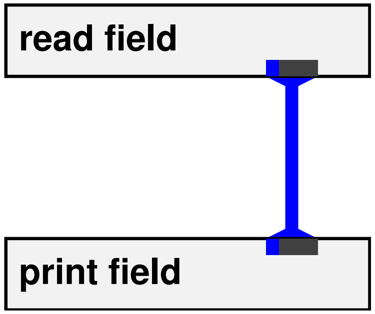
THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

```
variable 5 file=w-velocity.dat filetype=binary -
    skip=16 offset=0 stride=1

coord 1 file=ccm1.sph filetype=ascii -
    skip=1 offset=0 stride=1
coord 2 file=ccm1.sph filetype=ascii -
    skip=2941 offset=0 stride=1
coord 3file=ccm1.sph filetype=ascii -
    skip=5881 offset=0 stride=1
```

Debugging

- A useful module to include in your networks is **`print_field`**



- The module will display information on the header and data elements

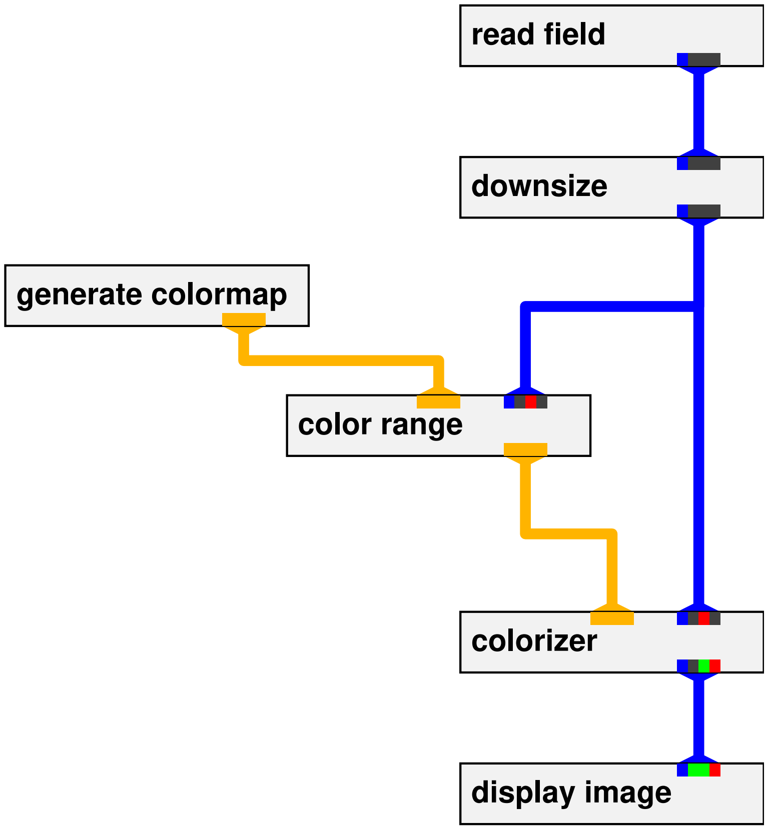
The MRI and temperature profile data

- a binary + ascii dataset
- regular arrays of data
- the MRI data is a series of bytes (binary)
- the temperature profile is a series of float point values (ascii)
- the exercise is to examine each dataset

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Importing the MRI data



Importing the MRI data

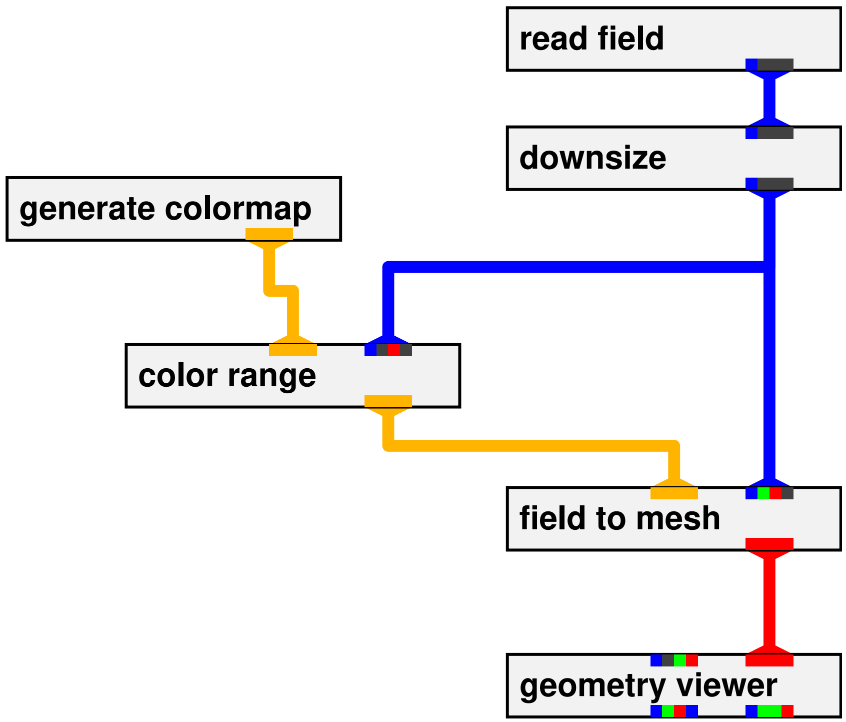
- **`generate_colourmap`**: provides a colourmap definition for other modules
- **`downsize`**: samples the data to decrease its overall size
- **`color_range`**: maps the colourmap to the extent of the data it receives
- **`colorizer:`** produces an image using the data values as an index to the colourmap

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Example of `colorizer`

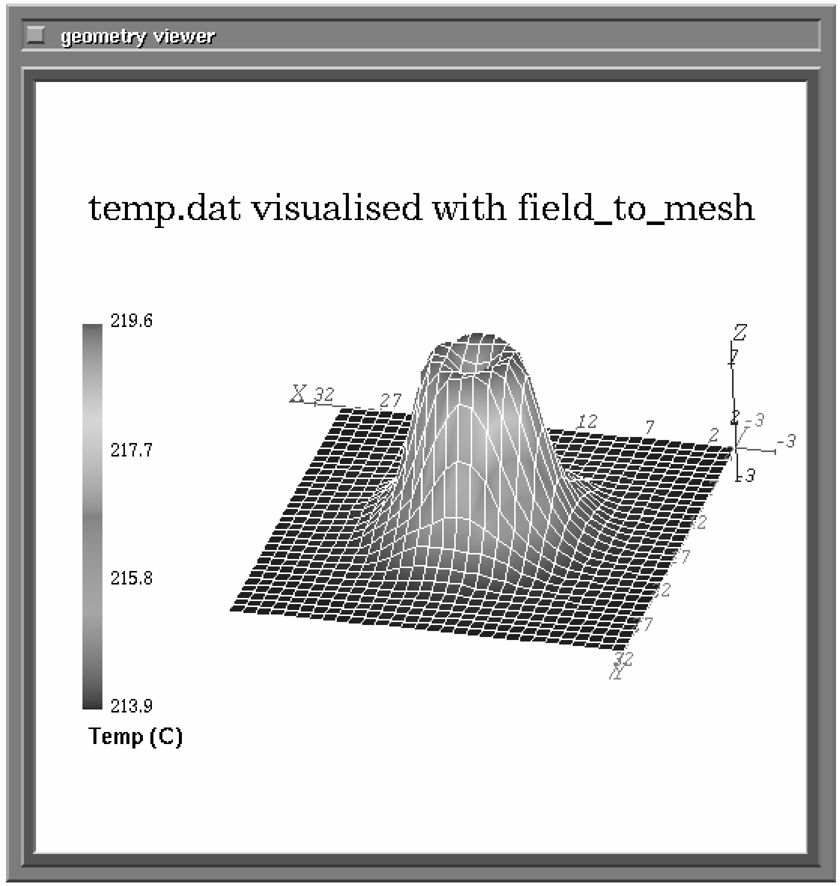
Importing the temperature profile



Importing the temperature profile

- generate_colourmap
- downsize
- color_range
- **`field_to_mesh`**: produces a 3D surface from a 2D array where the heights are produced from the data elements
- **`geometry_viewer`**: renders the output from **`field_to_mesh`**

Example of `field_to_mesh`



2D field exercise

Visualising 3D arrays

- importing a file of data showing the flow of air over a fin
- there are 5 data components and associated 3D coordinate information
- 3 of the scalar components contribute to form a vector

```
density x-mom y-mom z-mom stag x y z
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

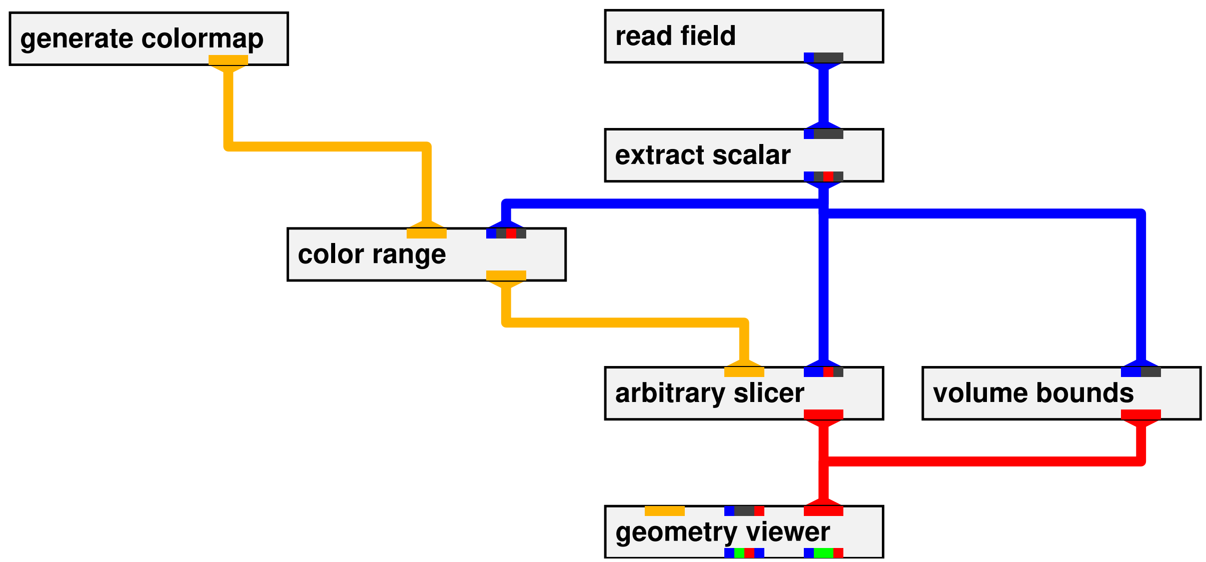
Field descriptor

```
ndim=
dim1=
dim2=
dim3=
nspace=
veclen=
data=
field=
variable 1 file=
variable 2 file=
variable 3 file=
variable 4 file=
variable 5 file=
coord 1 file=
coord 2 file=
coord 3 file=
```

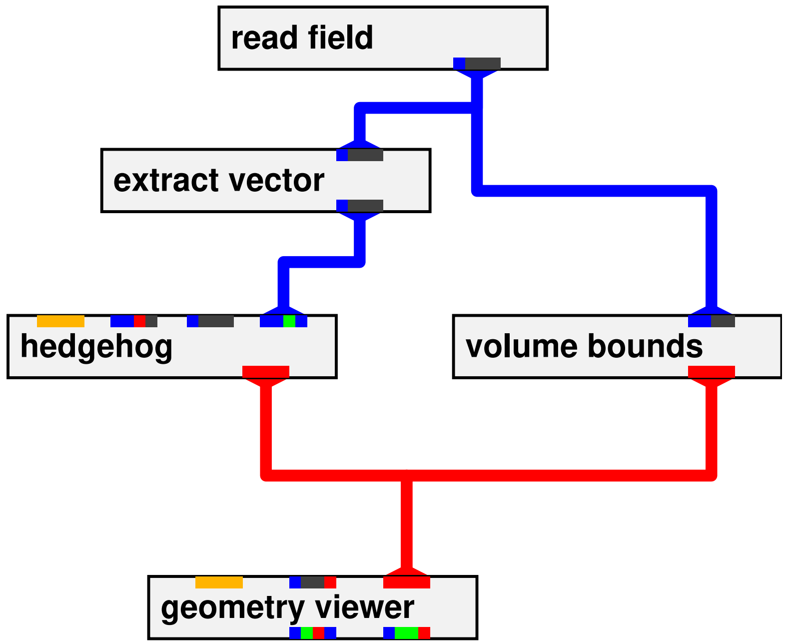
THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Visualising scalar components



Visualising vector components



`volume_bounds` module

- constructs a boundary around a 3D array of data
- it is useful to indicate your location in the array of data
- it allows you to switch on/off planes orthogonal to the axes

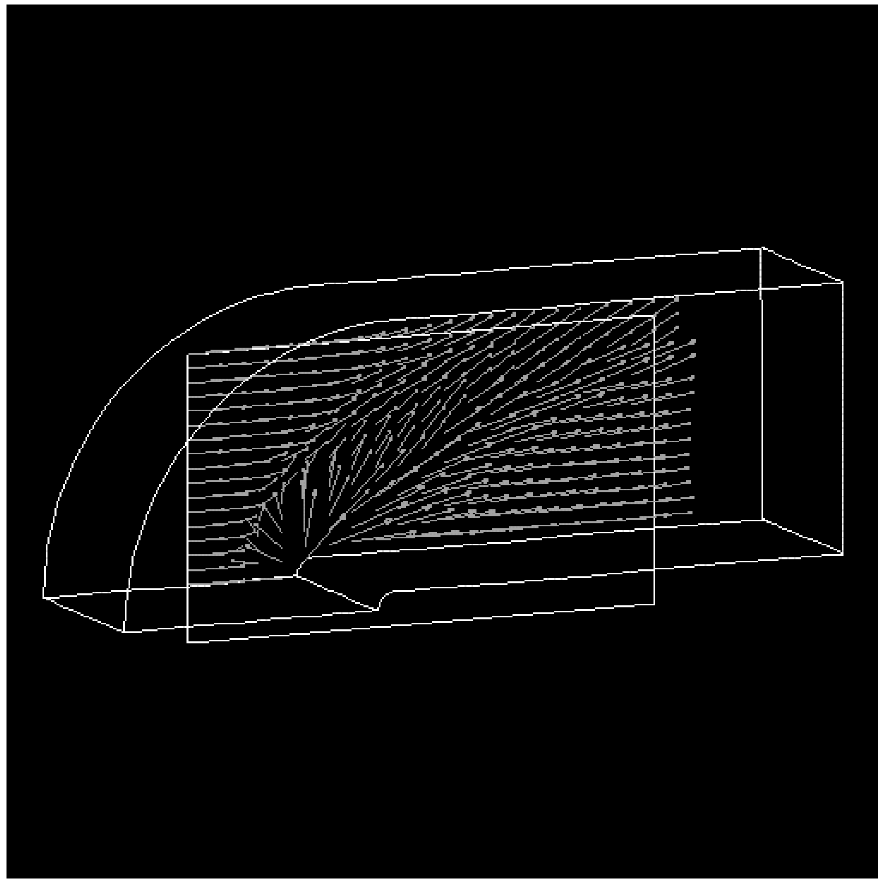
`hedgehog` module

- displays velocity data by drawing vectors which indicate the direction and magnitude of the velocity
- you can add arrow heads to the vectors
- the scale and number of samples can be altered
- various probes can be selected e.g., point, circle, line, plane, space

Manipulating the probe

- the probe is manipulated by first selecting it in the geometry viewer window
- transformations specified by the mouse allow the user to explore the data by moving the probe

Example of `hedgehog`



3D field exercise

AVS command line options

- `-version` which version are you running?
- `-network file` start AVS to use the network specified
- `-timer` performance measure on geometry viewer
- `-nohw` use the software renderer (essential if running remotely via X)
- `-usage` what are the other options?
- `-mod_time` unsupported feature for timing a network

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

AVS Command Line Interpreter (CLI)

- to access use **`avs -cli`**
- to record a session use **`avs> script -open myscript`**
- to stop recording use **`avs> script -close`**
- to playback use **`avs> script -p myscript`**

A sample CLI script

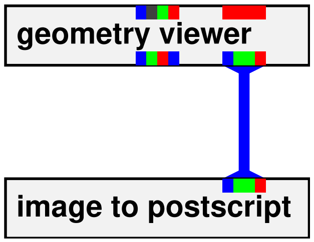
```
net_clear
module “read image.user.0” -xy 230,73 -ex $Path/
avs_library/mongo
module contrast.user.1 -xy 230,129 -ex $Path/
avs_library/mongo
module “display image.user.2” -xy 230,185
port_connect “read image.user.0”:0 contrast.user.1:0
port_connect contrast.user.1:0 “display
image.user.2”:0
parm_set “read image.user.0”:”Read Image Browser”
usr/avs/data/image/avs.x
parm_set contrast.user.1:cont_in_max 178.171722412
parm_set contrast.user.1:cont_out_min 78.480216980


script -close
```

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

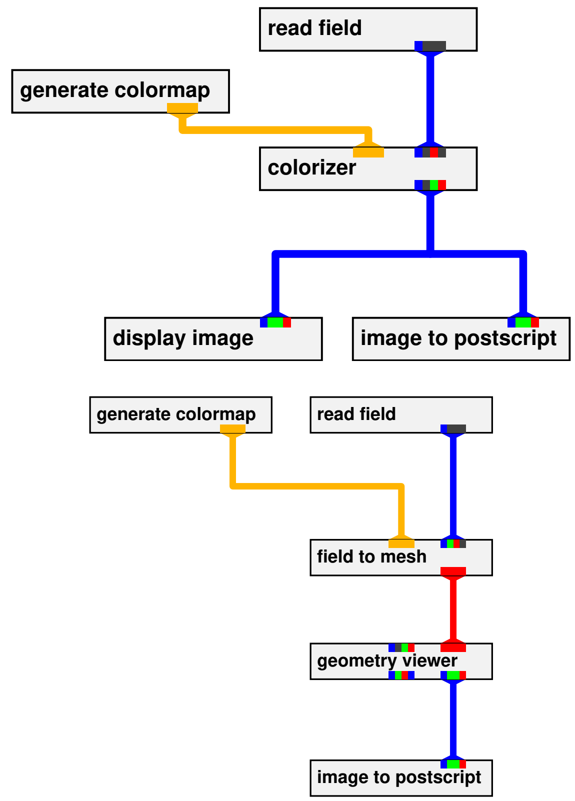
Obtaining hardcopy from AVS

- connect the module **`image_to_postscript`** to the **`geometry_viewer`** module



- you can specify landscape or portrait
- you can alter the size in inches

Practical uses



THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Customising AVS

- when AVS starts up it looks for the file **`.avsrc`** in your home directory
- if this file does not exist it uses the file

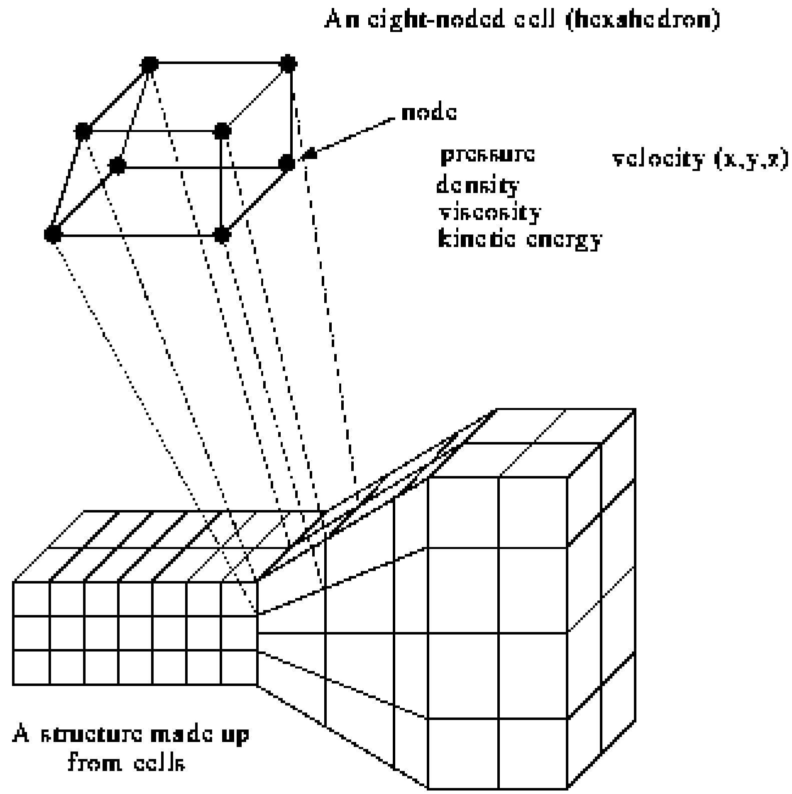
  `/usr/avs/runtime/avsrc`

- this file sets certain default attributes in AVS

The .avsrc file

```
Path /usr/local/avs
NetworkDirectory /usr/avs/networks
DataDirectory /usr/avs/data
NoHW 1
Bounding Box off
```

What is unstructured cell data?



How does AVS provide support?

- Unstructured Cell Data (UCD)
- NODES: contain data and position
- CELLS: have a geometric shape and reference nodes
- STRUCTURE: a collection of cells

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

THE UNIVERSITY of MANCHESTER

A closer look at UCD

- data at each node can be a collection of scalar and vector quantities
- the data can have labels and units
- each cell has a shape, material and cell data associated with it

MAN T&EC
The Manchester and North HPC Training and Education Centre

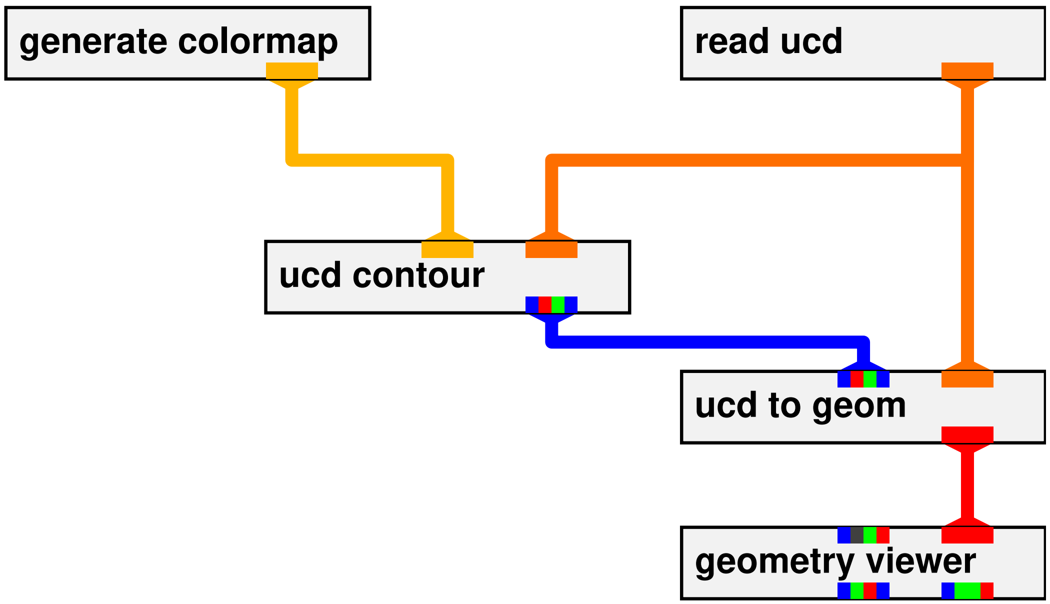
Importing UCD data

- you can write a reader using the **`UCD`** routines to parse your data file and create the appropriate UCD structure
- this is the preferred method but is not covered in this course
- AVS has a module **`read_ucd`** which imports ASCII files (**`.inp`**)
- you can change your data to match this format

An example of a `.inp` file

```
8 1 1 0 0
1 0.0 0.0 1.0
2 0.0 1.0 0.0
...
8 0.0 1.0 0.0
1 1 hex 1 2 3 4 5 6 7 8
1 1
stress, lb/in**2
1 499.999
2 567.888
...
8 5000.00
```

A sample UCD network



Other modules

- cropping and slicing (**`ucd_crop`** & **`ucd_rslice`**)
- vector data (**`ucd_hedgehog`** & **`ucd_streamlines`**)
- debugging (**`ucd_print`**)
- isosurfaces (**`ucd_iso`**)

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

THE UNIVERSITY of MANCHESTER

MAN T&EC
The Manchester and North HPC Training and Education Centre

Closing Comments

What we haven't covered!

- ❑ Remote module execution
- ❑ Macro modules and layout editor
- ❑ Writing modules in C and FORTRAN
- ❑ Animation in AVS