5 Visualizing UCD data

5.1 What is UCD Data?

UCD stands for Unstructured Cell Data. The format of UCD is just a different way for you to read your datafiles and visualize the information. UCD consists of:

- **Nodes**: contain data, which can be a collection of scalar and vector data, and position information.
- **Cells**: have geometric shape and are made up of reference nodes.
- **Structure**: collection of cells.

The UCD format is shown below:

```
<num of nodes> <num of cells> <data per node> <data per cell> <model data>
<nodeid> coordinates
<cellid> <cell shape> nodes
<data label> <data units>
<nodeid> data
```

UCD files have the suffix ".inp". An example of this is the `hex.inp` file which defines a single hexahedral cell with stress at each node. An extract from the file is given below:

```
8 1 1 0 0
   1  0.000   0.000   1.000
   2  1.000   0.000   1.000
   3  1.000   1.000   1.000
   4  0.000   1.000   1.000
   5  0.000   0.000   0.000
   6  1.000   0.000   0.000
   7  1.000   1.000   0.000
   8  0.000   1.000   0.000
   1  1 hex 1 2 3 4 5 6 7 8
   1 1
   stress, lb/in**2
   1  4999.9999
   2  18749.9999
   3  37500.0000
   4  56250.0000
   5  74999.9999
   6  93750.0001
   7  107500.0003
   8  5000.0001
```

5.2 Aims of the chapter

The writing of UCD format files is not covered by this course, but it is useful to know a little about their structure.

UCD is read into AVS/Express via the `Read_UCD` module. In order to help you to understand the process, there are several prepared UCD format files which can be found in the directory,

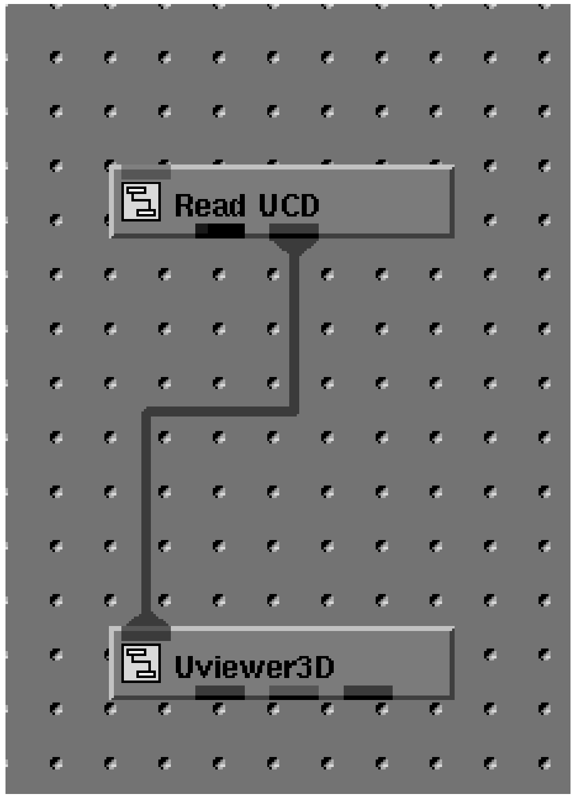
```
/usr/express/data/ucd...
```

The exercises in this chapter are designed to take you through several examples of UCD visualization and demonstrate the principal uses of the format.

5.3 Exercise 1

The aim of this exercise is to use the UCD file containing the AVS logo to demonstrate some simple networks which will display UCD data.

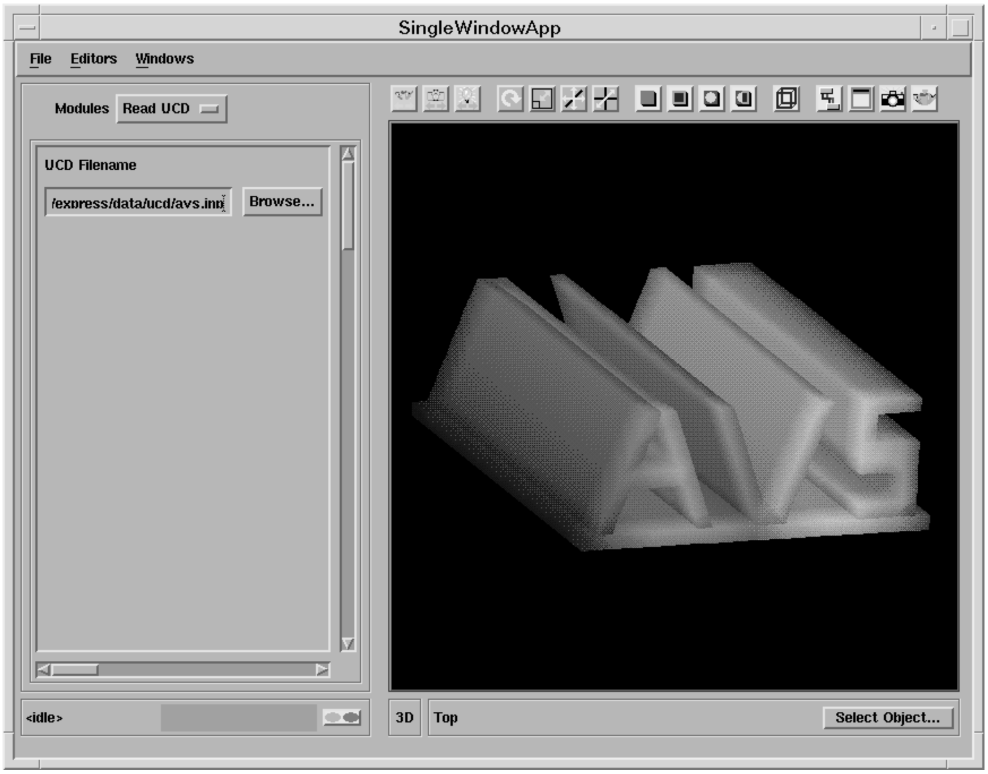
5.3.1 Displaying the UCD geometry (1)



Create the network above using the following modules:

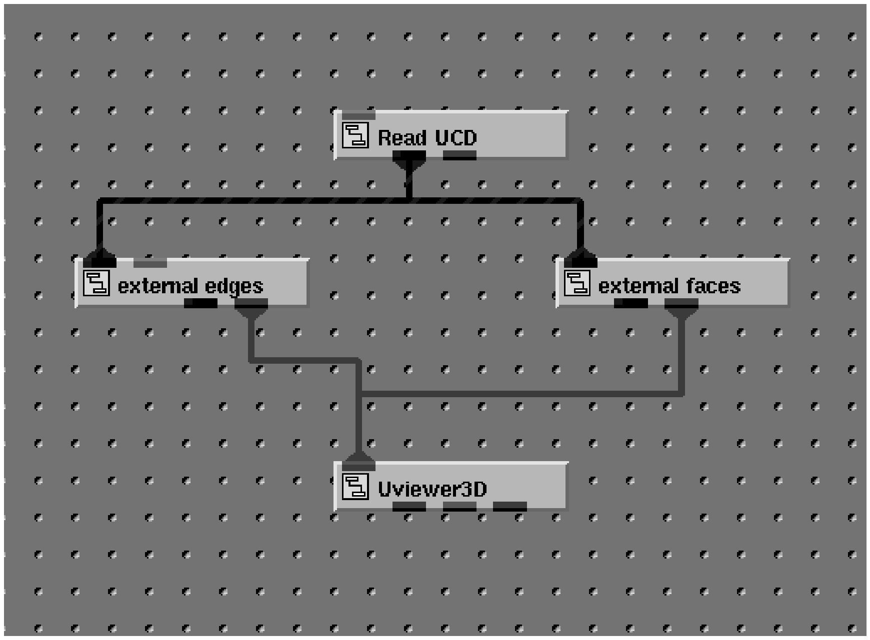
- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.3.2 Displaying the UCD geometry (2)



- Import a UCD file by selecting the file `avs.inp` using the `Read_UCD` file browser.
- Using the mouse transform and rotate the object until the picture is similar to that shown above.

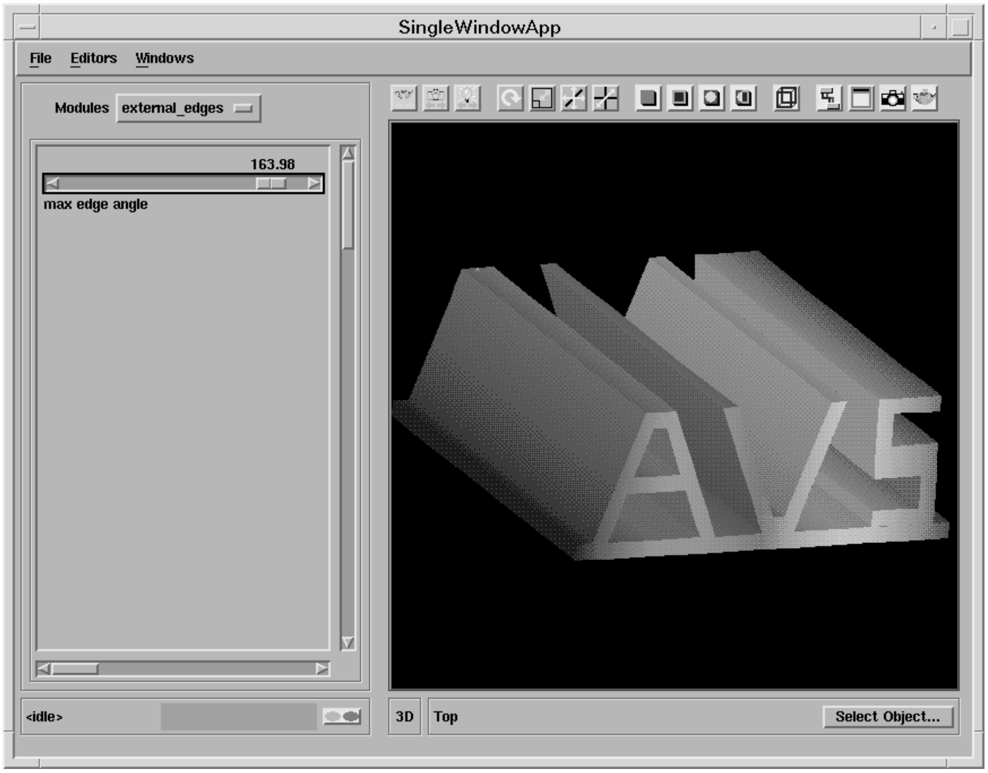
5.3.3 Displaying the UCD external edges and faces (1)



Create the network above using the following modules:

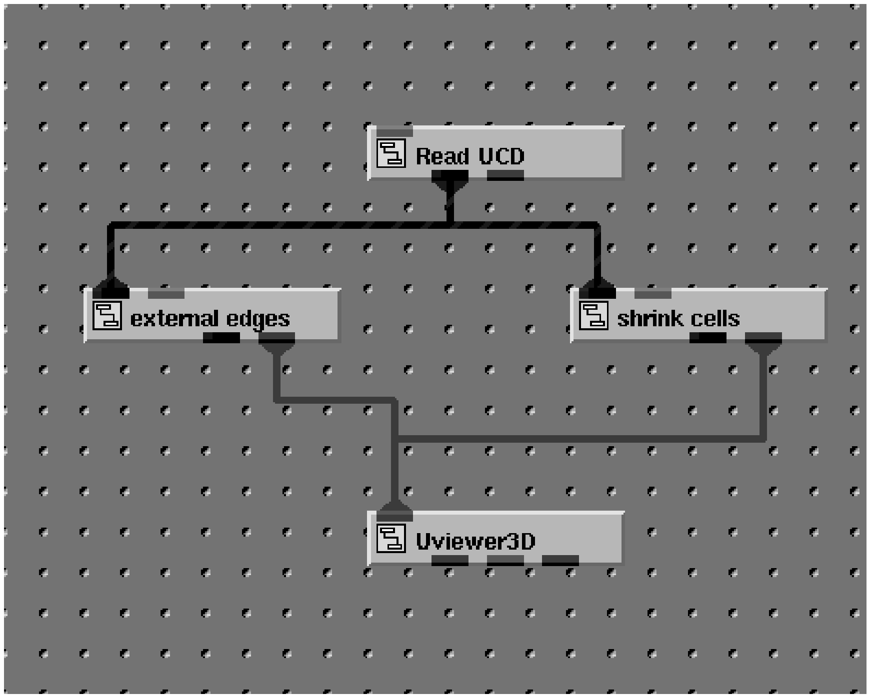
- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces a wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- `external_faces` (Mappers): produces a mesh which represents the exterior, visible faces of an unstructured mesh.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.3.4 Displaying the UCD external edges and faces (2)



- Select the `avs.inp` UCD file using the file browser on the `Read_UCD` module.
- Select the `external_edges` module's control panel.
- Alter the max line angle and observe the effect.

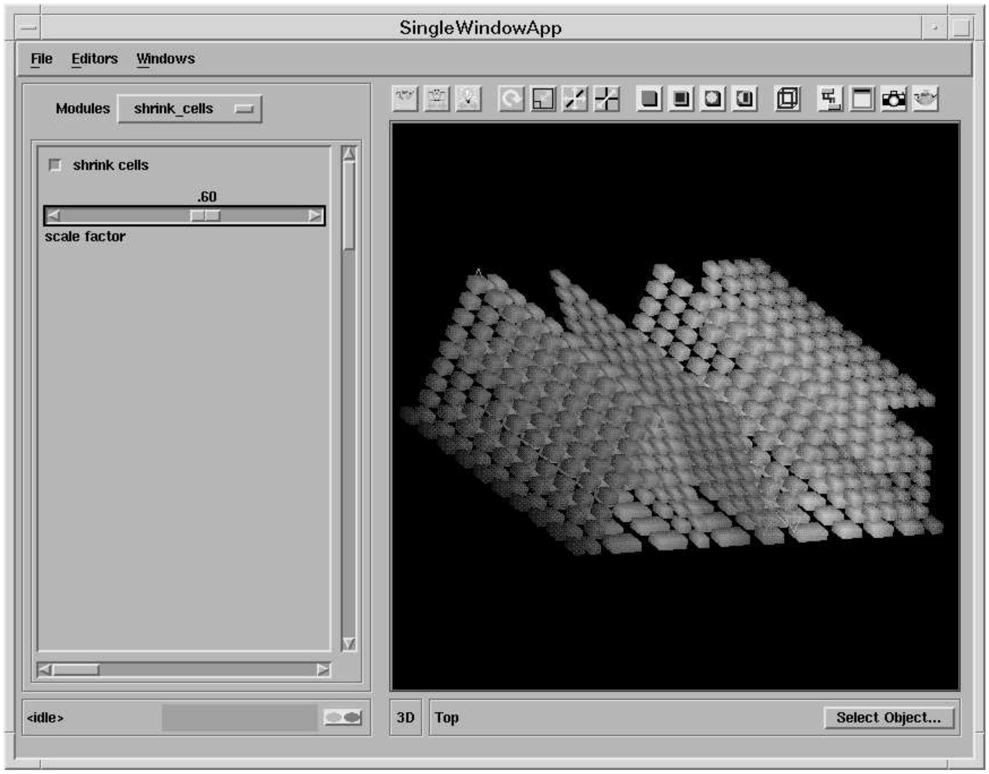
5.3.5 Shrinking the UCD cells (1)



Create the network above using the following modules:

- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces an wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- `shrink_cells` (Mappers): produces a mesh with cells shrunk relative to their geometric centres.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.3.6 Shrinking the UCD cells (2)



- Select the `avs.inp` UCD file using the file browser on the `Read_UCD` module.
- Alter the shrink factor and observe the effect.

5.4 Further exercises

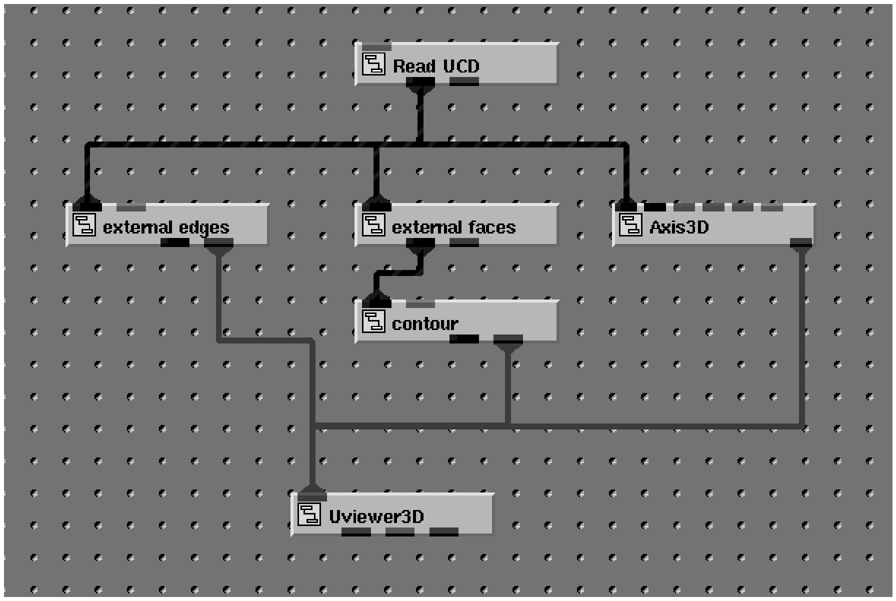
Now that you have worked though some basic UCD examples, you may wish to try out your networks on some other UCD files. Below is a list of several UCD files you may like to look at, they are to be found in the same directory as the AVS logo file.

- `pyr.inp`: example of a pyramid shaped cell
- `tet.inp`: example of a tetrahedral shaped cell
- `bluntfin.inp`: flow of air over a fin

5.5 Exercise 2

In this exercise you will look at several different networks which display the bluntfin data as a UCD.

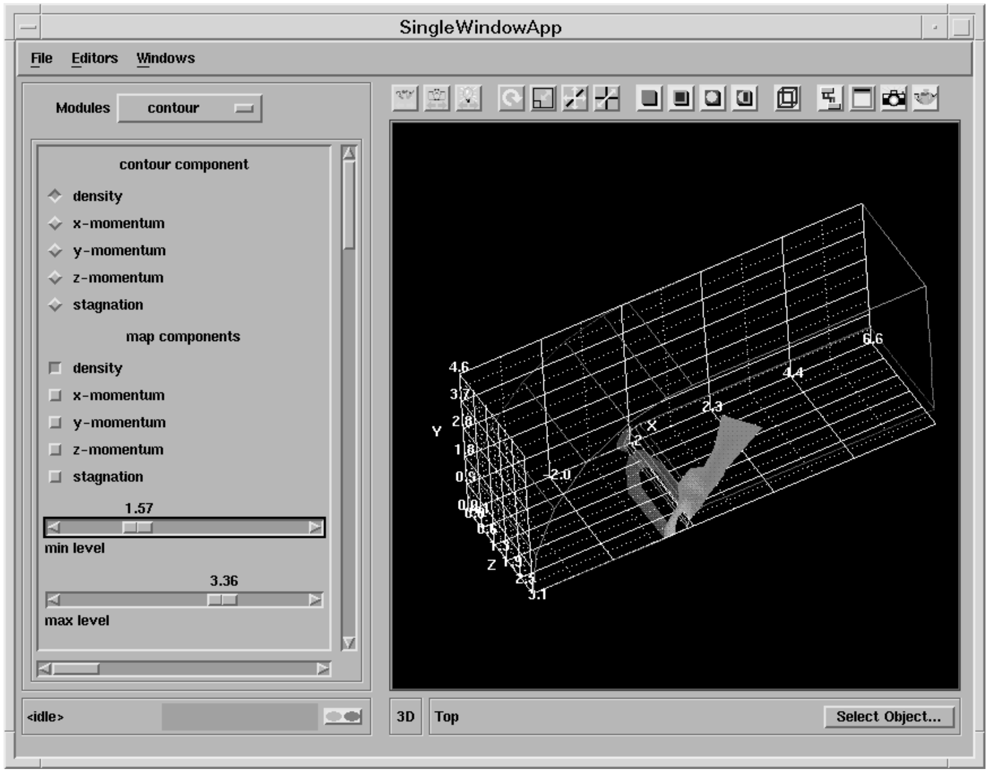
5.5.1 Displaying scalar data using `contour` module (1)



Create the network above using the following modules:

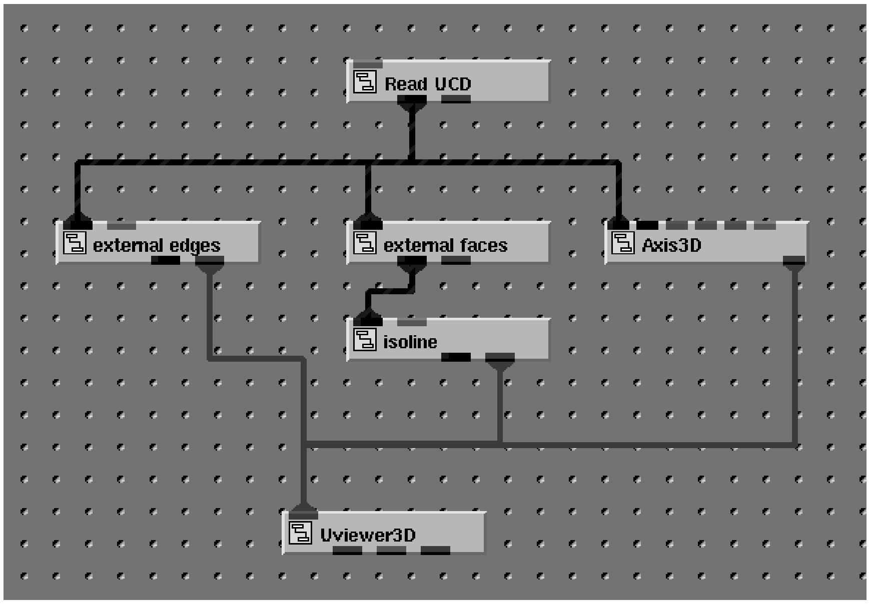
- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces an wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- `external_faces` (Mappers): produces a mesh which represents the exterior, visible faces of an unstructured mesh.
- `Axes3D` (Geometries): creates an XYZ axis grid with labels based on mesh extents.
- `contour` (Mappers): creates an isovolume bounded by two isosurfaces (3D) or isolines (2D).
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.5.2 Displaying scalar data using `contour` module (2)



- Select the file `bluntfin.inp` using the `Read_UCD` module's file browser.
- Using the mouse, rotate the object so that it appears as above.
- Select the `contour` module's control panel
- Make active the data component density
- Try altering the max and min values to be contoured and observe the effect.

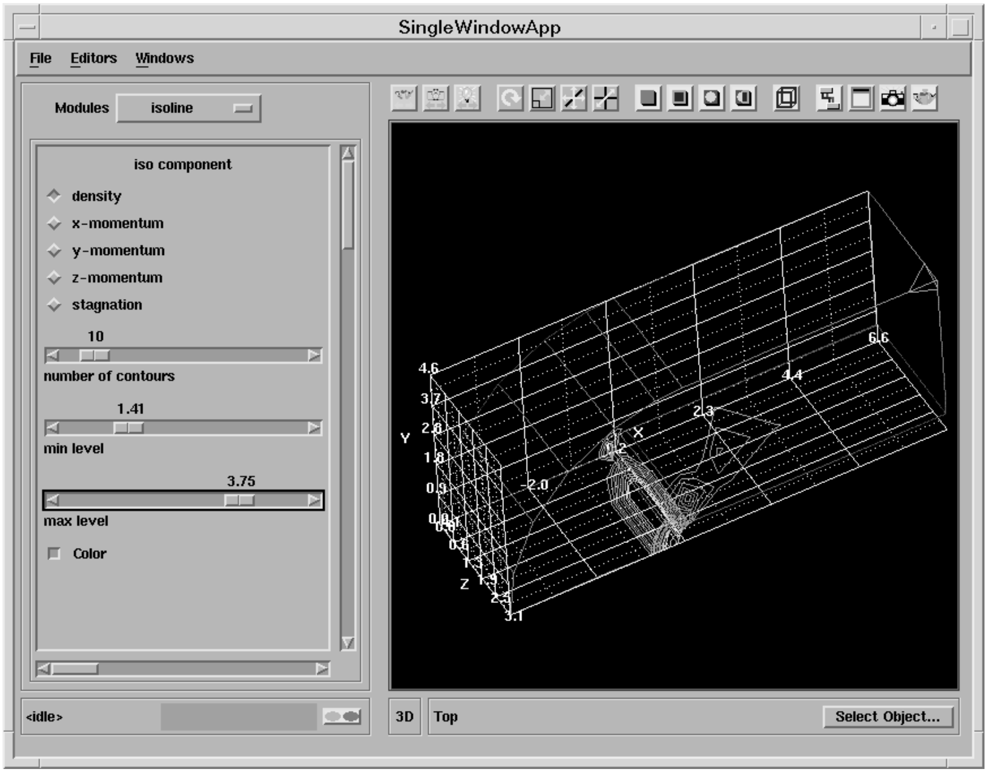
5.5.3 Using the `isolines` module (1)



Create the network above using the following modules:

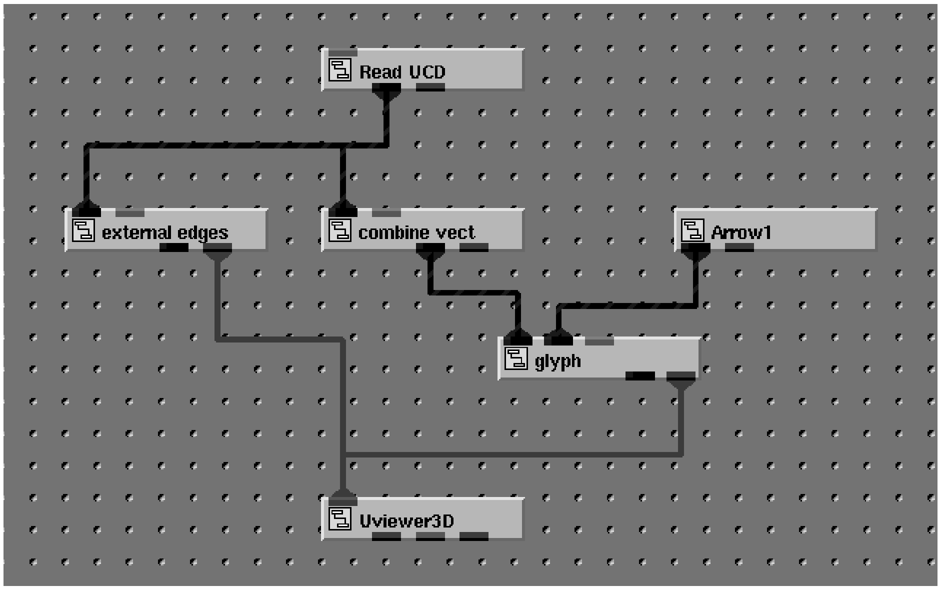
- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces an wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- `external_faces` (Mappers): produces a mesh which represents the exterior, visible faces of an unstructured mesh.
- `Axes3D` (Geometries): creates an XYZ axis grid with labels based on mesh extents.
- `isoline` (Mappers): creates contour lines of constant value.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.5.4 Using the `isolines` module (2)



- Select the file `bluntfin.inp` using the `Read_UCD` module's file browser.
- Select the `isoline` module's control panel.
- Make active the data component density.
- You should have obtained a picture similar to the one above.
- Alter the min and max data values to be contoured and observe the effect.

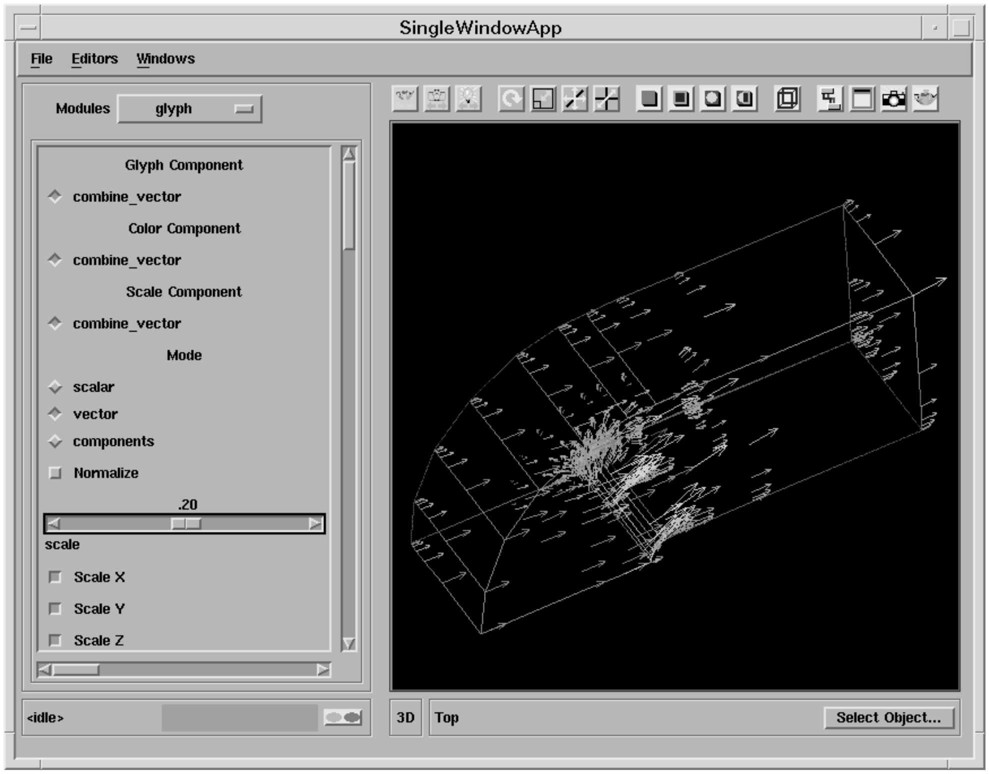
5.5.5 Displaying vector data (1)



Create the network above using the modules listed below:

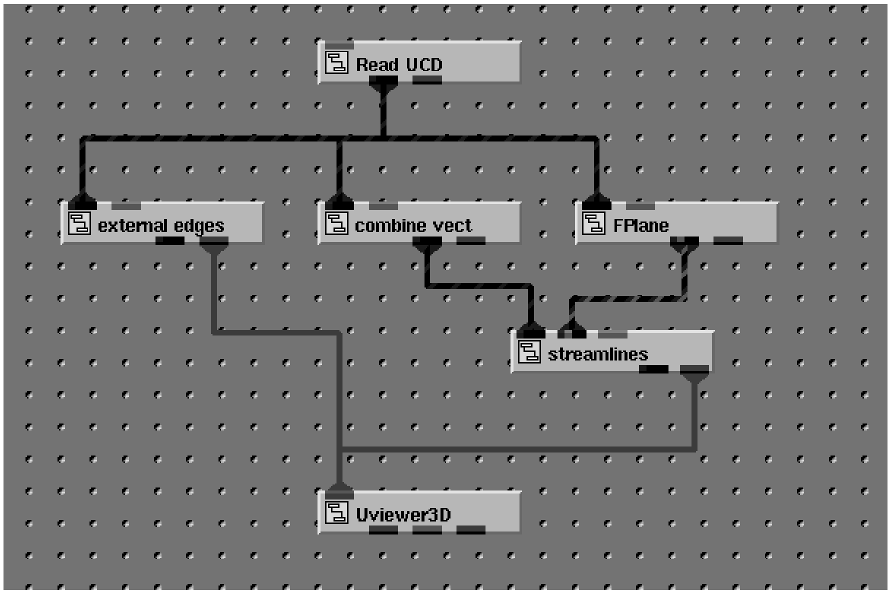
- Read_UCD (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- external_edges (Mappers): produces an wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- combine_vect (Filters): takes all selected scalar data components and combines to output a single vector component.
- Arrow1 (Geometries): creates a wireframe arrow shaped mesh.
- glyph (Mappers): places a geometrical object at each node of an input field. The glyph is coloured and sized according to the magnitude of the data component at that point.
- Uviewer3D (Viewers): creates a 3D image in the DataViewer.

5.5.6 Displaying vector data (2)



- Select the file `bluntfin.inp` using the `Read_UCD` module's file browser.
- Select the `combine_vectors` module and select x-momentum, y-momentum and z-momentum.
- Select the `glyph` module and reduce the **scale** to ~0.2 or until the arrows are a sensible size.
- You should have obtained a picture similar to the one above.

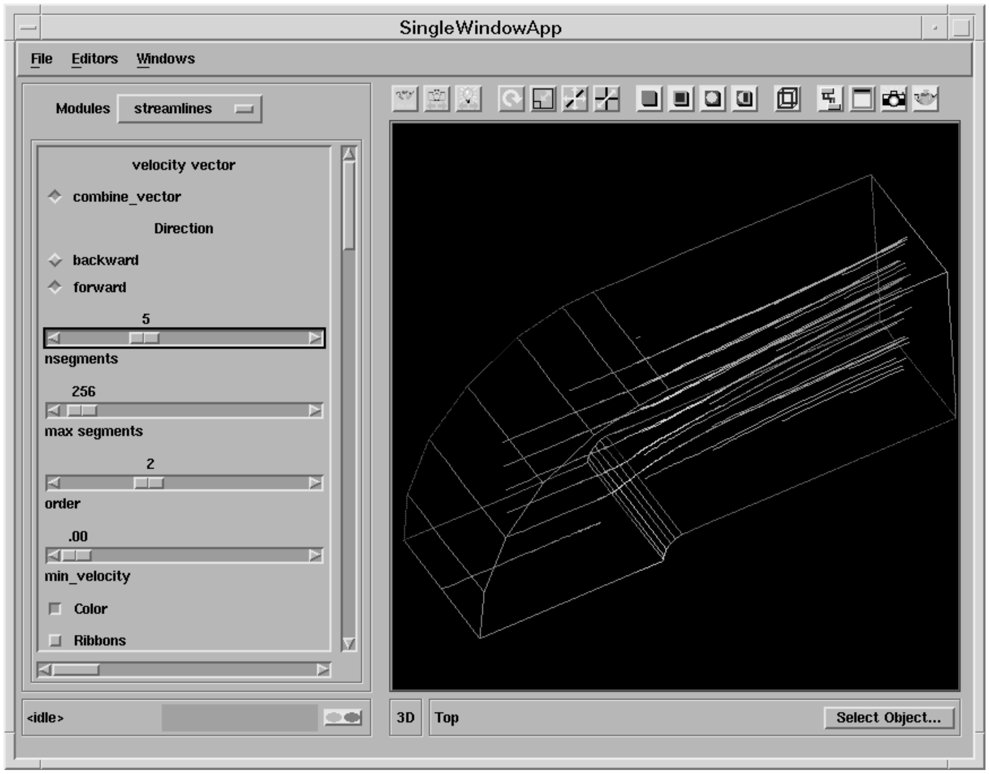
5.5.7 Using the `streamlines` module (1)



Create the network above using the following modules:

- `Read_UCD` (Data IO): reads an AVS UCD (.inp) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces an wireframe representation of the outside of an unstructured mesh to reveal objects inside.
- `FPlane` (Geometries): generates a 3D plane of variable size which can be transformed in three dimensions.
- `combine_vect` (Filters): takes all selected scalar data components and combines to output a single vector component.
- `streamlines` (Mappers): generates streamlines or streamribbons based on a field with a one, two or three element vector component.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

5.5.8 Using the `streamlines` module (2)



- Select the file `bluntfin.inp` using the `Read_UCD` module's file browser.
- Select the `combine_vect` module and ensure that x, y & z momentum are selected.
- Select the `streamlines` module's control panel.
- Try altering the Nsegment value to increase the numbers of streamlines.
- Investigate the effect of changing the minimum velocity value.
- Change the streamlines to streamribbons.
- Try altering some of the `FPlane` module parameters.