THE UNIVERSITY of MANCHESTER

AVS/Express Visualization Edition

An Introductory Course

Student Workbook

International AVS Centre
Manchester Visualization Centre

October 2003

MVC

IAC manchester

Document Editor

The original document was compiled and typeset by Fenqiang Lin and Steve Larkin, Manchester Visualization Centre using the document preparation system LATEX.

The document was compiled using the document publishing software FrameMaker. Additional exercises and improvements to the original text were supplied by Rachel Slinger. The work was undertaken whilst she was working in the Manchester Visualization Centre as a Summer Student Project 1996.

This document has been updated by Tobias Schiebeck, International AVS Center; and Trudy Coleman, Manchester Visualization Centre, in October 2003. It is now prepared using Microsoft Word.

The document is published and distributed by the International AVS Centre (IAC).

For further information on IAC see the following Word Wide Web page:

`http://www.iavsc.org`

or by email to:

`avs@iavsc.org`

or by mail to:

```
International AVS Centre
Manchester Visualization Centre
Manchester Computing
Manchester
M13 9PL
UK
```

Contents

Figures

1 Introduction

1.1 Introduction to AVS/Express – Student Workbook

The *Introduction to AVS/Express* - Student Workbook has been developed by the International AVS Centre and is based on the *Introduction to AVS/Express* course produced previously by the Manchester and North Training and Education Centre (MAN T&EC) based in the Manchester Visualization Centre at the University of Manchester.

1.1.1 What is AVS/Express?

AVS/Express is an object-orientated, visual development tool that enables you to build reusable application components and sophisticated applications. The AVS/Express Visualization Edition provides hundreds of application components for visualizing, analysing, manipulating and displaying data.

1.1.2 Who is the Workbook Aimed at?

The workbook is aimed at those who have a need to visualize both 2D and 3D data. Techniques covered include:

- Reading the data
- Creating 2D and 3D images
- Creating plots

1.1.3 Are there any Prerequisites?

The course is designed for the user with no prior experience of AVS/Express Visualization Edition. The course also assumes no prior experience of AVS5. No programming skills are required but basic experience of working in a Windows environment is required.

1.2 About the Workbook

1.2.1 The Visualization Process

The data visualization process is carried out using a **network**.

A network consists of **modules** and **connections**. Modules carry out certain tasks such as:

- Reading fields
- Mapping
- Filtering
- Data viewing.

The networks are built in the **Network Editor**.

Modules are chosen from the module library and connected in order to carry out a specific application.

Before data can be read into the network a **field file** must be created in order to describe the data to the network. The data is read into the network via the field file; the network carries out its specified function and presents the data visually.

Interaction with the visual data is carried out until the desired analysis has been achieved.

Networks can be saved as Applications and reused to analyse similar data sets.

1.2.2 Structure of the Exercises

Each chapter in this book works through a series of structured exercises in order to familiarise the user with the AVS/Express environment.

- Chapter 2 deals with using the Network Editor:
  - Building networks.
  - Making connections.
  - Reading data.
  - Saving, loading and deleting applications.
- Chapter 3 and Chapter 4 deal with creating field files to describe the 2D and 3D data respectively. Also deals with the 2D and 3D data visualization techniques.
- Chapter 5 deals with visualization of Unstructured Cell Data (UCD).
- Chapter 6 deals with the AVS/Express example demonstrations.

1.2.3 Datafiles and Solutions

Each chapter of exercises uses simple examples of data in order to demonstrate a particular function of AVS/Express. Some data files for these exercises are not part of the standard AVS/Express release. The course assumes they have been made available under the directory:

```
C:\work\avs
```

Copies of these data files can be obtained via the International AVS Centre website:

```
http://www.iavsc.org/training/express/intro/data/intro.tar.gz
```

The answers to the exercises in Chapter 3 and Chapter 4 can be found in Appendix B of this workbook.

2 Using the Network Editor

2.1 Aims of this Chapter

The exercises in this chapter will illustrate the general features of the AVS/Express Network Editor. Exercise 1 is a simple example designed to take you through the data visualization process:

- Starting AVS/Express.
- Constructing networks.
- Reading data.
- Saving, loading and deleting applications.
- Exiting from AVS/Express.

Exercise 2 is a similar example for you to attempt on your own. In both cases, it is not necessary to create a field file in order to read the data. The creation of field files will be covered in the subsequent chapters.

2.2 Exercise 1

In this example we wish to visualize the electron density within a hydrogen atom. In particular we want to take a slice through the atom and observe different electron densities throughout the atom.

2.2.1 Starting the AVS/Express Visualization Edition

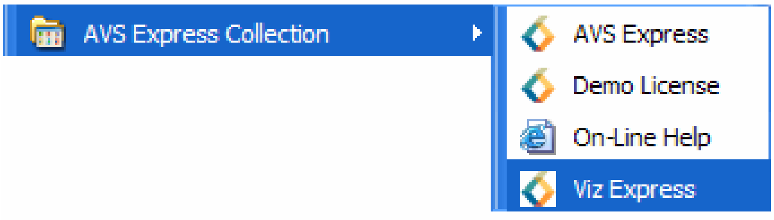

Figure 2.1: **Start** Menu

- To start AVS/Express Visualization Edition, select AVS Express Collection folder from the Start menu and load the Viz Express application as shown in Figure 2.1.
- At the start-up of AVS/Express the Network Editor is in the Start Library (Figure 2.2). This library enables new users to create and master AVS/Express applications quickly. The **Data Import**, **Visualization**, and **View Export**, sub-libraries provide a means for new users to create simple applications.
- A guide to using the wizards can be found in Appendix C, but for now will not be used.
- Select **File**, **New Application** from the tool bar to load a new application.

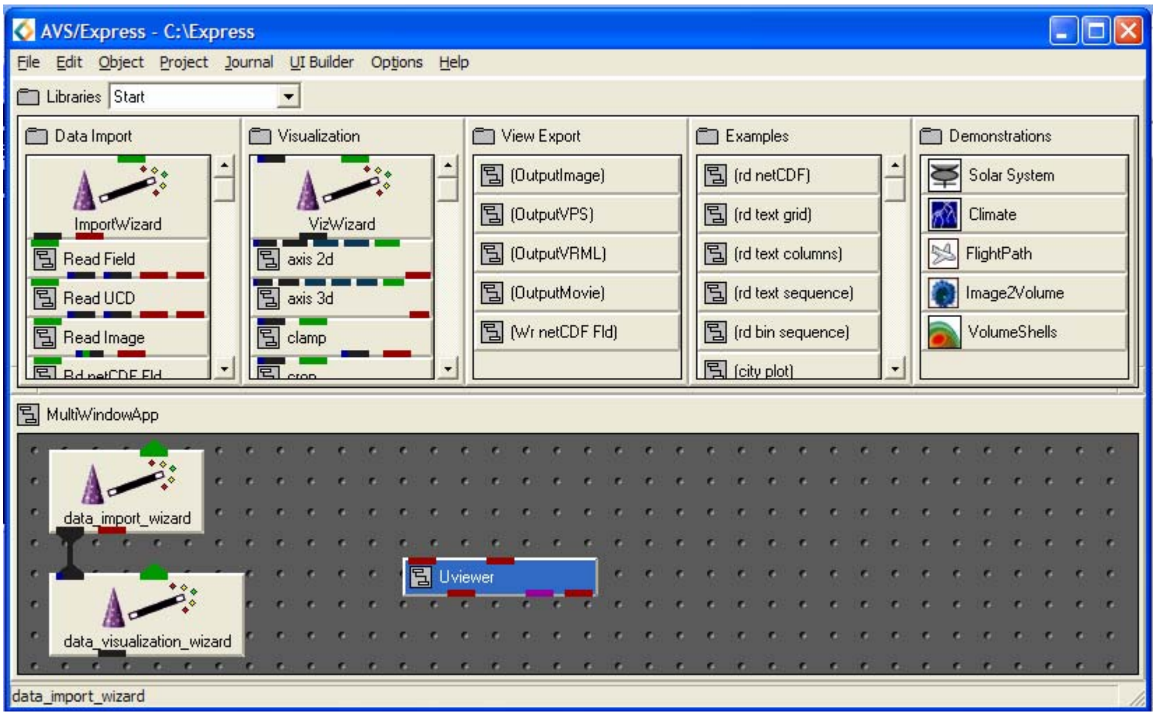

Figure 2.2: Network Editor in the **Start Library**

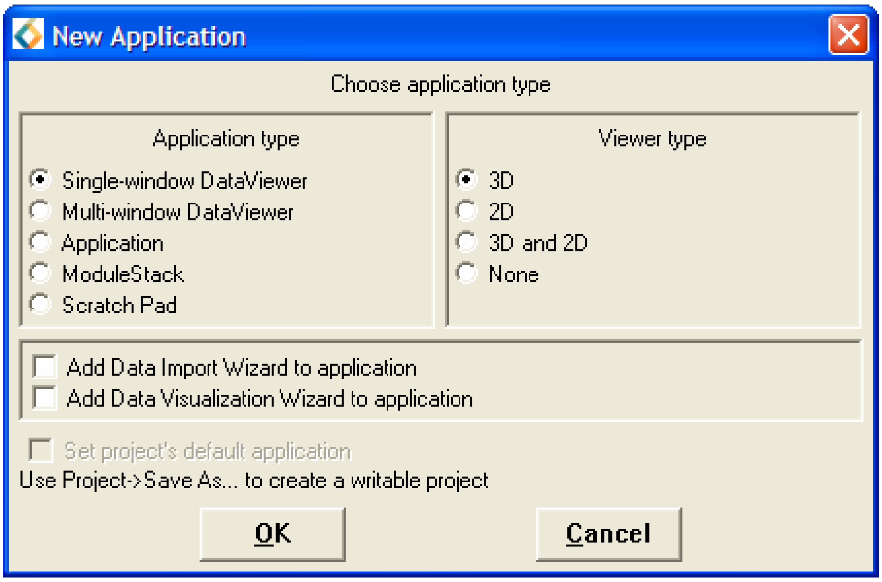

Figure 2.3: New Application Pop up window

- A New Application pop up window will appear in which the initial type of application and viewer required can be selected (Figure 2.3).
- Choose **Single-window Data Viewer** with a **3D** viewer type (usually the best choice when constructing simple applications).
- Click **OK** to proceed.
- When the new application loads although the Wizard icons will no longer be shown in the application window, the Start library modules will still be displayed.
- Select **Main** from the **Libraries** drop down menu in order to display the modules required for this exercise (Figure 2.4).

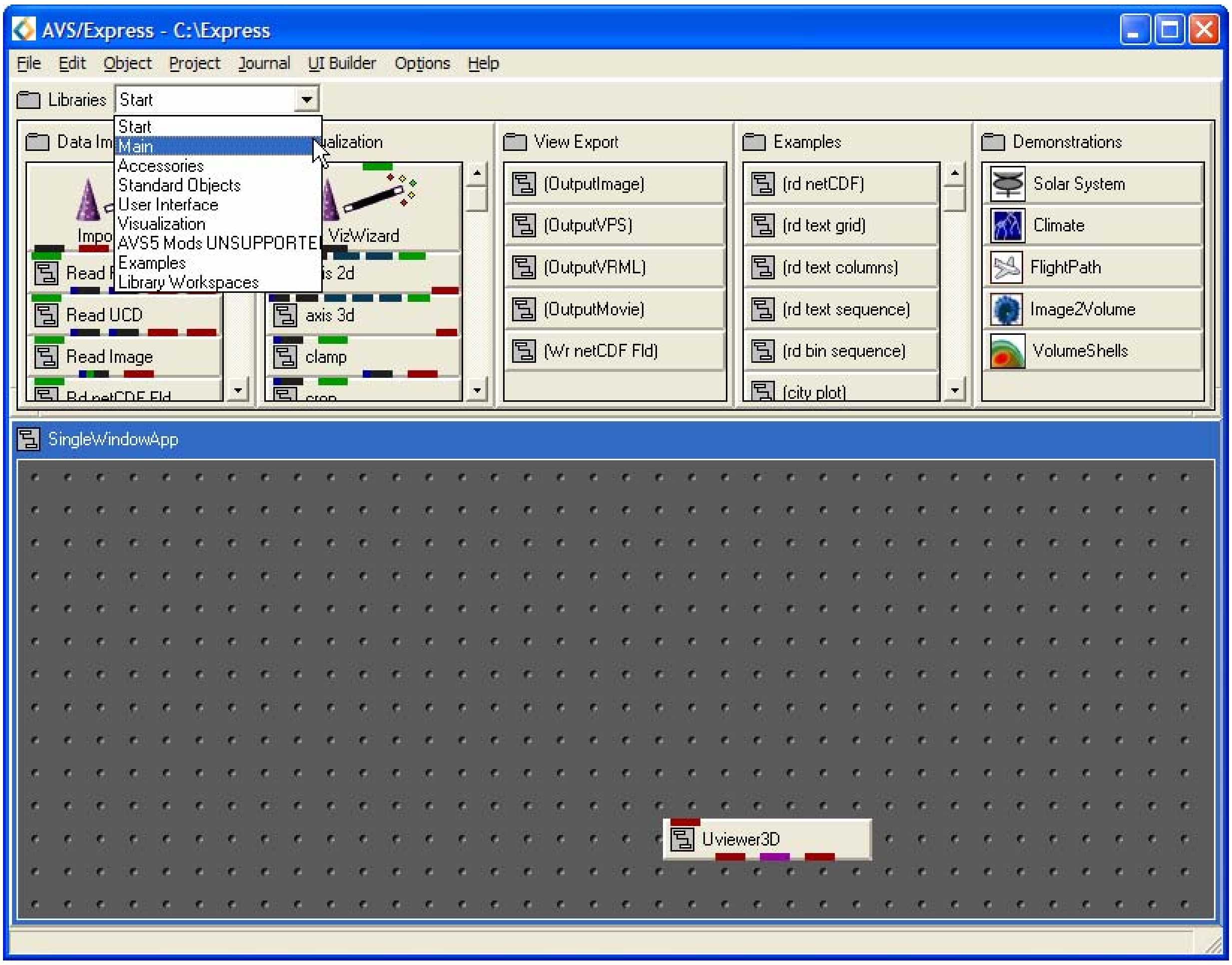

Figure 2.4: **Libraries** Drop down Menu

2.2.2 Structure of the Network Editor

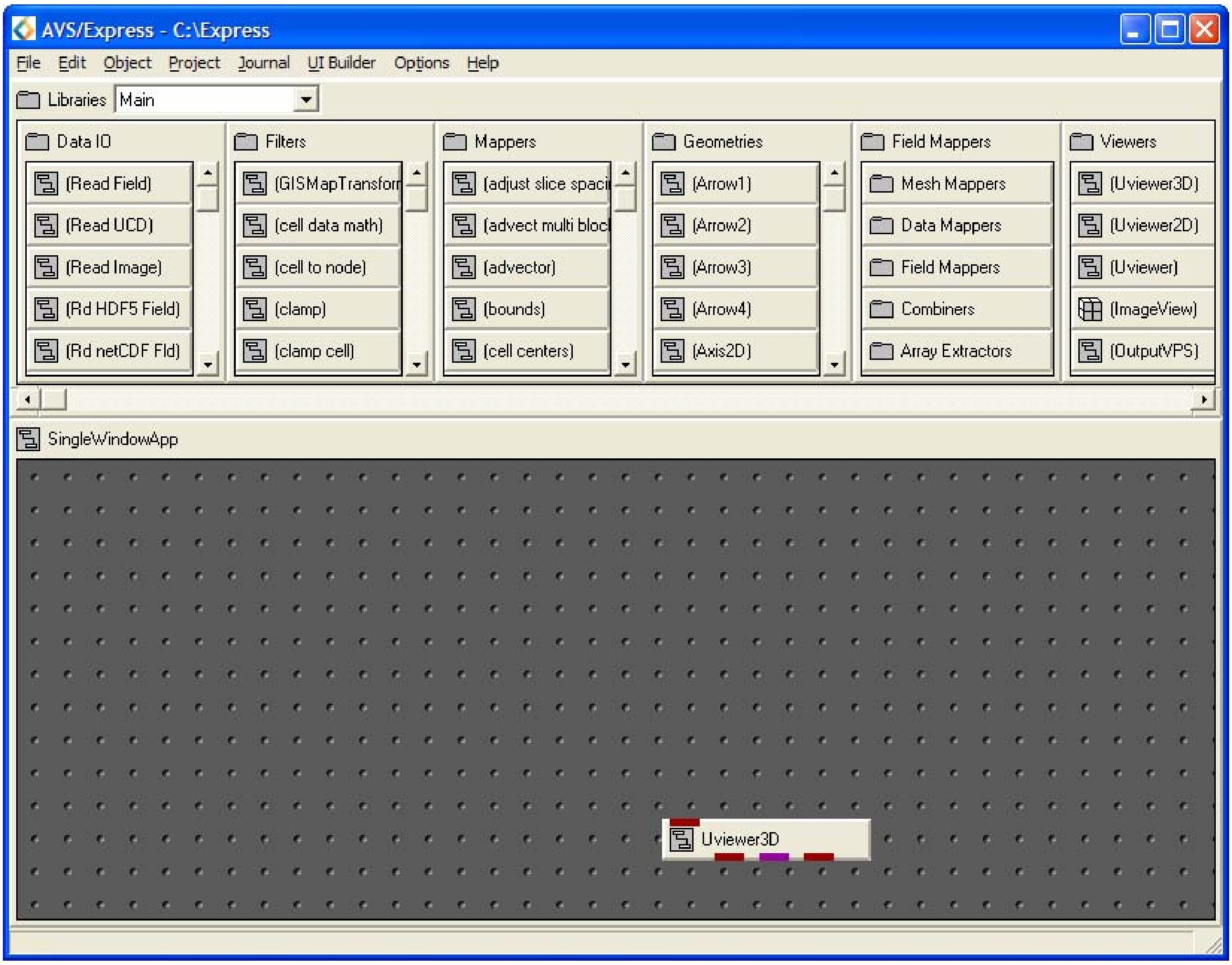

Figure 2.5: Network Editor in the Main library

- A layout similar to that of Figure 2.5 should now appear, showing the major components of the Network Editor.
- The library of modules known as **Main** is selected and on view above the application workspace. This library contains the major modules for network building. (Look at the drop down menu to observe the other available libraries.) Within the main library the modules are divided into sub-libraries:
  - Data IO
  - Filters
  - Mappers
  - Geometries
  - Field Mappers
  - Viewers
- **Data IO** contains the various modules for reading the data.
- **Viewers** contains the various modules for viewing data.
- The other sub libraries contain modules that are used to build up the intermediate parts of a network in order to manipulate the data for a specific application.

2.2.3 Using the Object Finder

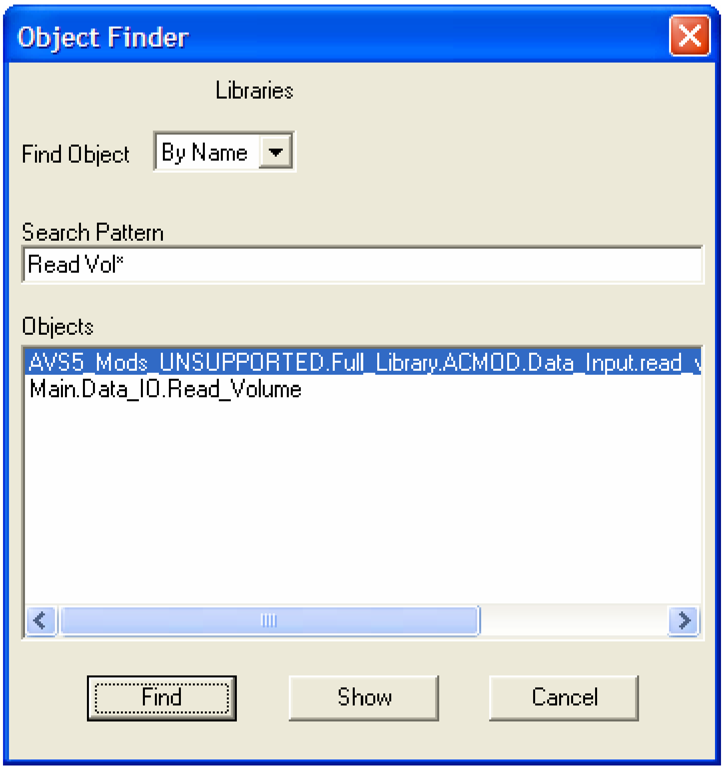

Figure 2.6: Object Finder

When selecting modules for a network it is possible to either scroll through the sub-libraries until the correct module is located or alternatively the **Object Finder** (Figure2.6) can be used to search for a specific module. A module can either be searched for in all libraries or in a specific library.

- *To search all the AVS/Express libraries*: choose, **Find in All Libraries** from the **Object** menu, (located in the menu bar along the top of the Network Editor), and type the module name in the **Search Pattern** box. Select **Find** and the location of the module will be given in the **Objects** box. If you wish the location of the module to be highlighted in the Network Editor library select **Show**.
- *To limit the search*: select the library or library page by clicking on it using the **left** mouse button. Now hold down the right button to display the popup menu, and choose the **Find** popup command. Alternatively select the library page, then select **Find in Selected Library** from the **Object** menu and follow the same procedure as before.
- Wildcards * can be used to increase search flexibility. For example the above diagram shows a search for any modules which read volume data – type "Read Vol*" in the search pattern field.

2.2.4 Selecting a Module

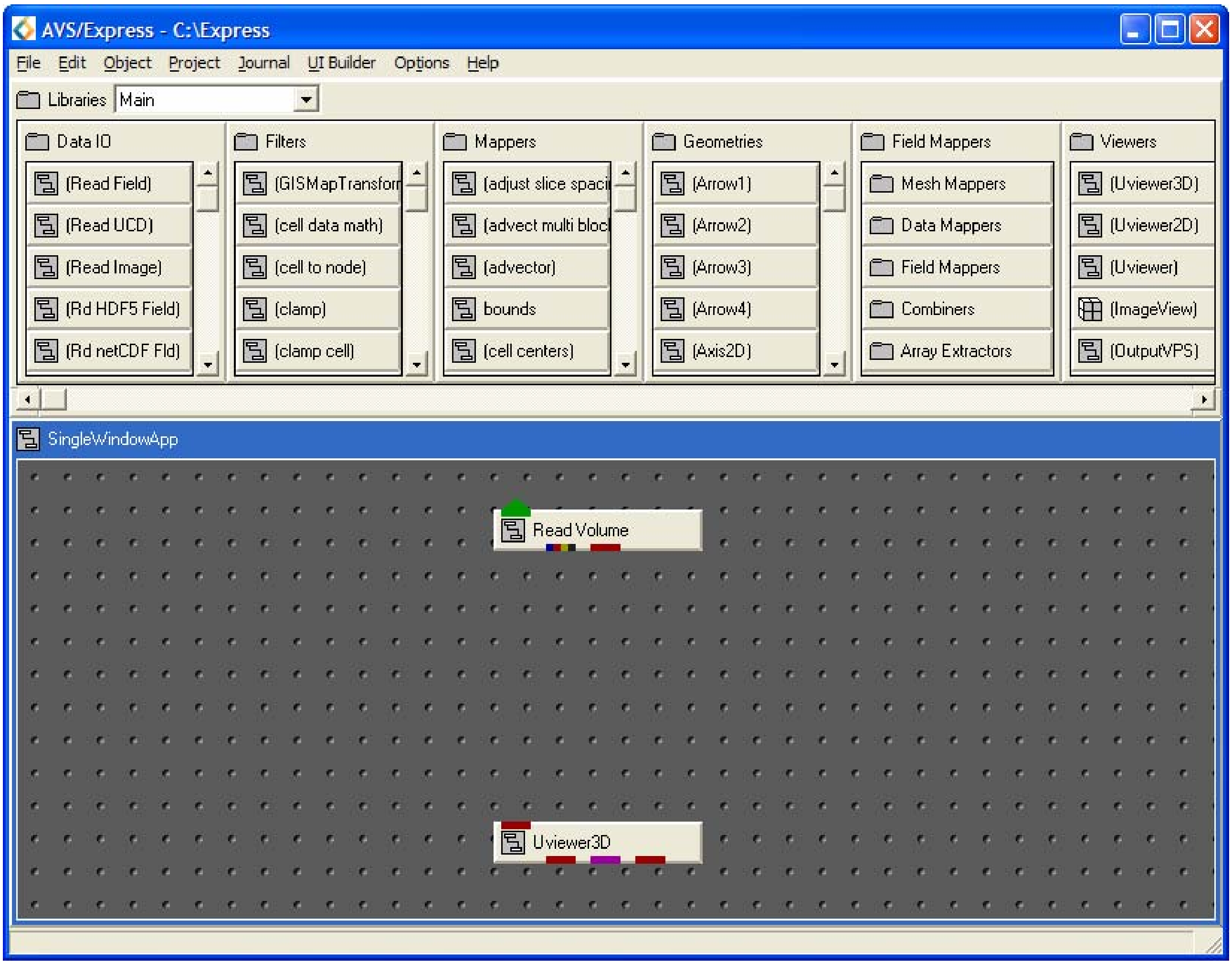

Figure 2.7: Network Editor containing a Data IO and Viewers Module

- Find the `Read_Volume` module which is located in the **Data IO** library.
- Point to the module using the mouse and press the **left** button.
- While keeping the mouse button pressed drag the module onto the application workspace. When you are happy with its location release the button.
- The module will be instanced to the workspace as shown in Figure 2.7.

2.2.5 Deleting a Module

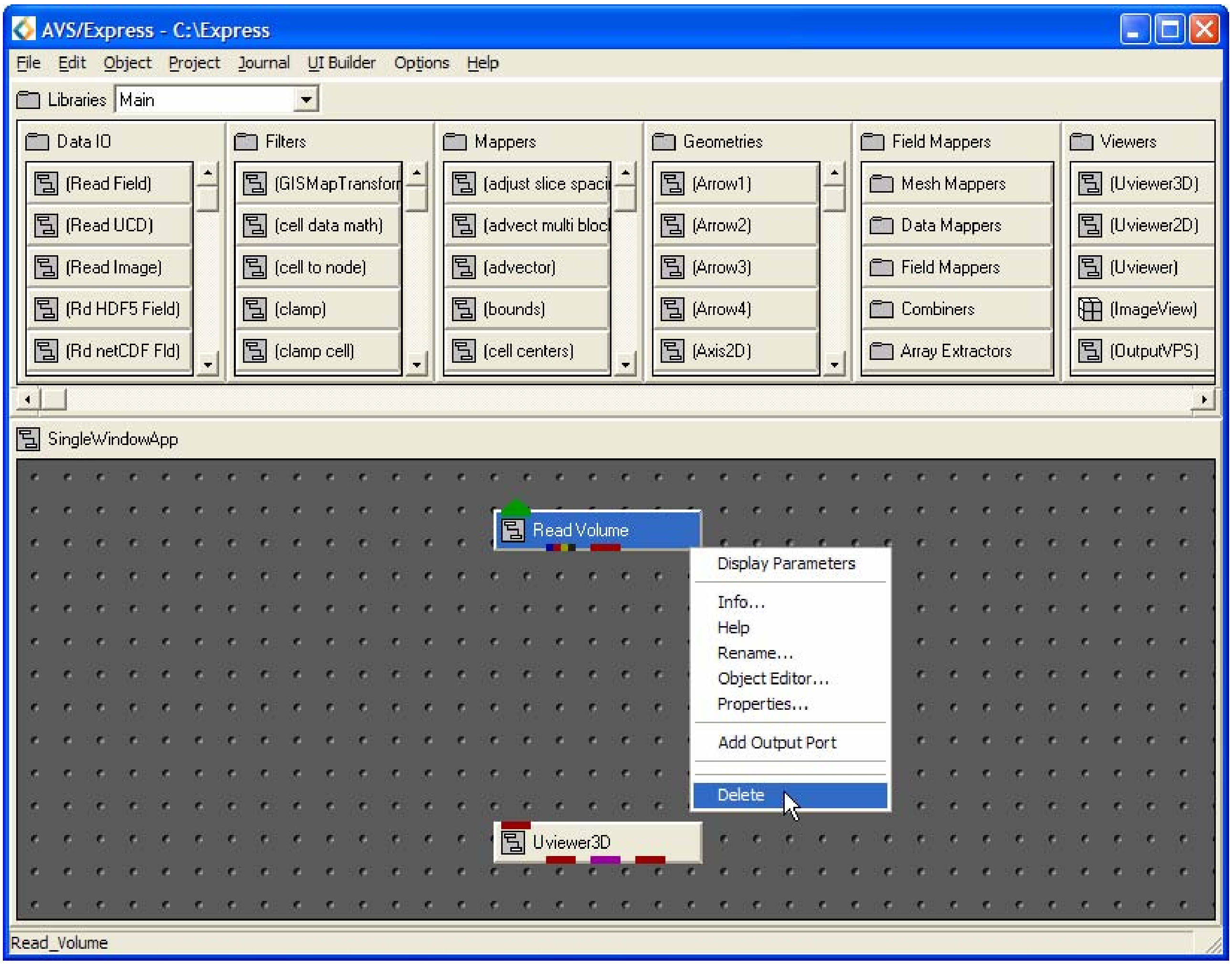

Figure 2.8: Module pop up menu

To delete a module:

- Select it.
- Hold down the **right** mouse button to display the popup menu.
- Choose the **Delete** popup command.
- When the mouse button is released the module will be deleted.

2.2.6 Accessing the Help Pages

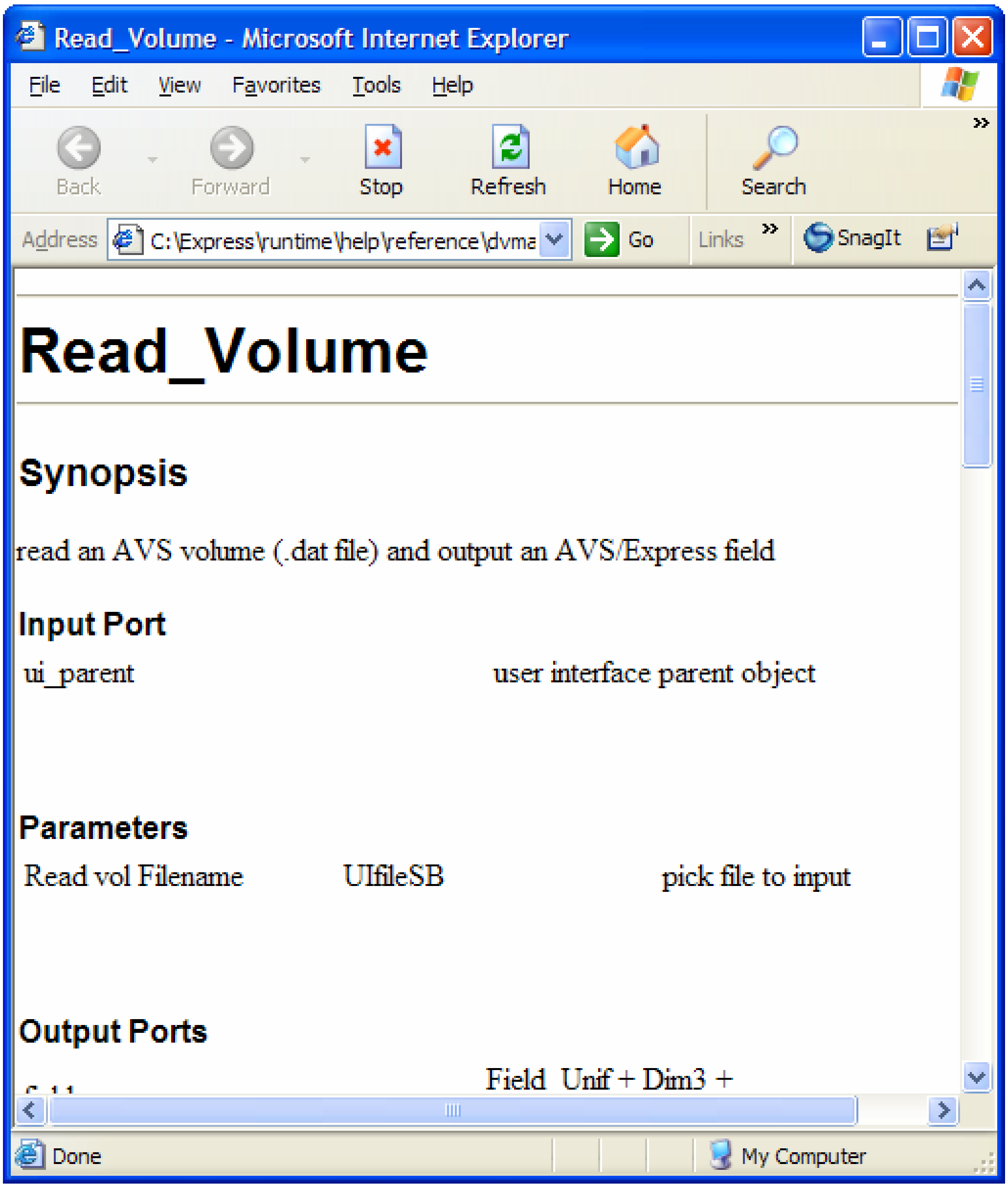

Figure 2.9: `Read_Volume` Help Page

To access the help pages for a particular module:

- Select the module and hold down the **right** mouse button to display the popup menu.
- Select the **Help** command

Alternatively:

- Select the module
- From the **Help** menu at the top right of the Network Editor choose **On Selected Object**.

Both these routes will cause the appropriate help page to be displayed.

2.2.7 Selecting the Other Modules

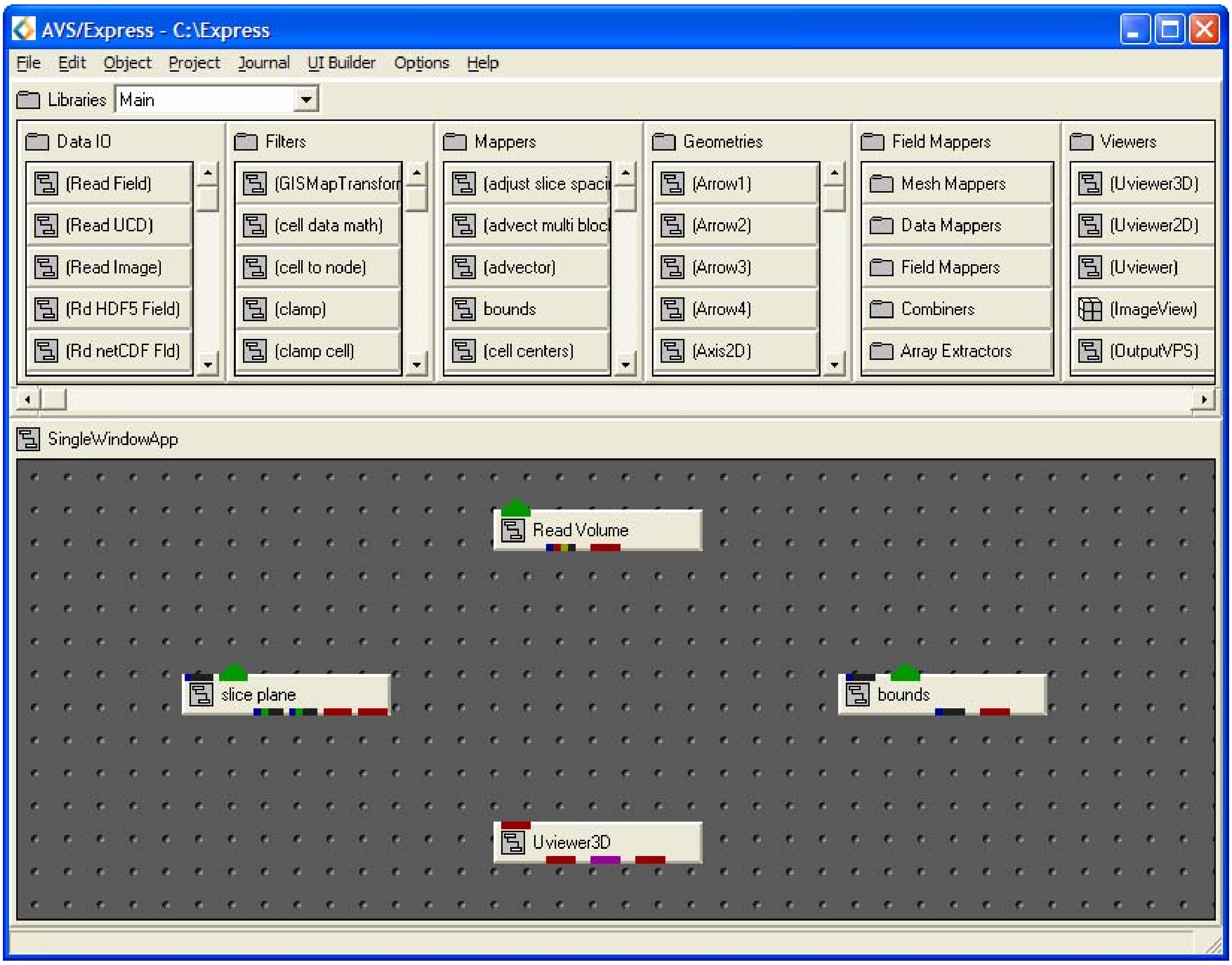

Figure 2.10: Selection of Modules in the Network Editor Workspace

Select the modules in the list below from the given sub-libraries and arrange them in the workspace as shown in Figure 2.10.

- `Read_Volume` (Data IO): reads a volume format file and converts it to a 3D AVS/Express field.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `slice_plane` (Mappers): extracts a 2D slice from a 3D field with an arbitrarily positioned slice plane.
- `Uviewer3D` (Viewers): creates a 3D image in the DataViewer.

Attempt to locate one or more of the modules using the **Object Finder**.

2.2.8 Connecting the Ports

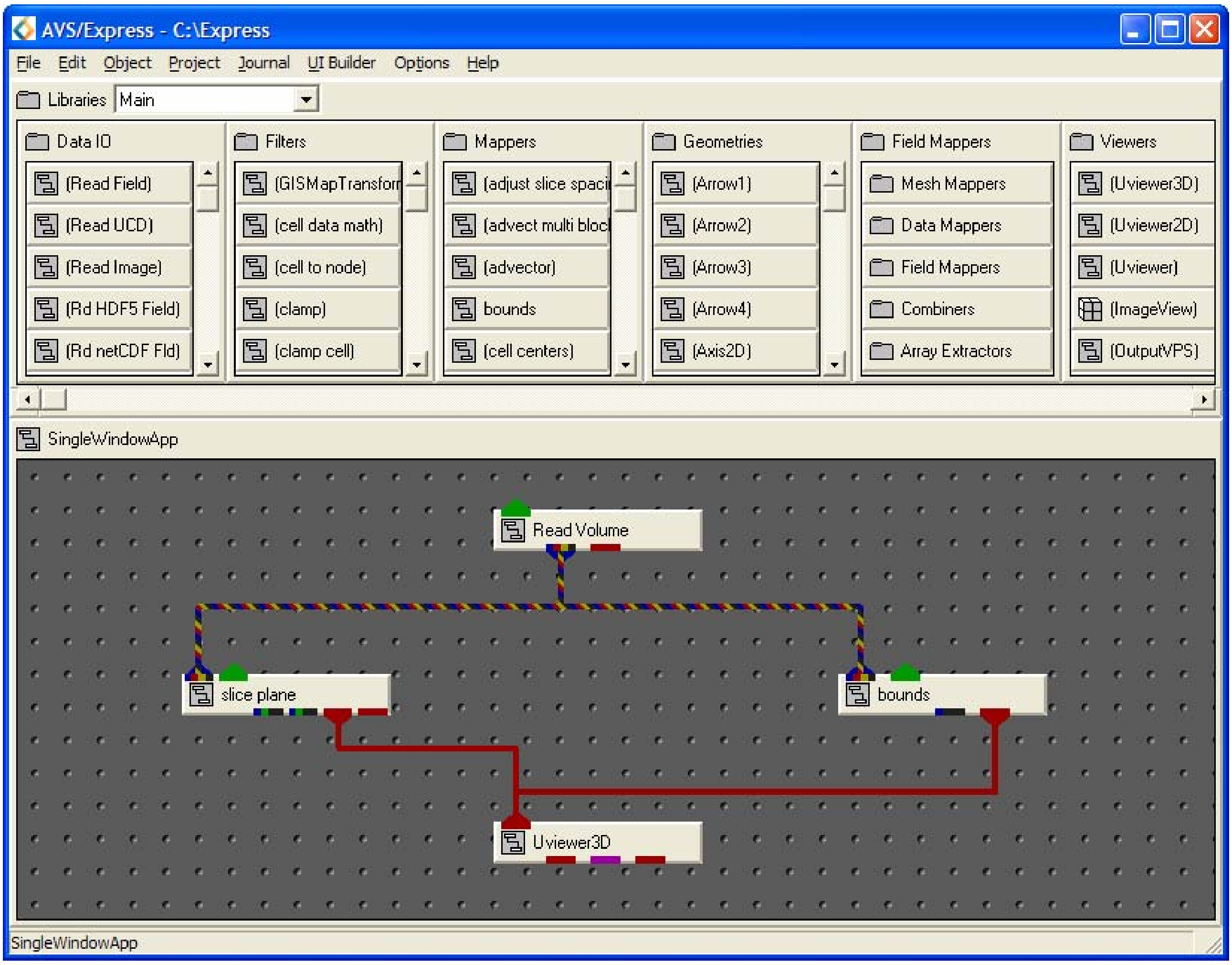

Figure 2.11: Connected Modules (Network)

Connect the modules together to form the network shown in Figure 2.11. It is important to make sure that the correct input/output ports are connected or the network will not perform the correct function. (Often AVS/Express will not allow incorrect connection to be made or will produce an error message if one is made.)

- Connecting modules:
  - Position the mouse pointer above the port you wish to connect.
  - Pressing the **left** button will display all of the available connections.
  - Keeping the button pressed move the mouse towards the module port to which you wish to make the connection. The connection chosen will be highlighted.
  - Release the mouse button to make the connection.
- Disconnecting modules:
  - Position the mouse pointer above the port you wish to disconnect.
  - “Remake” the connection i.e. press the **left** button while positioned on this port, move towards the connected port and when the connection is highlighted release the button.
  - The connection will be removed.

2.2.9 Reading a Data File (1)

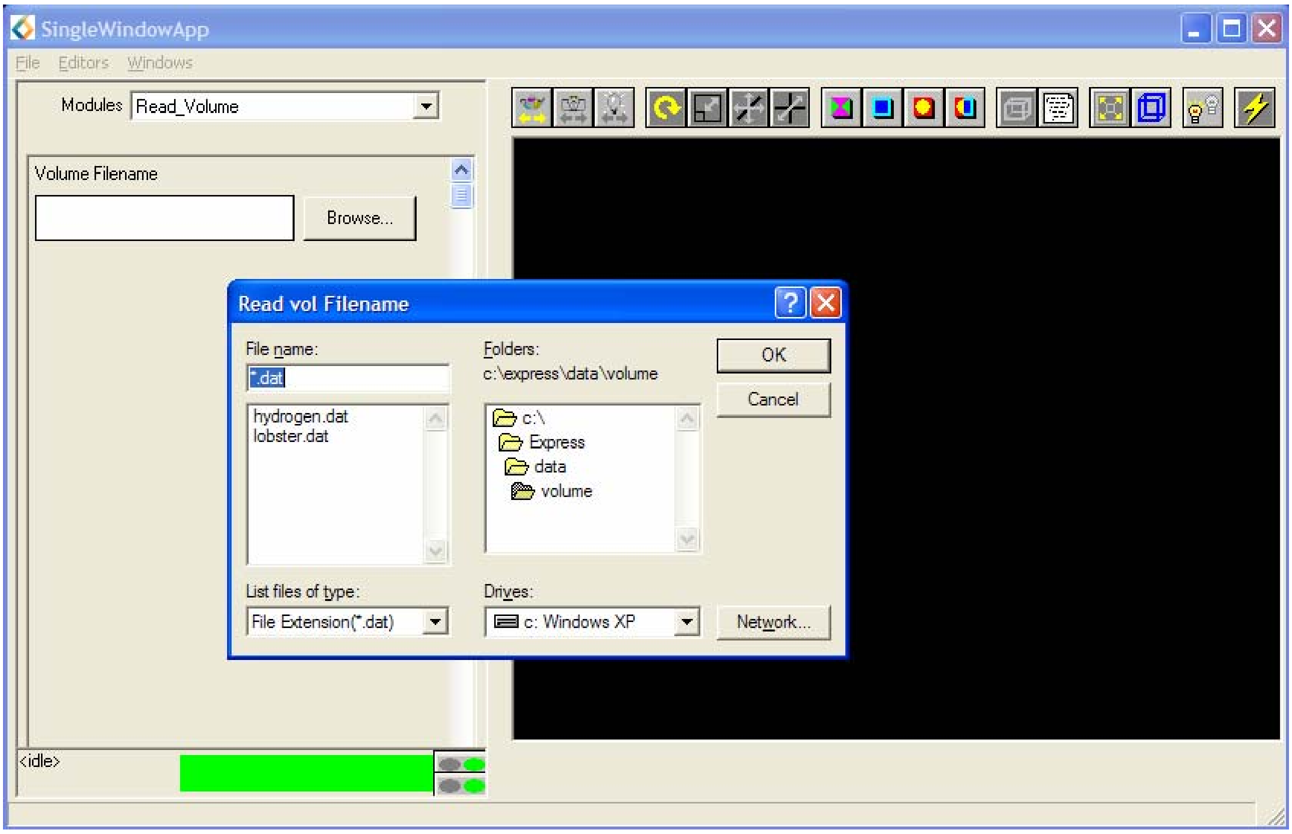

Figure 2.12: Data Viewer Pad with File browser

When a module is instanced to the workspace you can access controls for the module in the Editor Panel of the **DataViewer Pad**.

- Select the **Modules** command from the **Editors** pull-down menu. A menu of all modules instanced to the workspace will appear.
- Choose `the Read_Volume` module and select the **Browse** button. The file browser will now appear (Figure 2.12).
- Use the file browser to change the directory `c:\express\data\volume` and click on the `hydrogen.dat` file.
- The network will execute producing an image in the viewer window located in the **DataViewer pad.**

2.2.10 Reading a Data File (2)

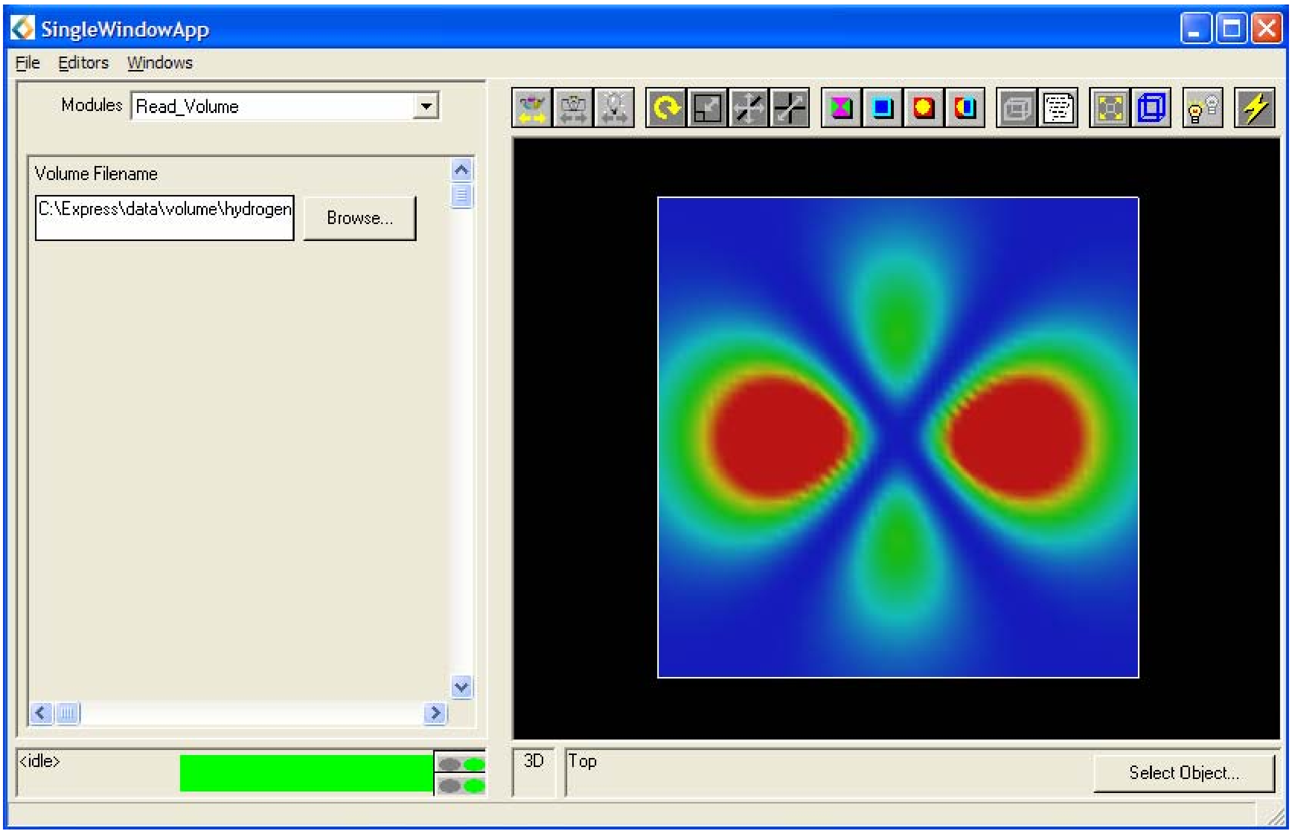

Figure 2.13: Data Viewer Pad

- The DataViewer window should display an image similar to that in Figure 2.13.
- After an image has been obtained, it is necessary to manipulate it into the desired format.

2.2.11 Interacting with the Objects (1)

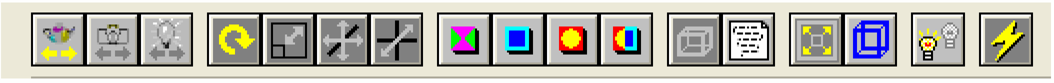

Figure 2.14: Data Viewer Interface

- The Toolbar located in the **DataViewer** interface, provides a quick way to open the **DataViewer components** and to manipulate images in the **DataViewer** window.
- Move the mouse over the icons to reveal a small label indicating the action the icon provides a short cut to, e.g.
    - Transform mode selection, rotation, translate xy, translate z, and scale.
    - Reset, normalize and centre objects.

2.2.12 Interacting with the Objects (2)

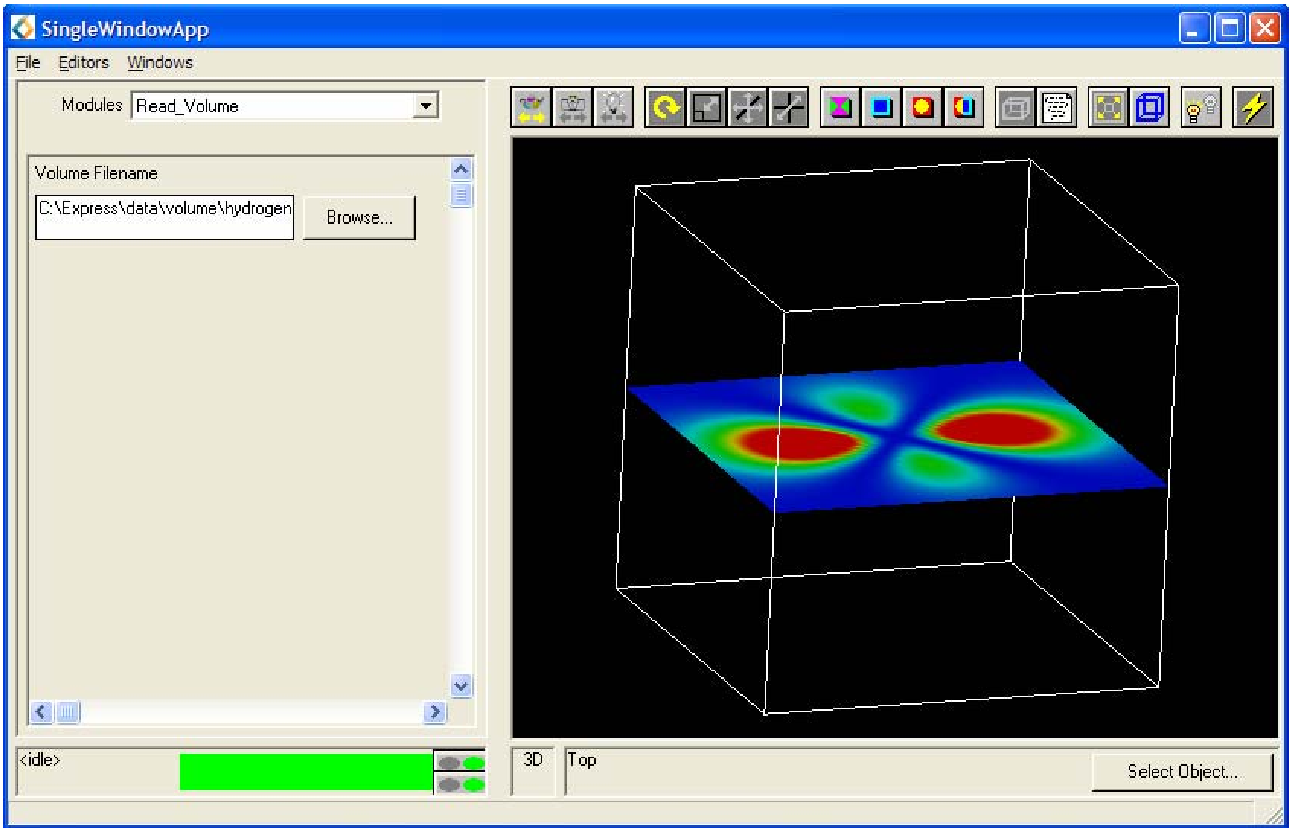

Figure 2.15: Transformed Image in the Data Viewer Pad

- Click on the **Reset/Normalise/Center** icon, so that the full image can be seen in the viewer window.
- Now perform a few simple transformations on the image in the **DataViewer window**:
  - **Scaling**: click on the Scale icon in the Toolbar, place the cursor in the window and while holding down the **left** mouse button, move the cursor up and down to enlarge and shrink the object.
  - **Rotating**: click on the Rotate icon in the Toolbar, place the cursor in the window and while holding down the **left** mouse button, rotate the object.
  - **Translating**: click on the Translate icon in the Toolbar, place the cursor in the window and while holding down the **left** mouse button move the mouse to translate the object in the window.
- Manipulate the object until the scene appears similar to the one shown in Figure 2.15.

2.2.13 Changing the `slice_plane` Parameters

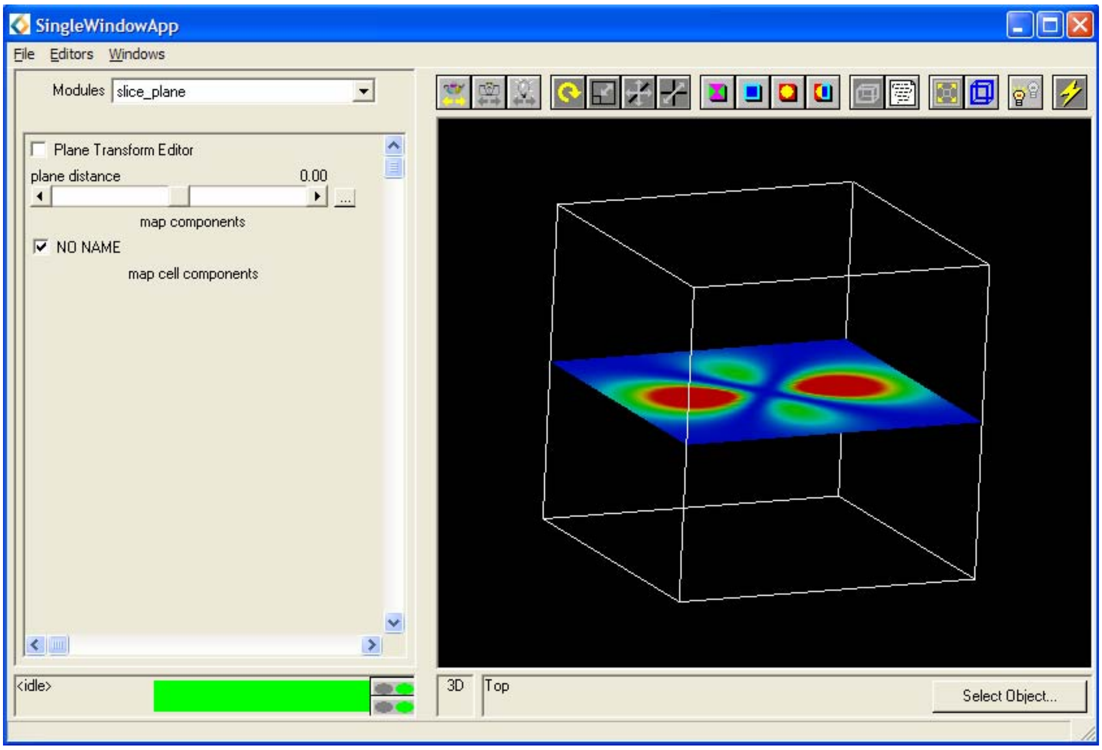

Figure 2.16: `Slice_Plane` Control Panel

- From the **Modules** command from the **Editors** pull-down menu and select the `slice_plane` module to open the Editor Panel. From here the parameters of the slice plane can be altered.
- It is possible to alter the position of the slice through the atom by changing the value of the **plane distance** slider.
- Turning on the **Plane Transform Editor** (Figure 2.16) will load a window containing a series of sliders. These sliders can be used to scale, rotate and transform the slice in the x, y and z directions.
- Alter some of the other slider parameters in order to alter the position of the slice.

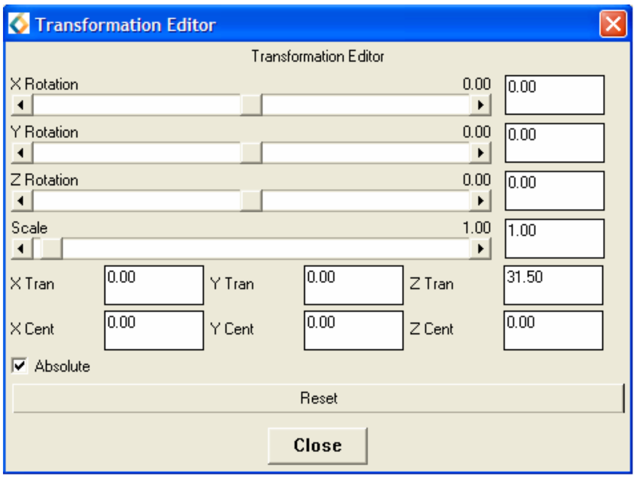

Figure 2.17: **Transformation Editor**

2.2.14 Saving Applications

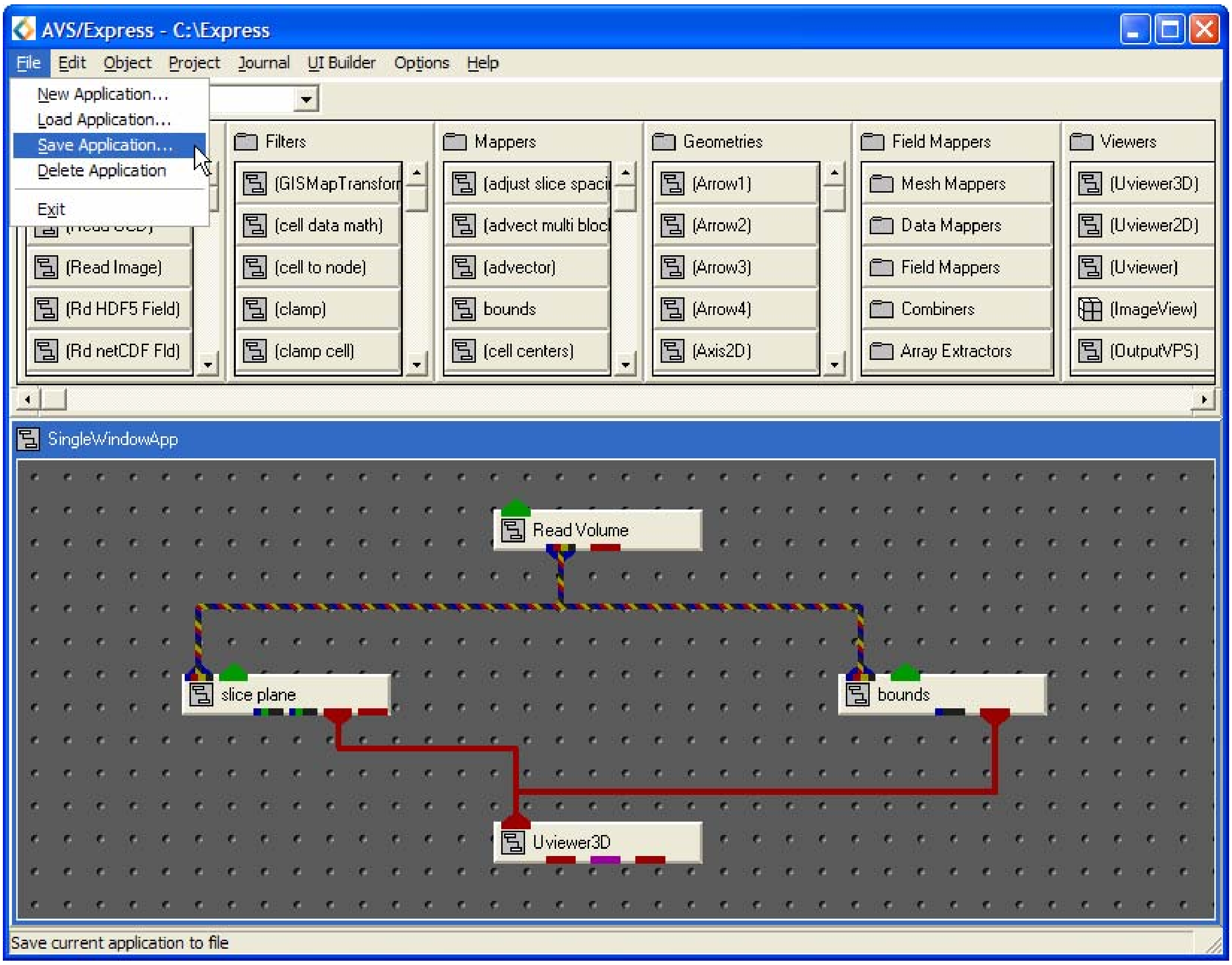

Figure 2.18: **Save Application**

- The network can be saved as an application by selecting the **Save Application** command from the **File** pull-down menu.
- A file browser will appear showing the files in the current working directory (Figure 2.19). Use the file browser to change the current directory to a directory suitable for saving the file in. (It is best to save networks in your home directory.)
- A good convention is to always save the application network by typing in the file name followed by the file extension "`.v`", e.g. "`fred.v`".

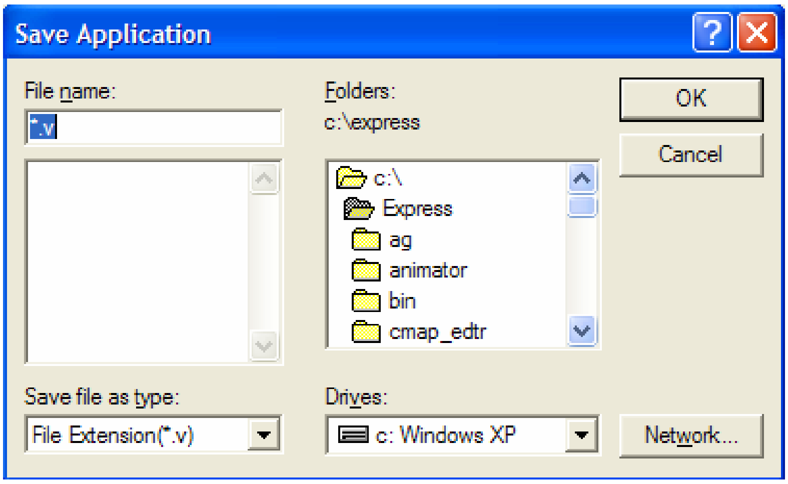

Figure 2.19: **Save Application** File Browser

2.2.15 Deleting an Application

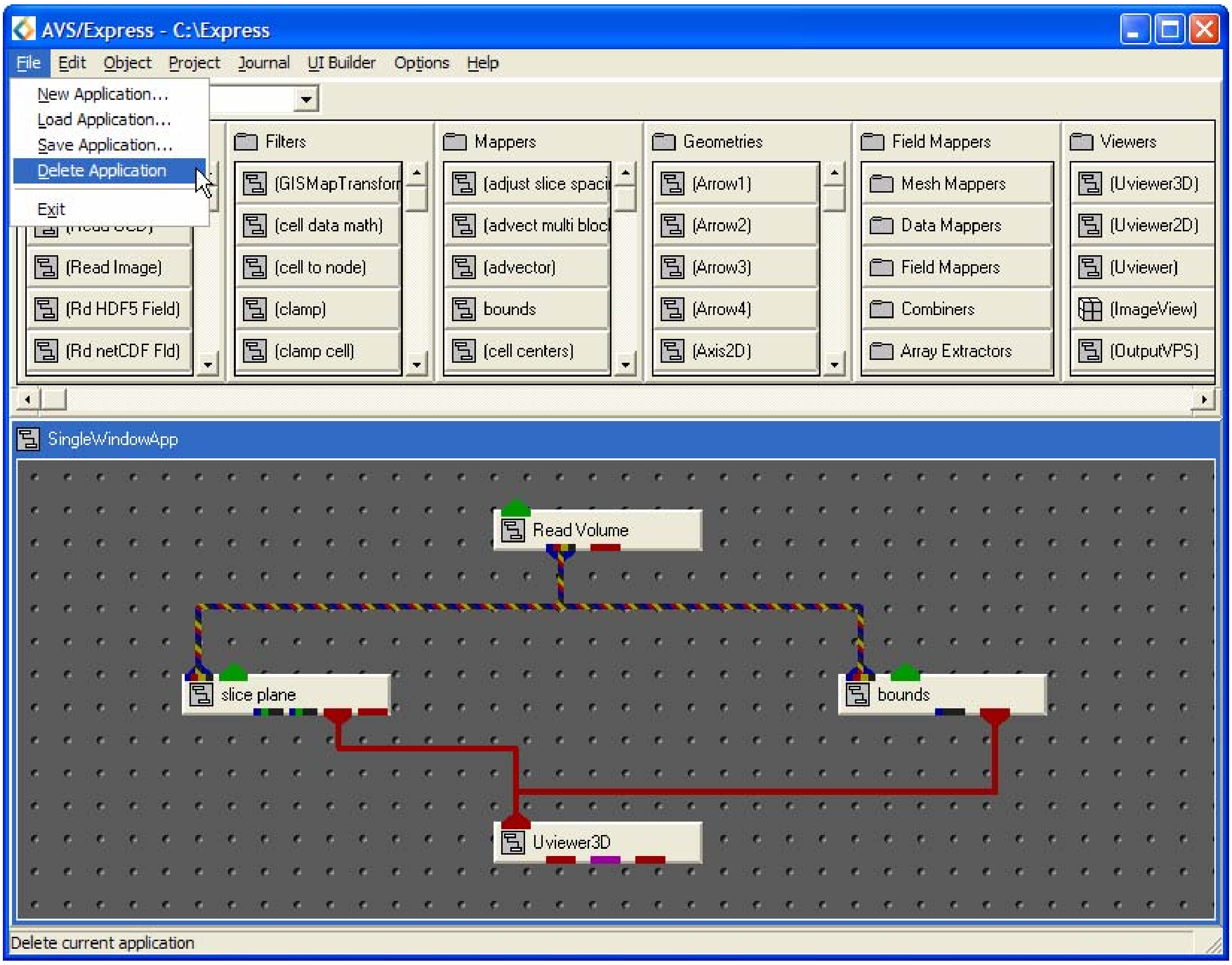

Figure 2.20: **File** Drop down Menu and **Delete Application** Option

To clear an existing application network from the workspace:

- Select the **Delete Application** command from the **File** pull-down menu.

2.2.16 Loading an Application

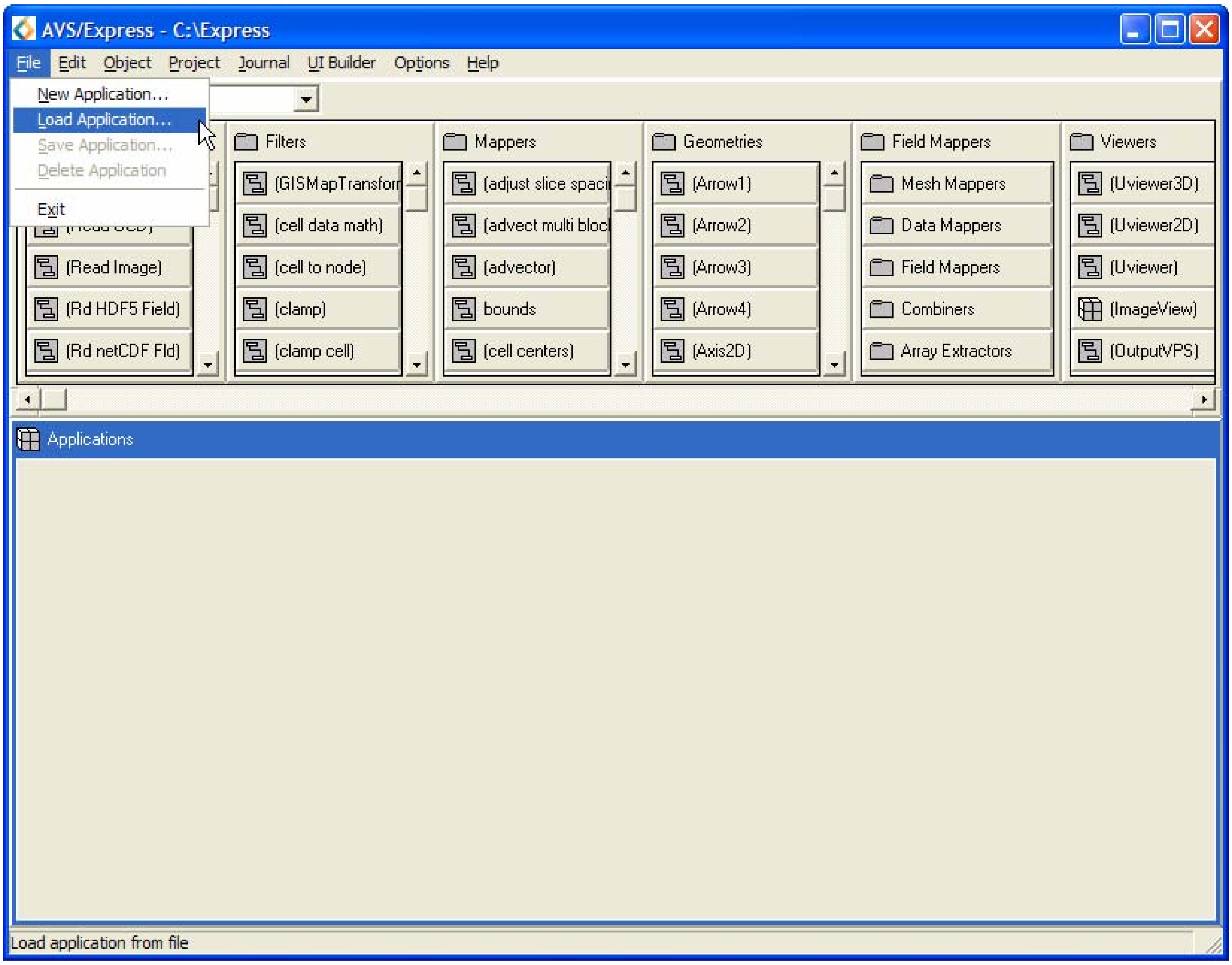

Figure 2.21: **Load Application**

- To load an application network select the **Load Application** command from the pull-down **File** menu. A file browser will appear showing the files in the current working directory (Figure 2.21).
- Use the file browser to change the current directory to the one containing the saved application network.
- Click on the file name in the file browser window and select **OK** to load the application network.

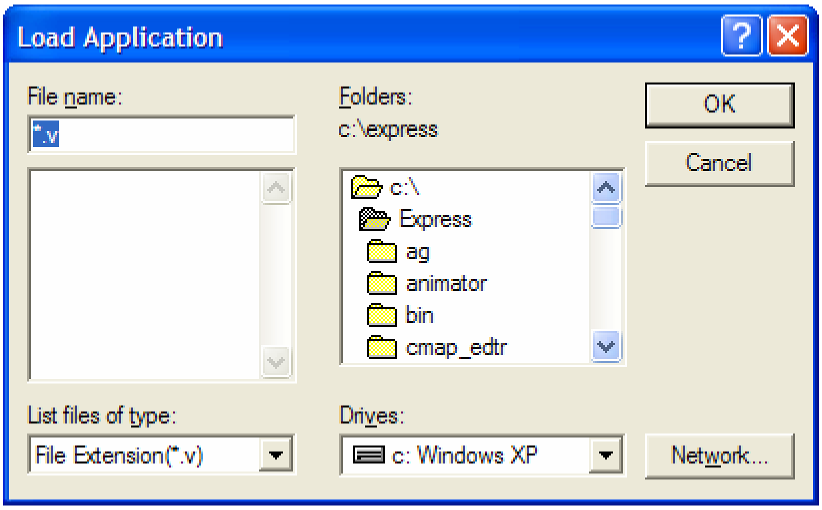

Figure 2.22: **Load Application** File Browser

2.2.17 Exiting from AVS/Express

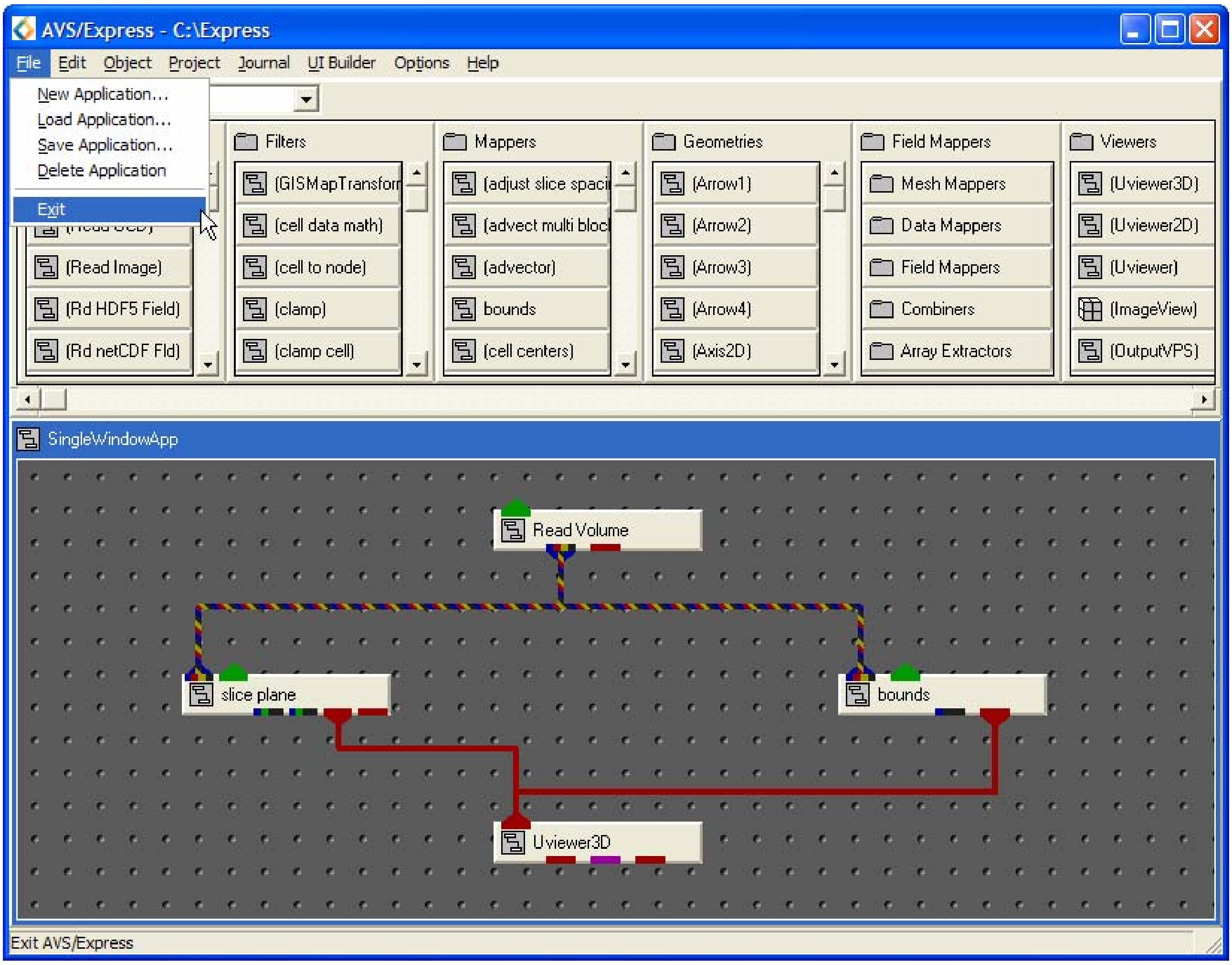

Figure 2.23: Exit

- To exit from AVS/Express select the **Exit** command from the pull-down **File** menu.
- A pop up window containing “Do you really want to exit?” will appear (Figure 2.24).
- Select the **OK** button to safely exit AVS/Express.

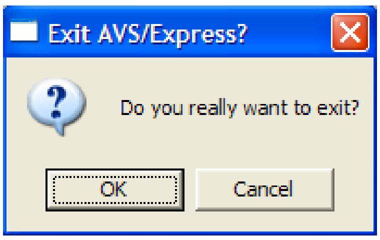

Figure 2.24: **Exit AVS/Express** Pop up Window

2.3 Exercise 2

The variation in bone density through the body of a lobster will be visualized and investigated within this exercise.

- Use the same network as the one constructed in Exercise 1.
- When using the volume browser select the file `lobster.dat` and click **OK**.
- Using similar manipulation techniques as those described in Exercise 1, investigate the bone density of the lobster within its exoskeleton.
- Select the `bounds` module from the **Modules** menu and alter some of its parameters.

3 Visualizing 2D arrays of Data

3.1 Aims of this Chapter

- Attempt the field file examples:
  - Understand the structure of field files.
  - Gain the ability to create field files
  - Solutions are given in the Appendix B.
- Work through several structured exercises which cover visualization techniques for 2D data:
  - Creating field files
  - Constructing networks
  - Manipulating images.
  - Solutions to the field files in the 2D data examples are given in Appendix B.

3.2 Creating Field Files

Field files are used to represent arrays of data in AVS/Express. The data is described by the field file which in turn reads the data into the network via the `Read_Field` module.

3.2.1 Field File Format

A basic field file has the following format[1], although a particular field file need not have all the components shown:

```
#   AVS field file
#
ndim=?
dim1=???
dim2=???
nspace=???
veclen=???
data=???
field=???
label=???
unit=???
variable n file=??? file type=??? skip=??? offset=??? stride=???
coord n file=??? filetype=??? skip=??? offset=??? stride=???
```

- The file should have the file extension "`.fld`" (you may want to create a sub directory to keep all your field files in).
- The first line should read "`#    AVS field file`".
- The next line, also beginning "#" could be a line describing the data. This way it is obvious which application the field file refers to.

[1] A one line return must be entered into the text editor after the field file information, in order to be read by AVS/express.

3.2.2 Defining the Data Structure

- The dimensions of the data must be specified, both in computational space and as it is passed to AVS/Express, in coordinate space.
  - `ndim` and `nspace` give the number of dimensions in computational and coordinate space, respectively.
  - `dim1`, `dim2` and `dim3` specify the size of the x, y and z dimensions.
- The number of data values per element is specified by `veclen`. For example, if there is a value of temperature and pressure for each element in the data set then `veclen=2`.
- The type of the data components is given by `data`. The data components can be `integer`, `float`, `double` or `byte`.
- The mapping type of the field is given by `field`. The mapping type can be one of 3 types:
  - **Uniform**: data is not transformed as it is imported. Used to import regular arrays of data. Number of dimensions in computational and coordinate space must be the same.
  - **Rectilinear**: each dimension in the data has explicit coordinate mapping, but the spacing along the coordinate axis need not be the same. Number of dimensions in computational and coordinate space must be the same.
  - **Irregular**: each element is mapped to a point in coordinate space, e.g. 1D scatter to 3D coordinates. Number of dimensions of computational and coordinate space must be different.
- The data components in the dataset can be labelled and given units by label and unit, e.g. for 2 components, temperature and pressure:

```
label=temperature pressure
unit=C Pa
```

3.2.3 Specifying the Format of the Data

Once the data structure has been defined, it is necessary to specify the location and format of the datafiles to be read.

- 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>`
- Although each new variable and each new coordinate begins on new lines, all format keywords for a particular item must be on the same line.
- Format keywords:
  - `file`: filename of the datafile to be read, (full pathname required).
  - `filetype`: format of the data, binary or ascii.
  - `skip`: number of lines or bytes in the data file before the data is actually found.
  - `offset`: number of columns to jump before the data is read (ASCII only).
  - `stride`: how many steps to jump before the next item is reached.

3.2.4 Summary

Field file format keywords:

- `ndim`: number of dimensions in computational space
- `dim1`: size of dimension 1
- `dim2`: size of dimension 2
- `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
- `variable n file` =??? `filetype` =??? `skip` =??? `offset` =??? `stride` =???
- `coord n file` =??? `filetype` =??? `skip` =??? `offset` =??? `stride` =???

Where

- `file`: location and data file
- `filetype`: format of datafile
- `skip`: number of lines (ASCII) or bytes (binary) before the data is found
- `offset`: number of columns to jump before data is read (ASCII only)
- `stride`: number of steps forward to reach next item

3.3 Exercises

Write field files to describe the datafiles[2] described in Exercises 1, 2, and 3. The solutions are given in Appendix B.

3.3.1 Exercise 1

The dataset called `image.dat` is a 2D image (512x256) consisting of greyscale values in the range 0-255,

```
10    0   15   200   255   ...
...
...
```

3.3.2 Exercise 2

The datafile called `volume.dat` consists of a 3D array of density components in a regular volume, 128x128x128. The file has one line of header information.

```
Patient:  Steve 10/11/93
0   255   2   1   7   ...
5   6    ... ... ...  ...
...
```

3.3.3 Exercise 3

The data is a 2D array (5x10) with floating point values of temperature and pressure at each point in space. The field is irregular. The temperature and pressure data is contained in the file `flow.dat` and the coordinates are contained in the file `coord.dat`. Neither file contains any header information.

```
flow.dat
   t1   p1   t2   p2   t3   p3   ...   ...   ...
coord.dat
   x1   y1   x2   y2   x3   y3   ...   ...   ...
```

[2] No datafiles exist for this Exercise – it is theoretical

3.4 Visualization of 2D Data

Now that creation of field files has been covered it is necessary to progress to the creation of applications which are able to process the data described within these field files. The exercises in the rest of the chapter demonstrate several simple examples of the visualization of 2D data.

3.5 Moving to the Home Directory

In this chapter, it is necessary to read the field files which should have been saved in specific sub-directories within the home directory. By default when a version of the `Read_Field` or other data reading module is instanced, it will point to the directory `C:\express\data`. Move to your home directory within the browser by navigating through the directory and sub-directory folders.

3.5.1 Exercise 1

The aim of this exercise is to visualize data that was obtained from an MRI scan of a brain.

3.5.2 Importing the `mri.dat` Data (1)

Below is a description of the data contained in the datafile `mri.dat`.

The dataset is 2D and contains `512x512` single scalar bytes for each data value. The data file is binary and the data is arranged contiguously with no header information.

- Create a file called `mri.fld` to describe the data contained in `mri.dat`.
- Use the template field file given on page 24 and the information given above to complete the specification.

3.5.3 Displaying the MRI Data

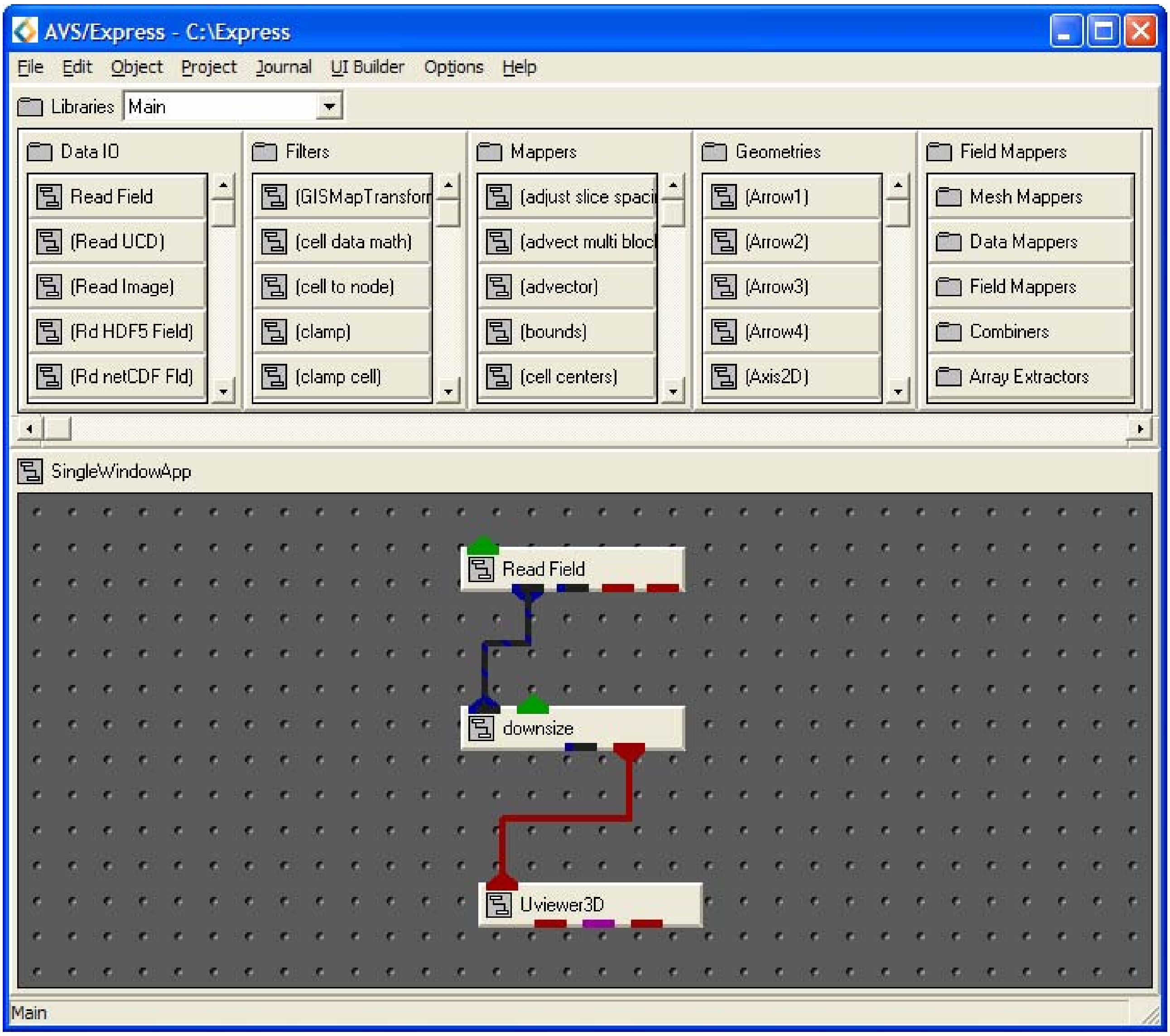

Figure 3.1: Network for `mri.fld`

- Using the Network Editor, instance the modules in the list below into the workspace and construct the network shown in Figure 3.1 (be careful to connect the correct ports).
  - `Read_Field` (Data IO): reads an AVS field file and converts it into an AVS/Express field.
  - `downsize` (Filters): resample a field using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
  - `Uviewer2D` (Viewers): creates a 2D image in the DataViewer.

3.5.4 Importing the `mri.dat` Data (2)

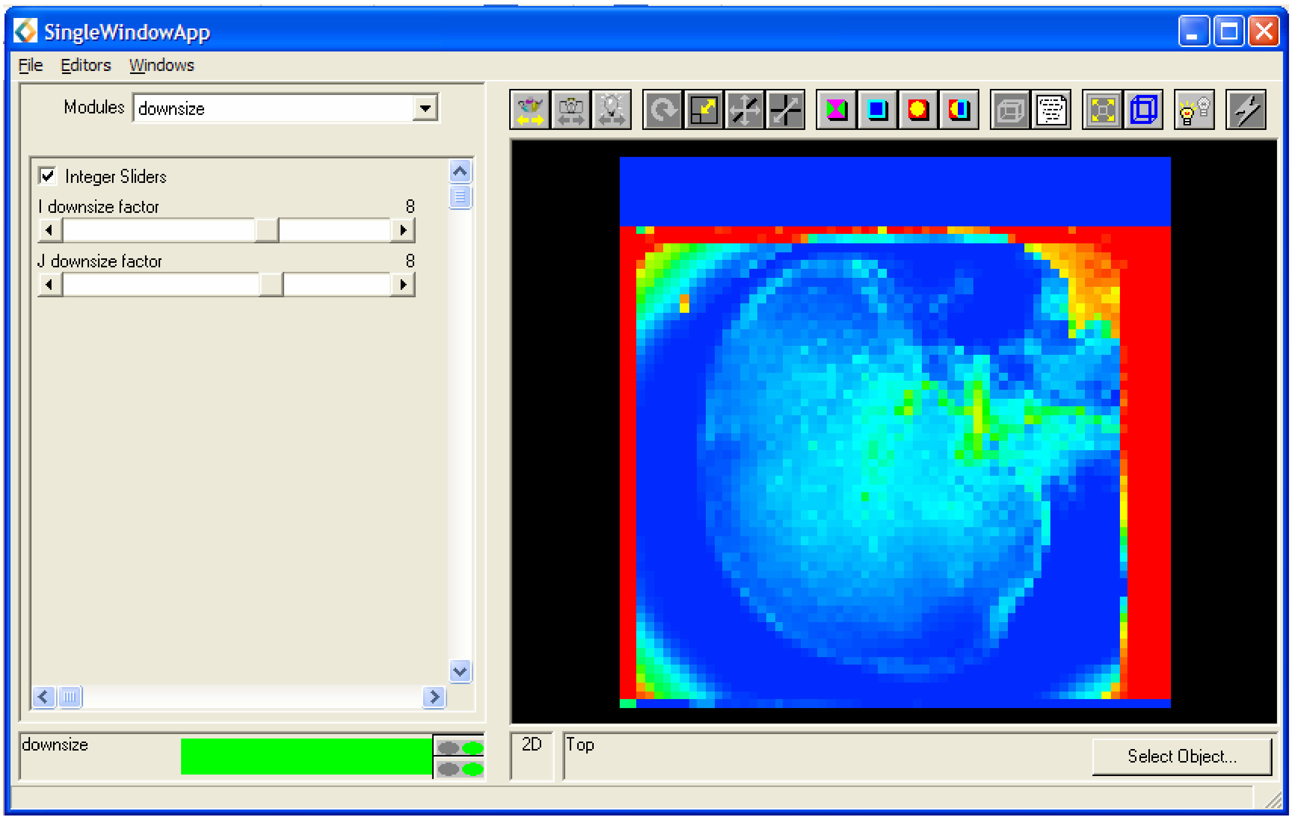

Figure 3.2: `mri.fld` in the Data Viewer Pad

- In the **DataViewer** window select **Modules** from the **Editors** pull-down menu and select the `Read_Field` module.
- To import the data into the network use the file **browser** to move to the correct directory and select the file `mri.fld` previously created.
- An image similar to the one shown in Figure 3.2 should be obtained.

The image displayed has been produced using a subset of data, due to the `downsize` module. The image is likely to be poor.

3.5.5 Changing the Downsizing

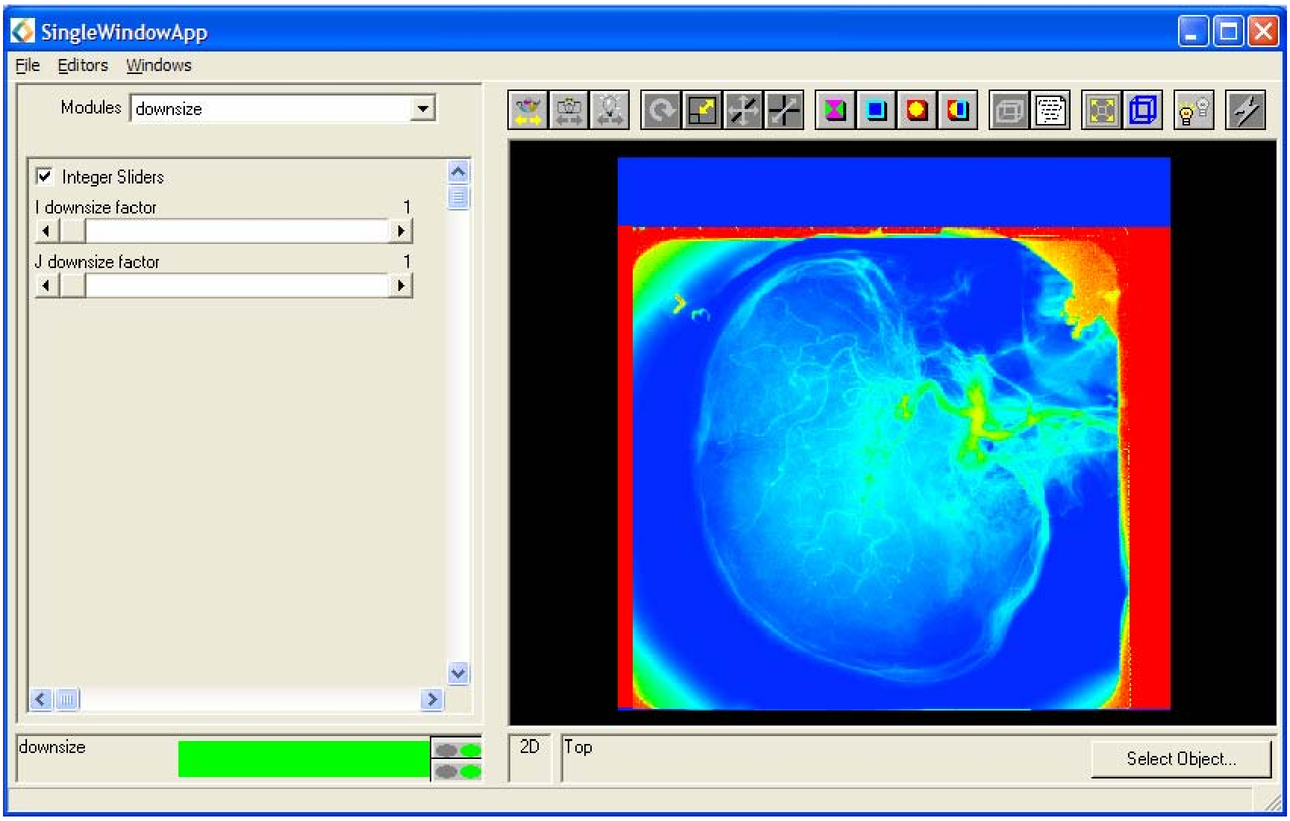

Figure 3.3: `downsize` Editor Panel

- Select the `downsize` module from the **Modules** drop down list.
- Alter the **I** and **J** downsize factor. (The best resolution of the image is obtained it both values are set to **1**.)

3.5.6 Changing the Colour Map (1)

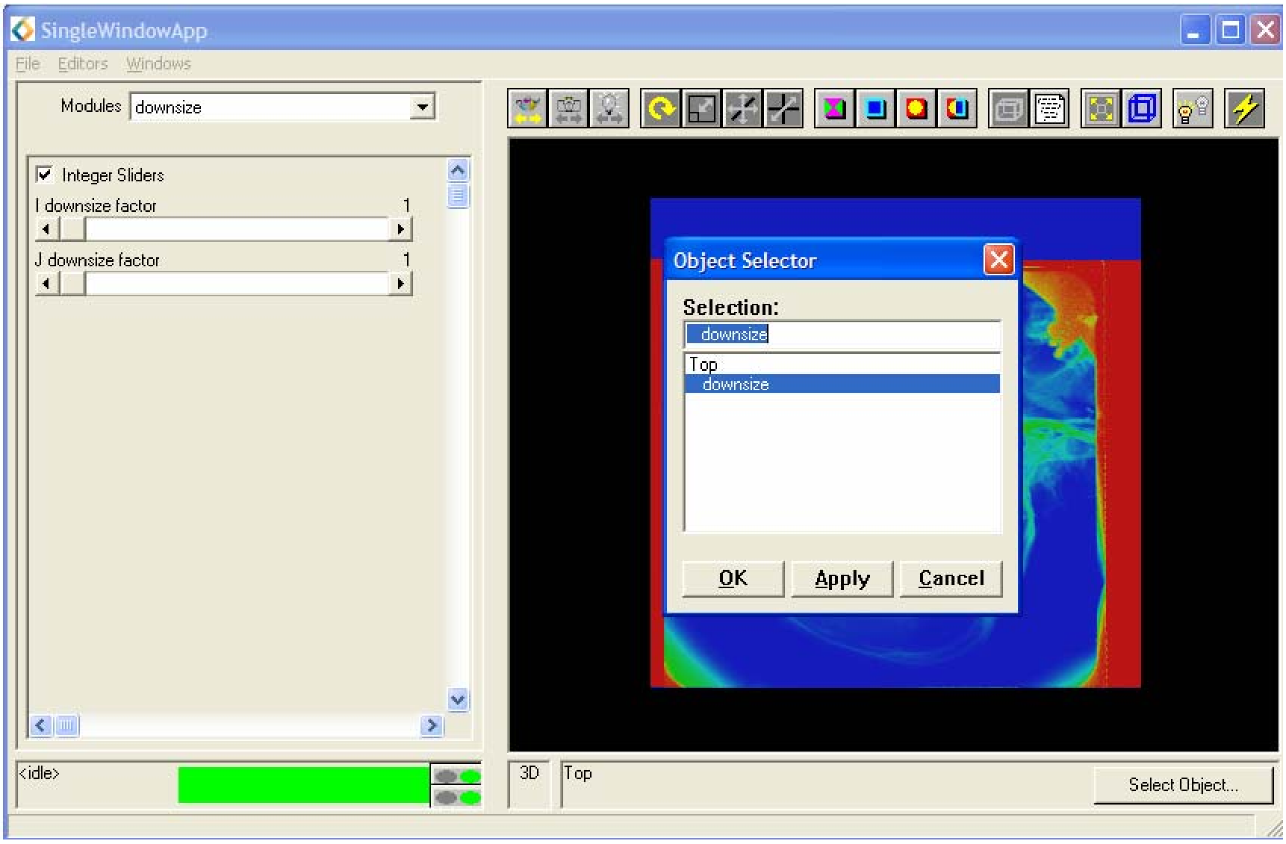

Figure 3.4: **Object Selector** Pop up Window

The image is coloured according to the Default AVS/Express **Datamap**. It is possible to alter the colour map of the image in order to create more useful and informative visualizations.

In AVS/Express, each object in the image hierarchy can be selected and manipulated separately. *In order to change the datamap, the image itself must be selected as the current object.*

- Click the **Select Object**… button, which is found in the bottom right corner of the DataViewer.
- Select the object which in this case is **downsize**, in the object selector pop up window and click **Apply** or **OK**.

3.5.7 Changing the Colour Map (2)

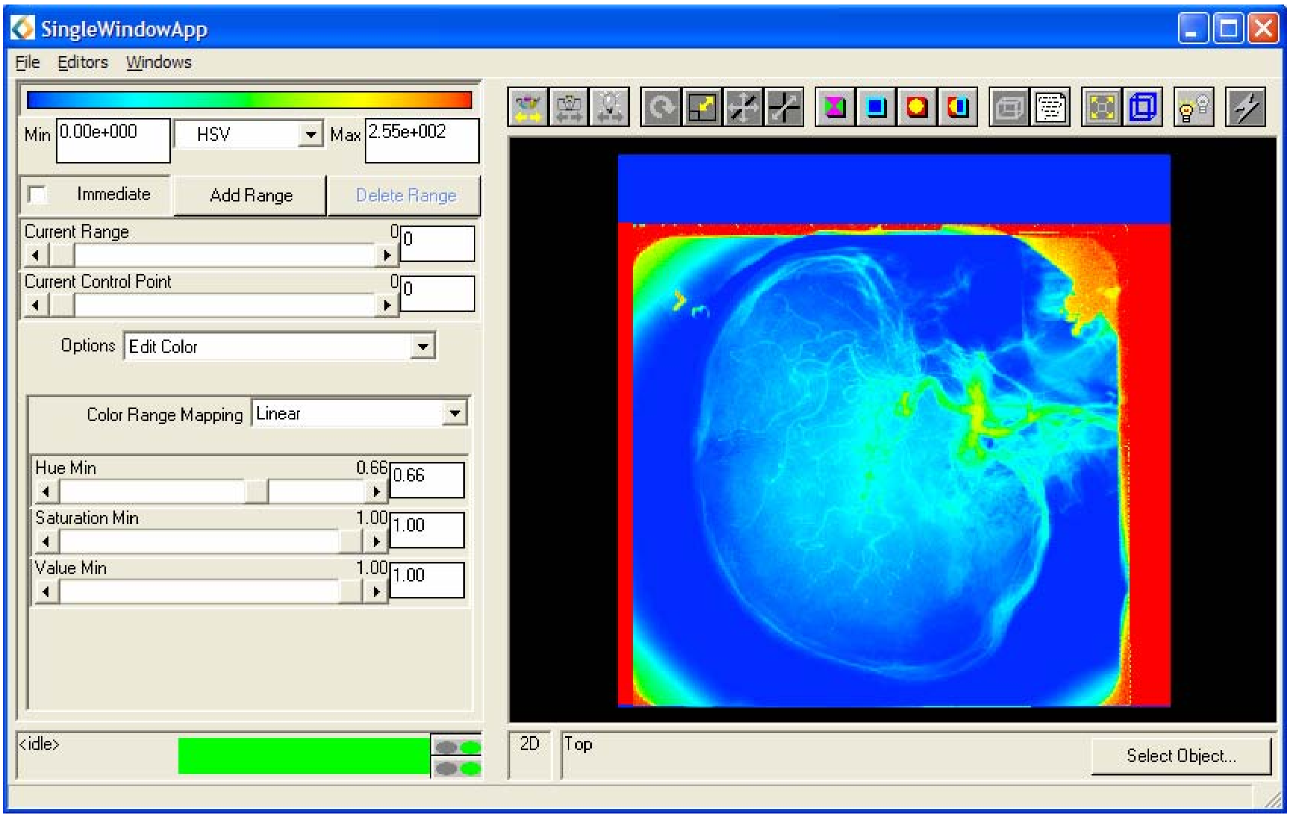

Figure 3.5: **Datamap** Editor

- Select the **Editors** pull down menu and select **Datamap**. The Datamap editor should appear in the Data Viewer pad as shown in Figure 3.5.

Details regarding the main properties of the Datamap editor are listed below:

- There are two colour models which may be used:
  - **HSV** (**H**ue **S**aturation **V**alue)
  - **RGB** (**R**ed **G**reen **B**lue)
- The **Colour Range Mapping** can be either:
  - **linear** - interpolated between max and min values of data
  - **Stepped** - constant colour for each step).
  - The mapping is set by **Control Points**.
- The datamap range and the data range can be altered or additional sub-ranges added to give a more precise representation of the data.

3.5.8 Changing the Colour Map (3)

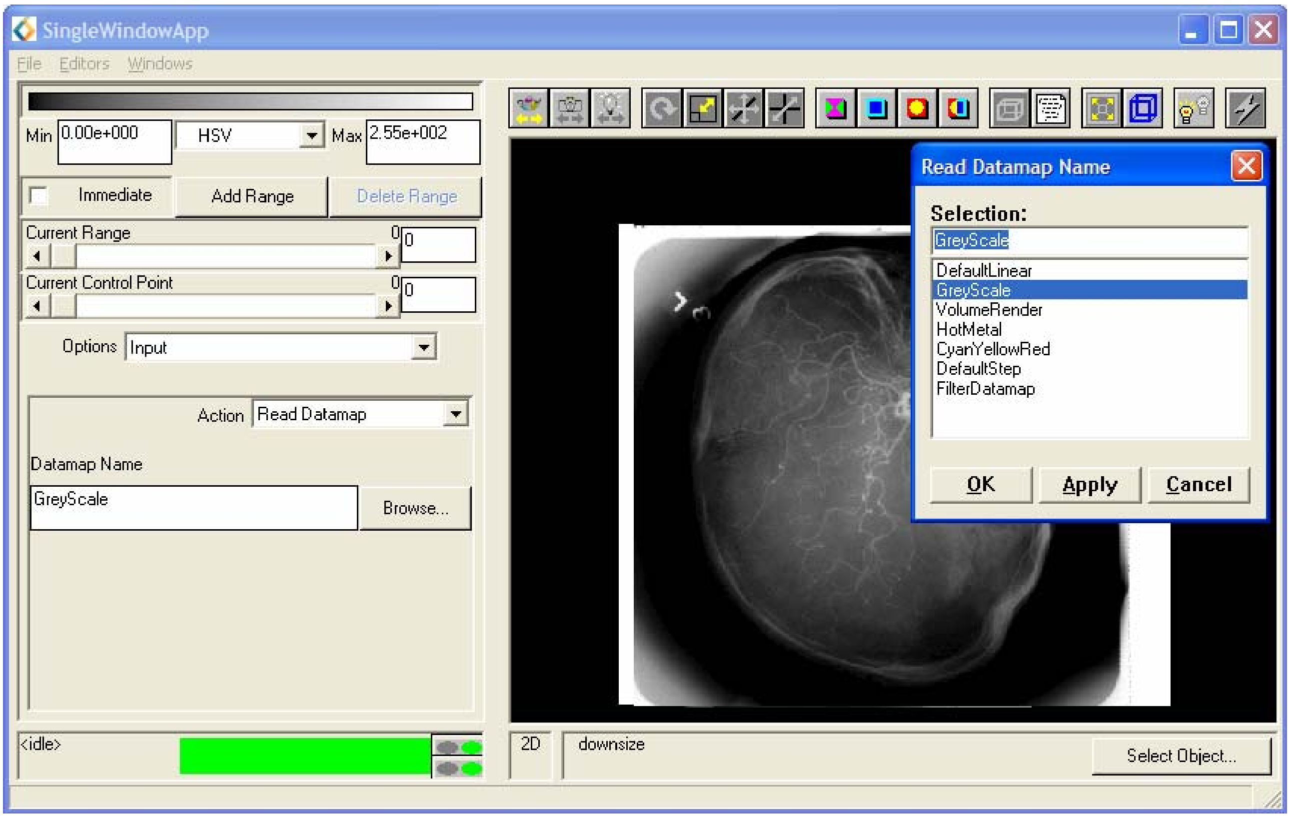

Figure 3.6: Greyscale Datamap

There are several pre-defined datamaps which can be used to view the image. The way in which to alter the datamap in order to obtain a greyscale image is as follows.

- In the **Options** menu select **Input**.
- In the **Action** menu select **Read Datamap**.
- Select the **Browse** button.
- From the **Read Datamap Name** pop up window select **Greyscale** and click **Apply**.
- The image in the DataViewer should now be a greyscale image, with black set to 0 and white set to 255.

3.5.9 Changing the Colour Map (4)

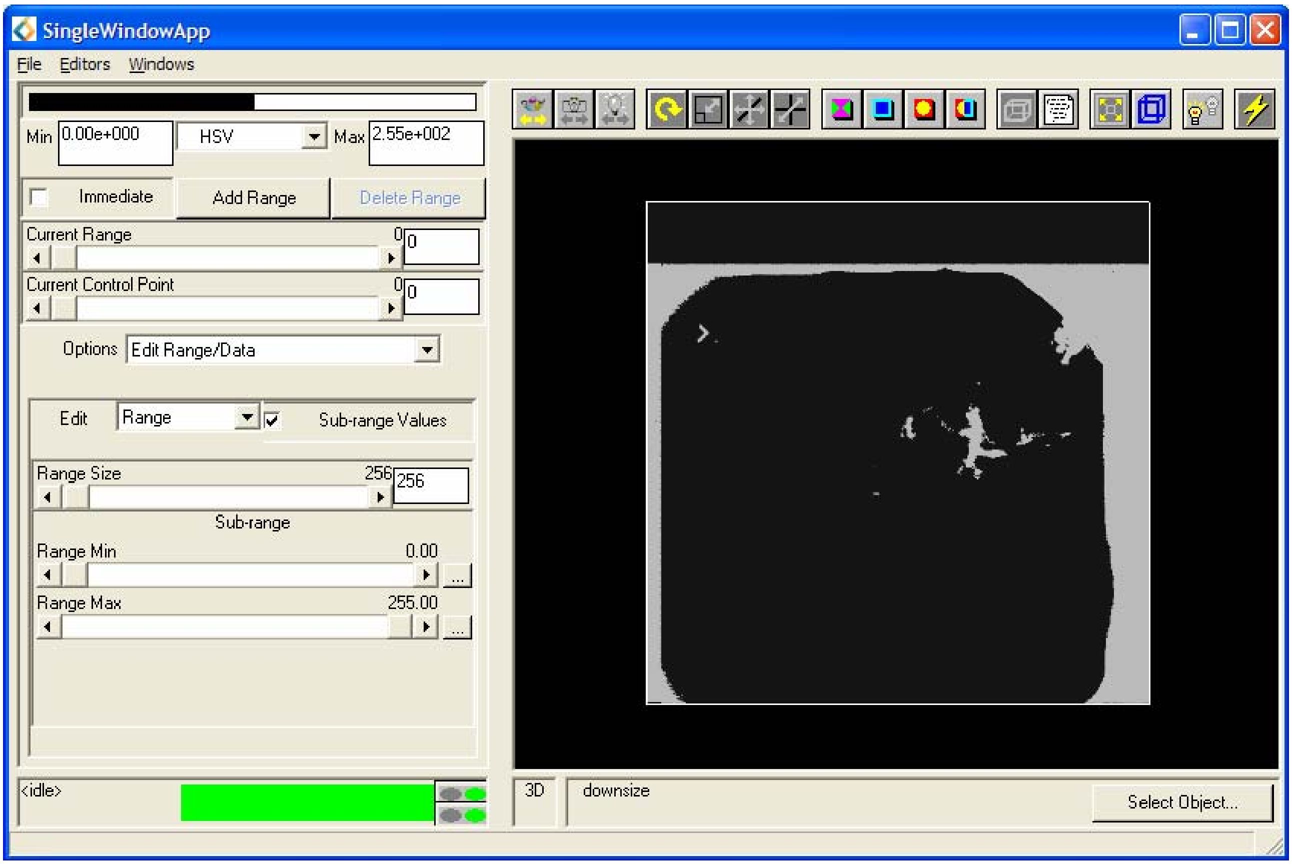

Figure 3.7: Binary Black and White Datamap

By further altering the datamap it is possible to transform the greyscale image into a binary black and white image.

- From the **Options** menu select **Edit Colour.**
- Change the **Colour Mapping Range** from **Linear** to **Constant**. A binary map should be obtained as shown in Figure 3.7.
- Select **Edit Range/Data** from the **Options** menu.
- Select the **Sub-range Values** and alter the values of max and min in order to alter the position of the black/white division.
- Try out some of the other datamap parameters.

3.6 Exercise 2

The aim of this exercise is to represent the temperatures taken from a cross section of an injection moulding system.

3.6.1 Importing the `temps.dat` Data (1)

Below is a description of the data file contained in the datafile `temps.dat`:

The ASCII data file contains data on a regular, `31x31` grid. The data component at each element is a single floating point value and the first two numbers in the file indicate the x and y dimensions of the grid respectively.

- Create a file called `temps.fld` to describe the data contained in `temps.dat` (rather than re-type the field format edit the existing `mri.fld` and save it under a new filename).
- Use the template field file given on page 24 and the information given above to complete the specification.

3.6.2 Displaying the Temperature Data with `surf_plot`

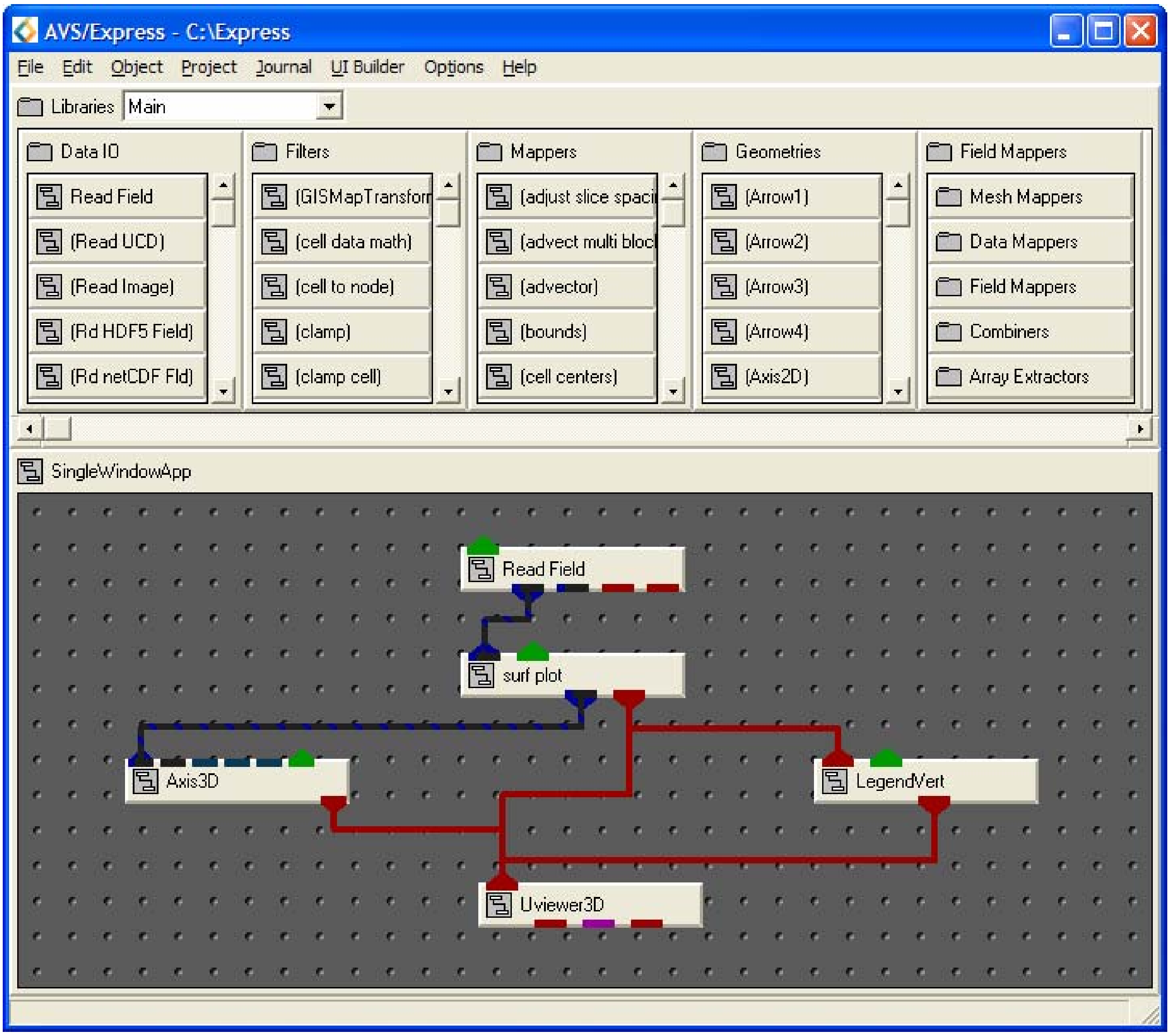

Figure 3.8: Network containing `surf_plot` for `temps.fld`

Using the Network Editor construct the network shown in Figure 3.8. Descriptions of the modules are given below:

- `Read_Field` (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- `LegendVert` (Geometries): produce a colour legend which displays the objects datamap.
- `surf_plot` (Mappers): creates a 2D or 3D mesh whose height is proportional to the scalar data values at each point. 2D data produces a 3D mesh and 1D data produces a 2D mesh.
- `UViewer3D` (Viewers): produces a 3D image in the DataViewer.

3.6.3 Importing the `temps.dat` Data (2)

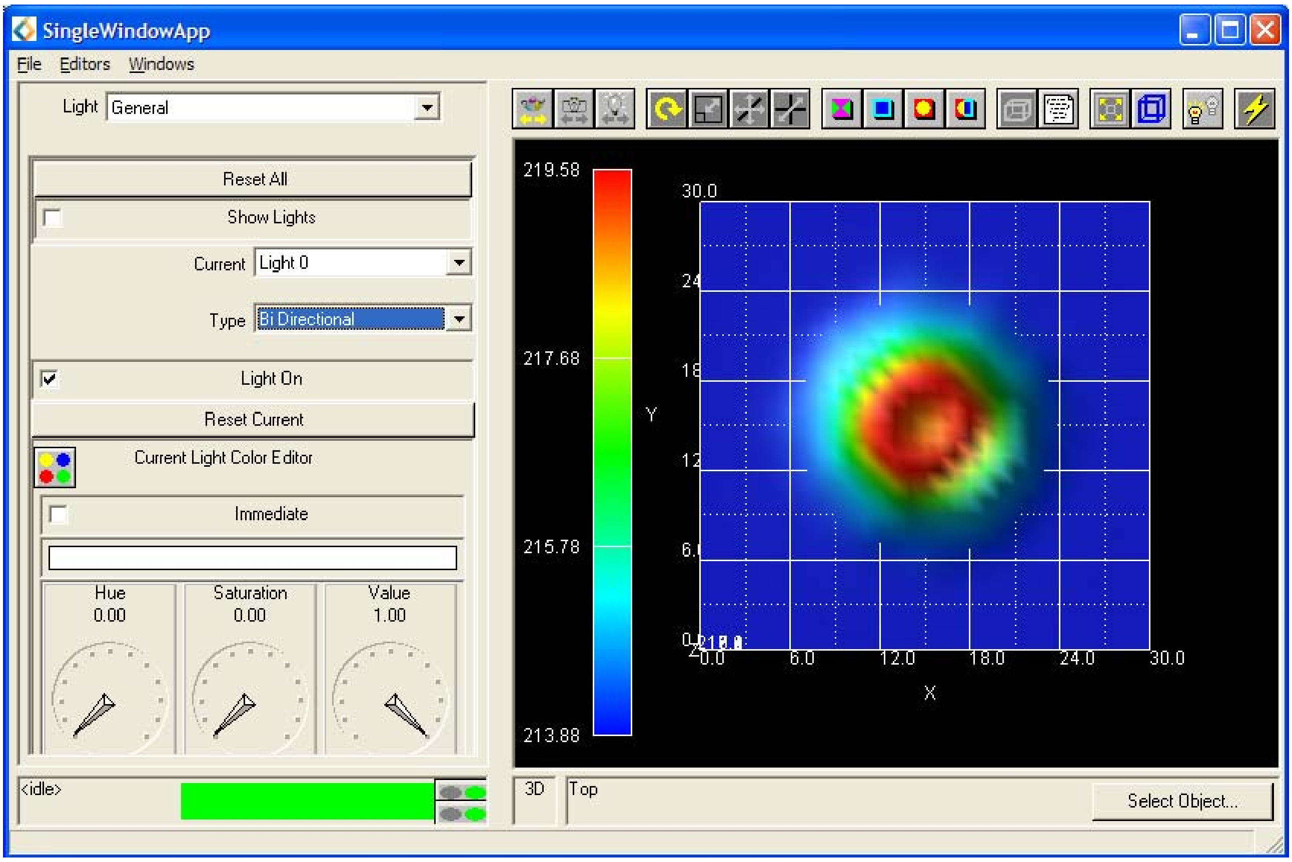

Figure 3.9: **Light Editor** and `temps.fld` Visualization

- Select the `Read_Field` module from the **Modules** drop down list and select the **Browse** button. The file browser should appear.
- To import the data into the network, select the file `temps.fld` using the file browser.
- If an image similar to the one shown in Figure 3.9 appears then the field file has been created successfully.
- It often improves the appearance of the image upon rotation to change the light to bi-directional:
  - In the **Editor**s menu select **Light**.
  - In the **Type** menu select **BiDirectional**.

3.6.4 Normalizing the Image

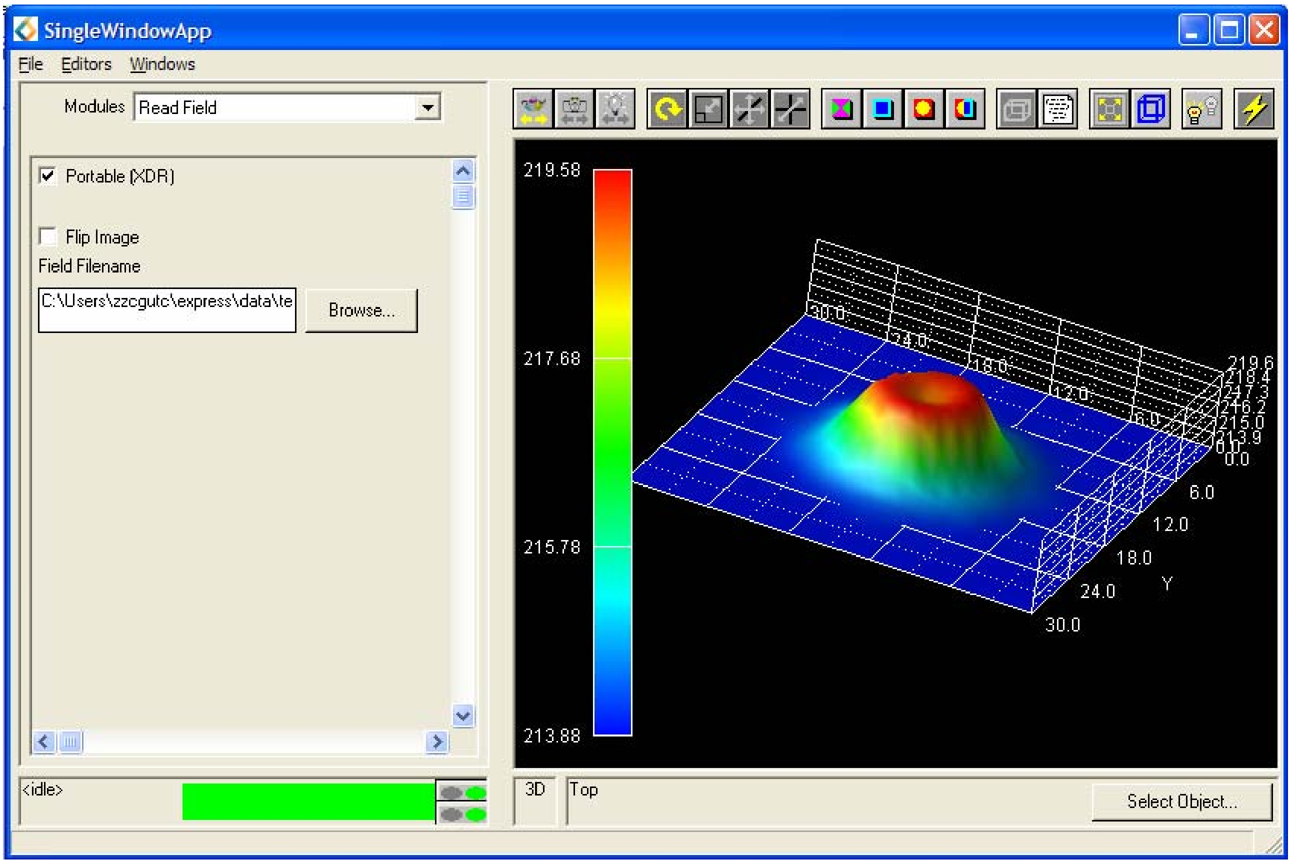

Figure 3.10: Transformed Image in the Data Viewer Pad

- Select the **Reset/Normalize/Center** icon on the Toolbar so that the whole object fits into the window.
- Using the mouse, transform the object so it appears similar to that of Figure 3.10.

3.6.5 Altering `LegendVert` and `Axes3D`

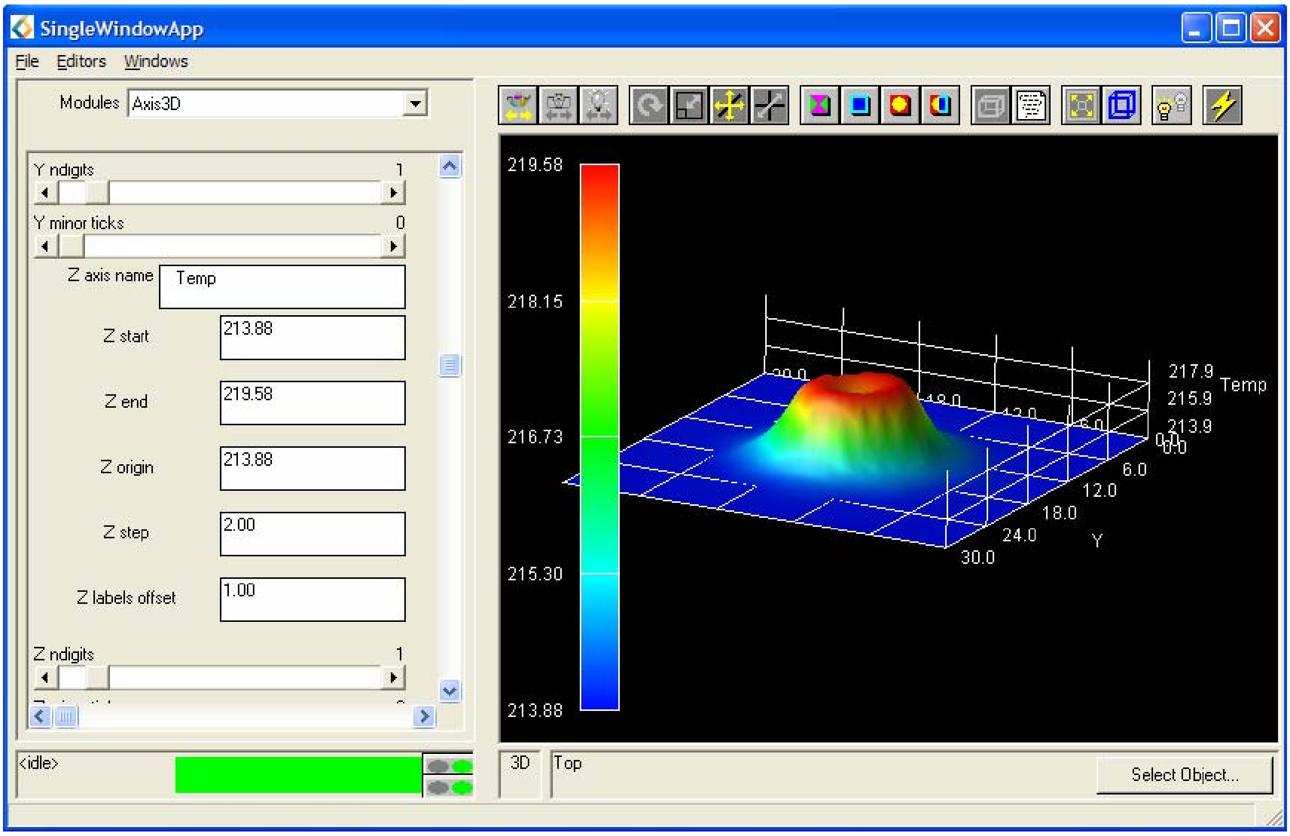

Figure 3.11: `Axes3D` Editor Panel

- Select the `LegendVert` module from the **Modules** drop down list.
- Change the parameters of the legend.
  - o Decrease the Intervals
- Similarly, select the `Axes3D` module from the **Modules** drop down list:
  - o Increase the number of steps on the z-axis.
  - o Label the z-axis "Temp".
  - o Remove the minor ticks

3.6.6 Using `surf_plot`

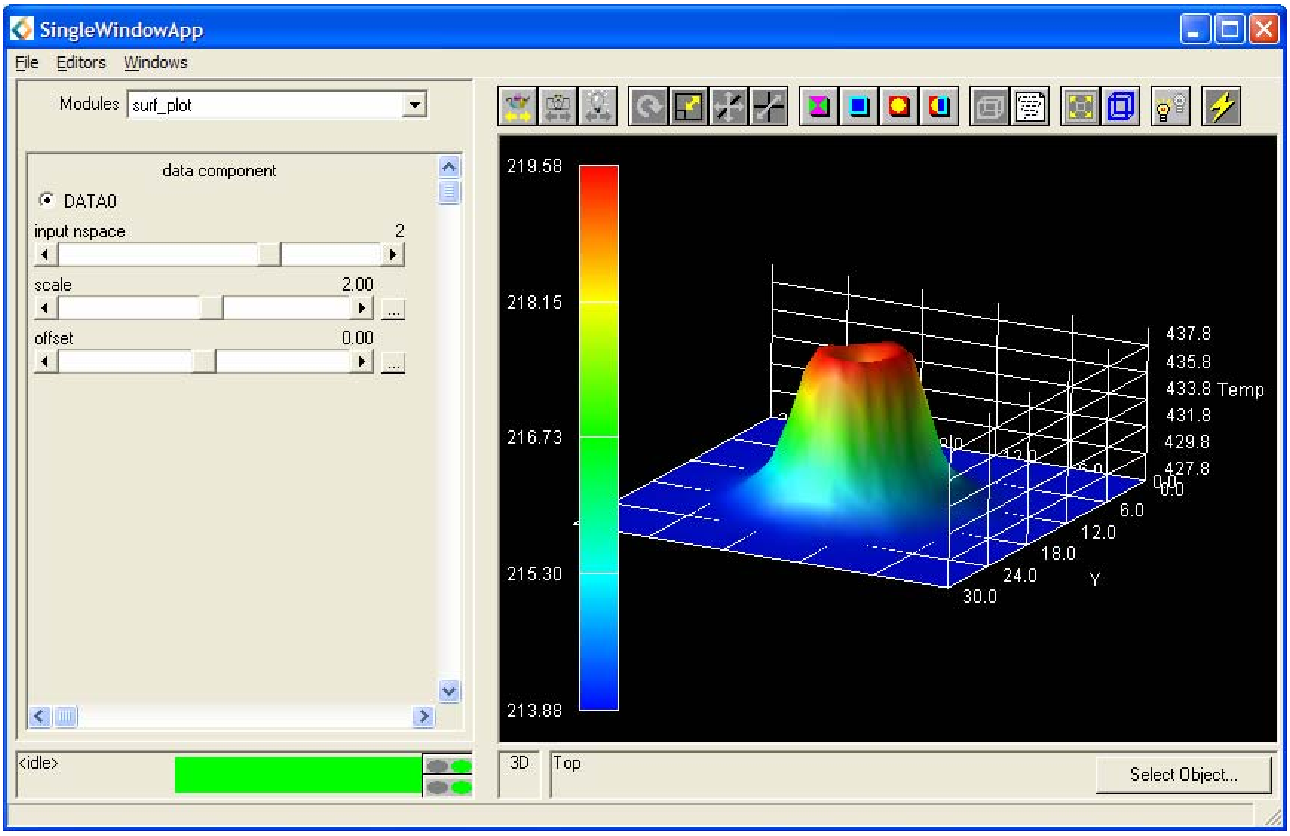

Figure 3.12: `surf_plot` Editor Panel

- Select the `surf_plot` module from the **Modules** drop down list.
- Alter the **scale** slider to increase the scaling.
- The module should now provide an image similar to the one shown in Figure 3.12.

3.6.7 Enhancing the Image

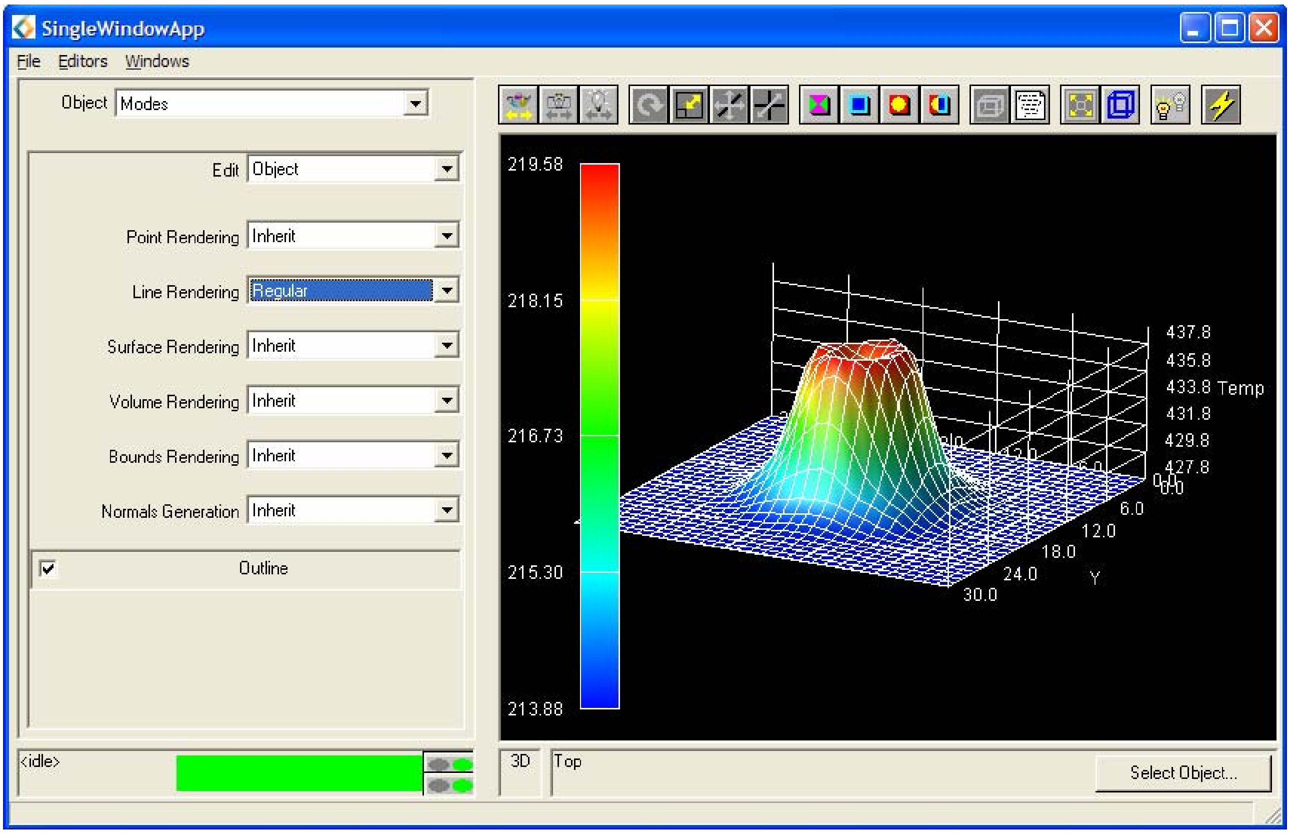

Figure 3.13: **Object Editor**

- Select **Object** from the **Editors** pull-down menu to invoke the **Object editor**.
- Select **Modes** from the **Object** drop down menu.
- Set the mode of **Line Rendering** to **Regular**. This will shade the image and superimpose a wireframe structure onto the `surf_plot` image.

3.6.8 Displaying the Temperature Data with `city_plot`

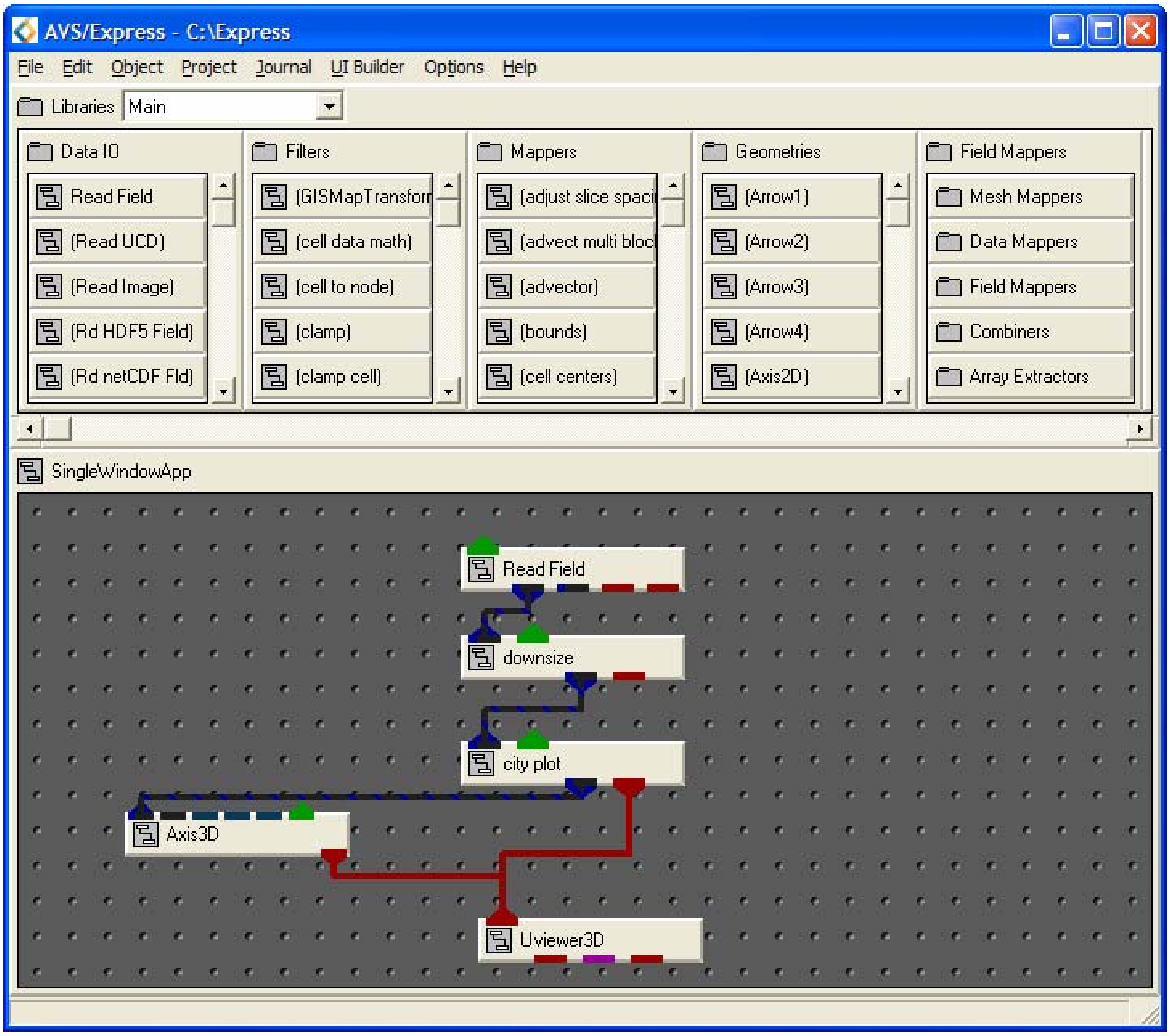

Figure 3.14: Network containing `city_plot`

Using the Network Editor construct the network shown in Figure 3.14. The location of the modules as listed below:

- `Read_Field` (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- `downsize` (Filters): resamples a field using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
- `city_plot` (Mappers): creates a plot of blocks whose height is based on the value of the mesh extents.
- `Uviewer3D` (Viewers): creates a 3D image in the Data Viewer.

3.6.9 Importing the `temps.dat` Data

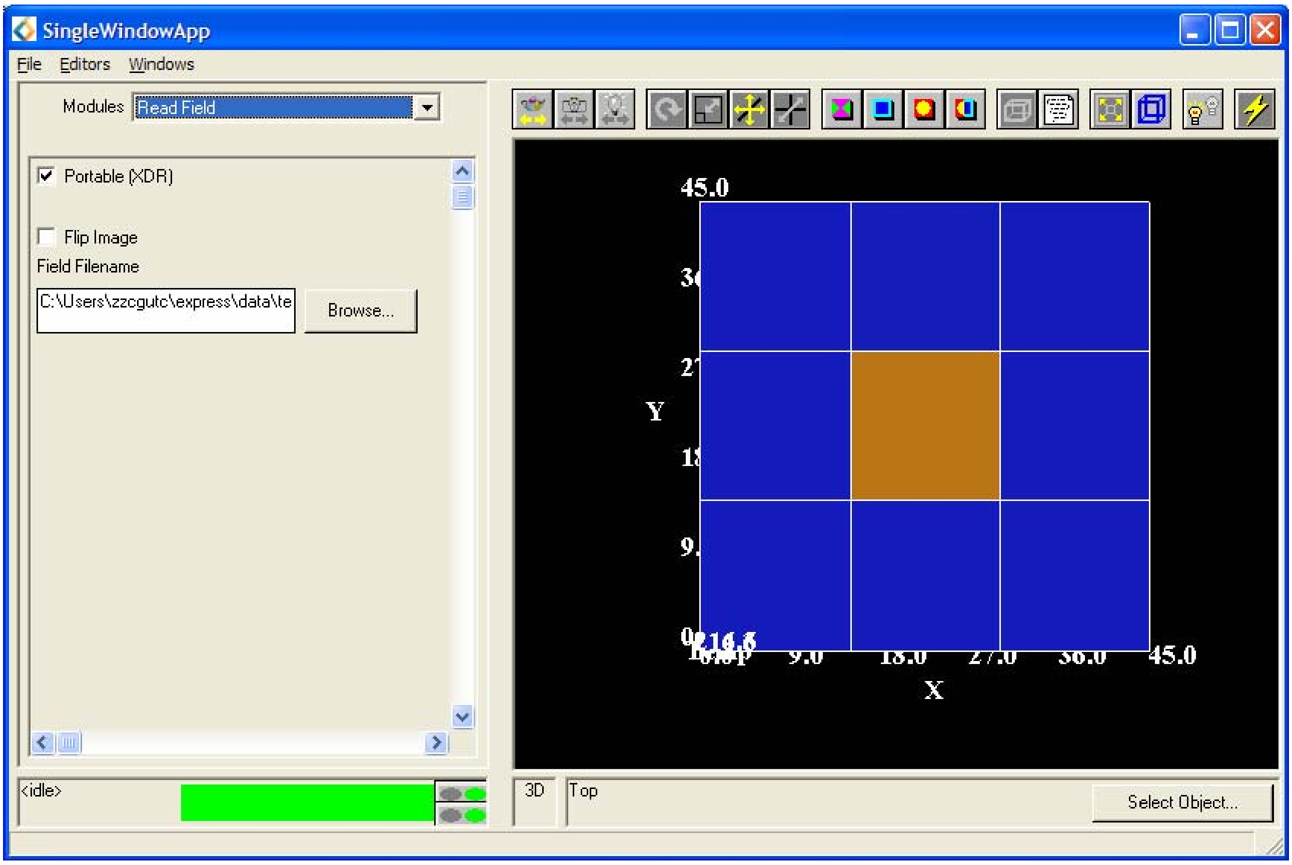

Figure 3.15: Image produced by the `city_plot` Network

- Select the `Read_Field` module from the **Modules** drop down list and select the **Browse** button. The file browser should be shown.
- To import the data into the network, select the file `temps.fld` using the file browser.
- If an image similar to the one shown in Figure 3.15 then the field file has been successful.

3.6.10 Normalizing the Image

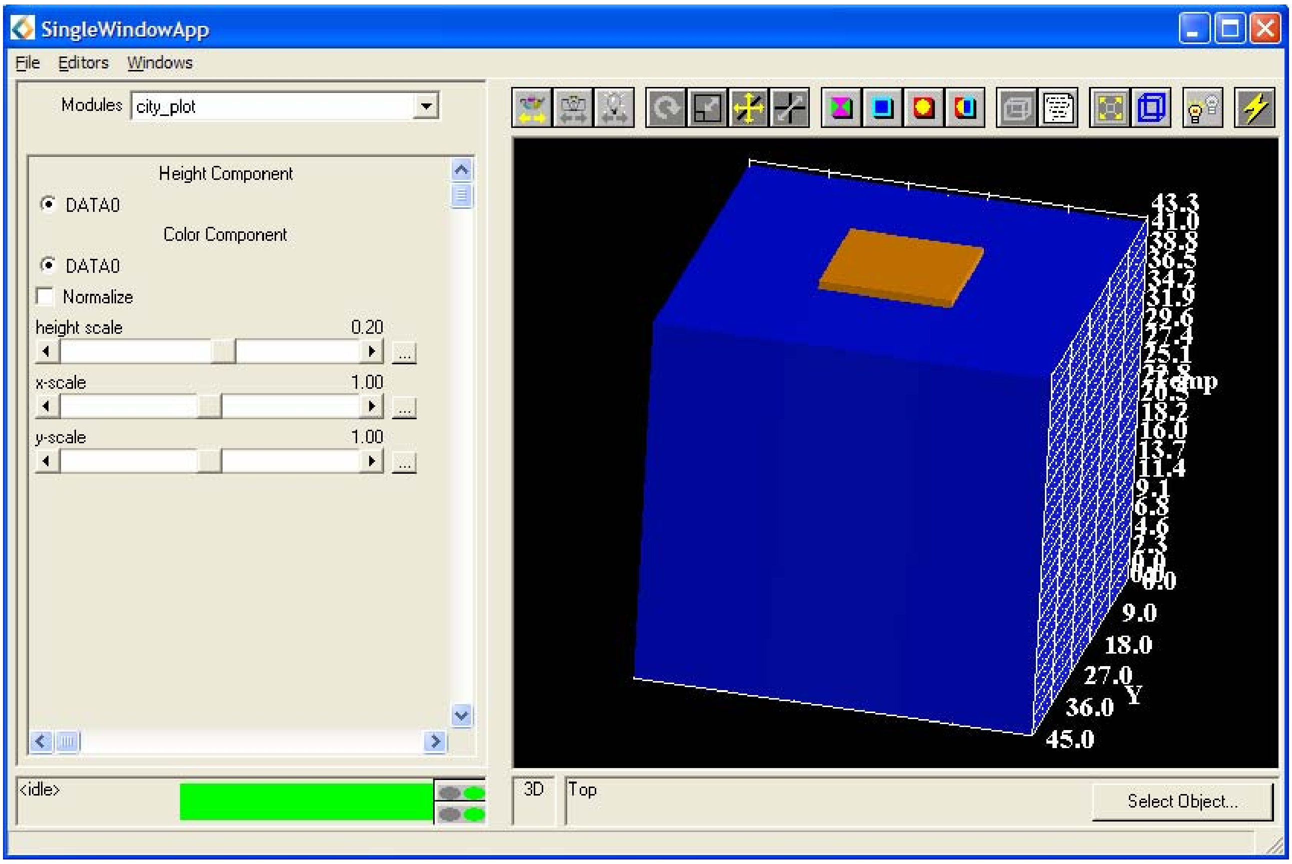

Figure 3.16: `city_plot` Editor Panel and Transformed Image

- Click the **Reset/Normalize/Center** icon in the Toolbar so the whole object is shown in the window.
- Select the appropriate icons from the Toolbar and use the mouse to transform the object so it appears as shown in Figure 3.16.
- To reduce the bar chart's length select the `city_plot` module from the **Modules** drop down list. Change the height scale value on the corresponding slider.
- Alter some of the scale values (as shown in Figure 3.16) using the sliders and observe the effects.

3.6.11 Changing the Downsizing

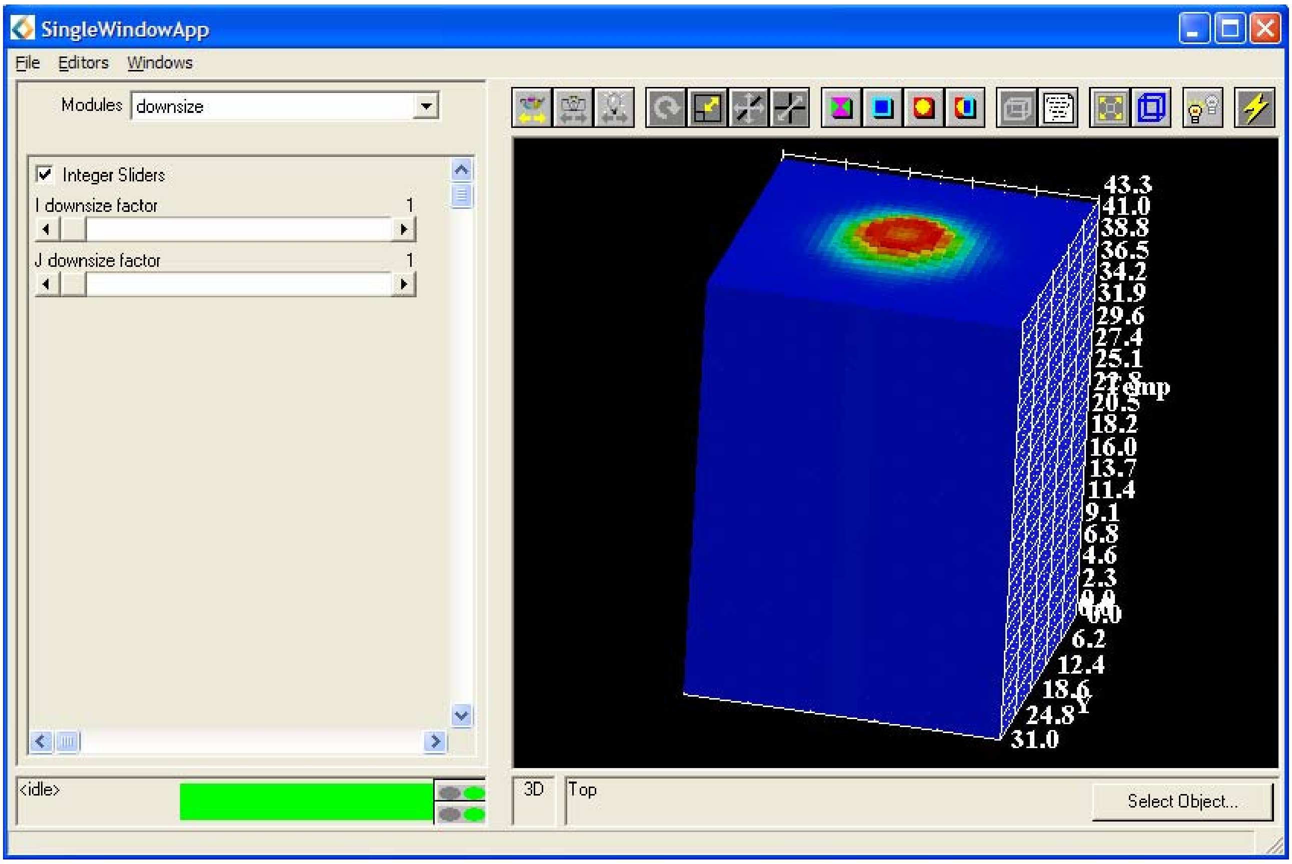

Figure 3.17: `downsize` Editor Panel

The image shown in Figure 3.17 has been produced using a subset of the data, due to the `downsize` module.

- Select `downsize` from the **Modules** Editor Panel.
- Alter the values of the **I** and **J** downsize factors. (Again the best resolution of the image will be obtained if both of the values are set to **1**.)
- Locate and select the **Perspective** toggle icon on the Toolbar to observe its effect.

3.7 Exercise 3

3.7.1 Importing the `hslice.dat` data

Below is a description of the data contained in the datafile `hslice.dat`:

This data file represents a single slice through a hydrogen atom where the data values represent the electron density. The dataset is 2D and contains `64x64` single scalar bytes for each data value. The data file is binary and the data is arranged contiguously with no header information.

- Create a file called `hslice.fld` to describe the data contained in `hslice.dat` (rather than re-type the field format edit an existing field file and save it under a new filename).
- Use the template field file given on page 24 and the information given above to complete the specification.
- Construct a network as in Exercise 2 using `surf_plot`
- Select the `Read_Field` module from the **Modules** drop down list and select the **Browse** button. The file browser should now be shown.
- Import the data into the network and investigate the possible alterations to the image, given in Exercise 2.

4 Visualizing 3D arrays of Data

4.1 Aims of this Chapter

The aim of this chapter is to demonstrate techniques which are used to visualize 3D data. It consists of several structured exercises which illustrate the visualization process:

- Creating field files.
- Constructing applications.
- Manipulating data in 3D.
- The solutions to the field files for the 3D data are given in Appendix B.

4.2 Exercise 1

The aim of this exercise is to visualize a downsized version of the AVS/Express sample dataset `fin.fld` which shows the flow of air over a blunt fin.

4.2.1 Importing the `fin.dat` Data

Below is a description of the data contained in the datafile `fin.dat`

The data file is called fin.dat and is an ASCII file which represents a `10x8x8` irregular grid. Each data element has 5 floating point data components and each element has some associated coordinate data to position it in 3D space.

The data file format is as follows:

```
Density x-momentum y-momentum z-momentum stagnation x-coord y-coord z-coord
```

And a small extract from the file is shown below:

```
2.171   0   0   0   10.7602   0              0          0
2.1523  0   0   0   10.6423   0.0160473   0.129677   0
1.9742  0   0   0   9.61927   0.0757717   0.266559   0
1.6659  0   0   0   7.96548   0.173189    0.379764   0
1.2728  0   0   0   5.94705   0.299646    0.45922    0
0.86119 0   0   0   3.92577   0.443802    0.497866   0
0.7654  0   0   0   3.6417    0.618551    0.501378   0
0.73957 0   0   0   3.52176   1.08654     0.501378   0
0.55002 0   0   0   2.62795   2.47197     0.501378   0
0.45067 0   0   0   2.20615   6.57433     0.501378   0
```

- Create a file called `fin.fld` to describe the data contained in `fin.dat.`
- Use the template field file given on page 24 and the information given above to complete the specification.

4.2.2 Display the Vector Data

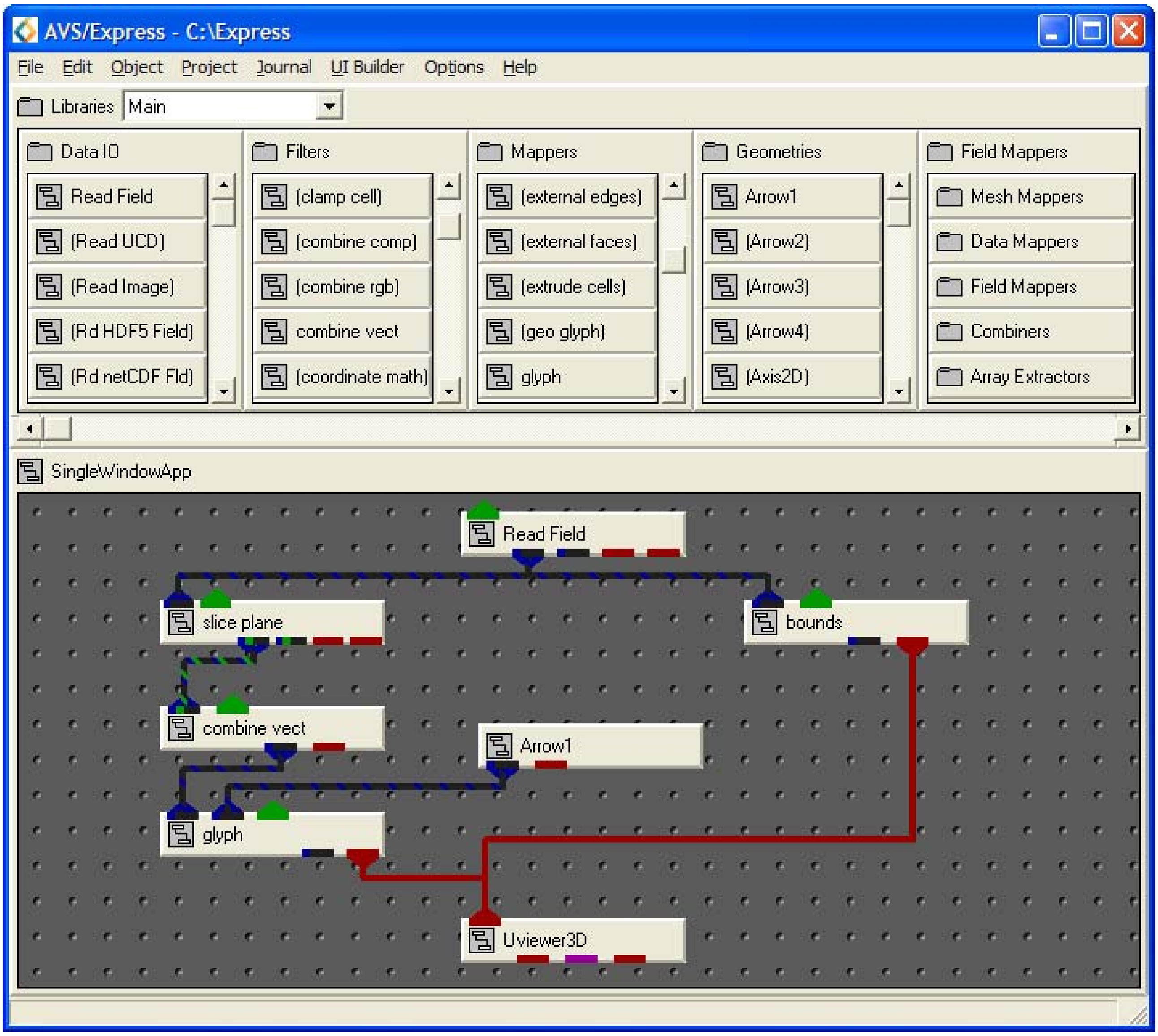

Figure 4.1: Network for Displaying `fin.dat` as Vector Data

Create the above network using the following modules:

- `Read_Field` (Data Input): reads an AVS field file and converts it into an AVS/Express field.
- `slice_plane` (Mappers): extract a 2D slice from a 3D field with an arbitrarily positioned slice plane.
- `combine_vect` (Filters): takes all the selected scalar data components and combines to output a single vector component.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `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.

4.2.3 Selecting the Map Components

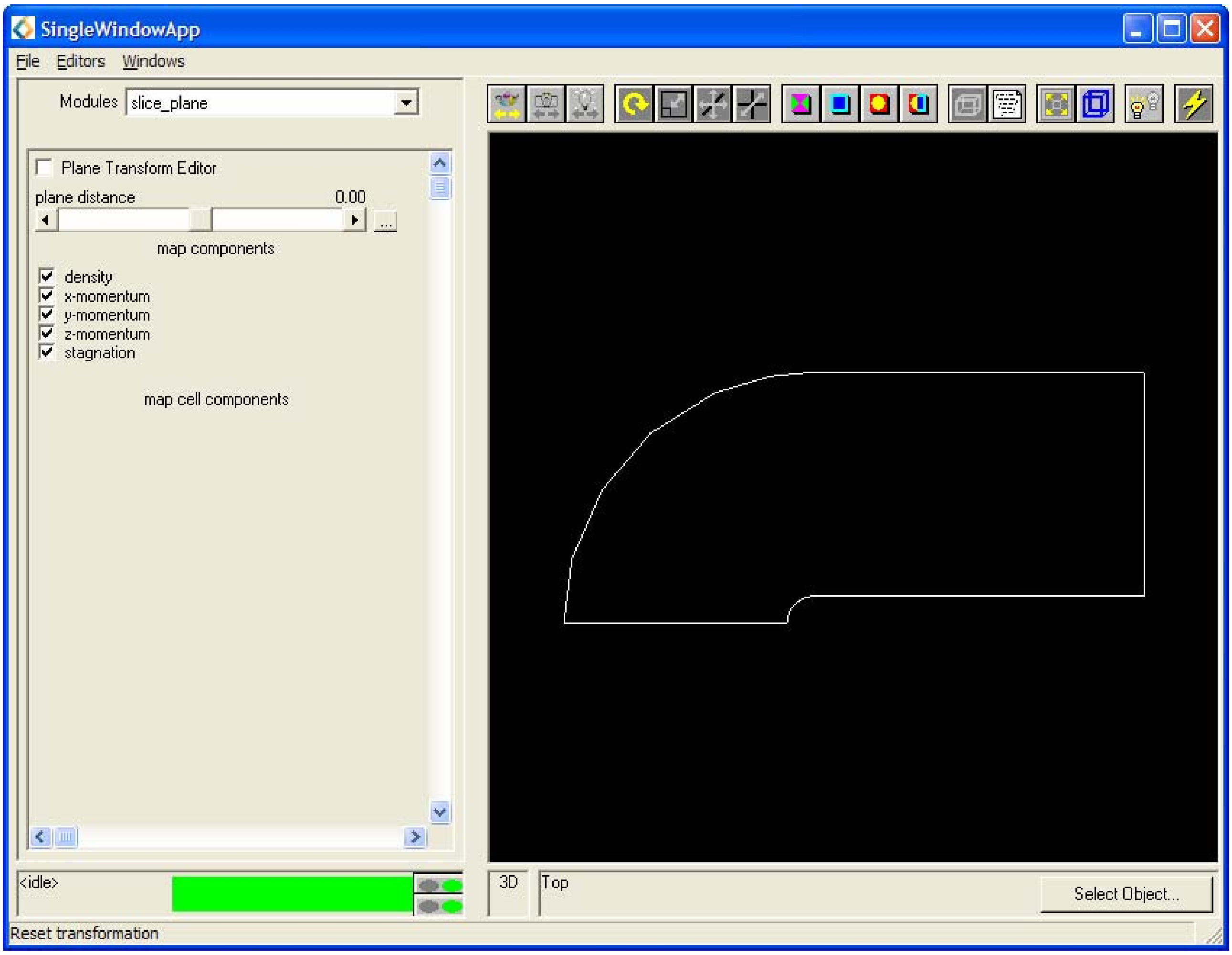

Figure 4.2: `slice_plane` Editor Panel

- Select the `Read_Field` module from the **Modules** drop down list and select the file `fin.fld` to import using the file browser.
- Select the Editor Panel for the `slice_plane` module and activate all five data components by selecting each one with the mouse as shown in Figure 4.2. Temporarily ignore any error messages which appear.

This module will now pass on a 3D field with a 5-vector data element consisting of density, x, y, and z momentum and stagnation to the other modules in the AVS/Express network.

4.2.4 Extracting the Vector Data

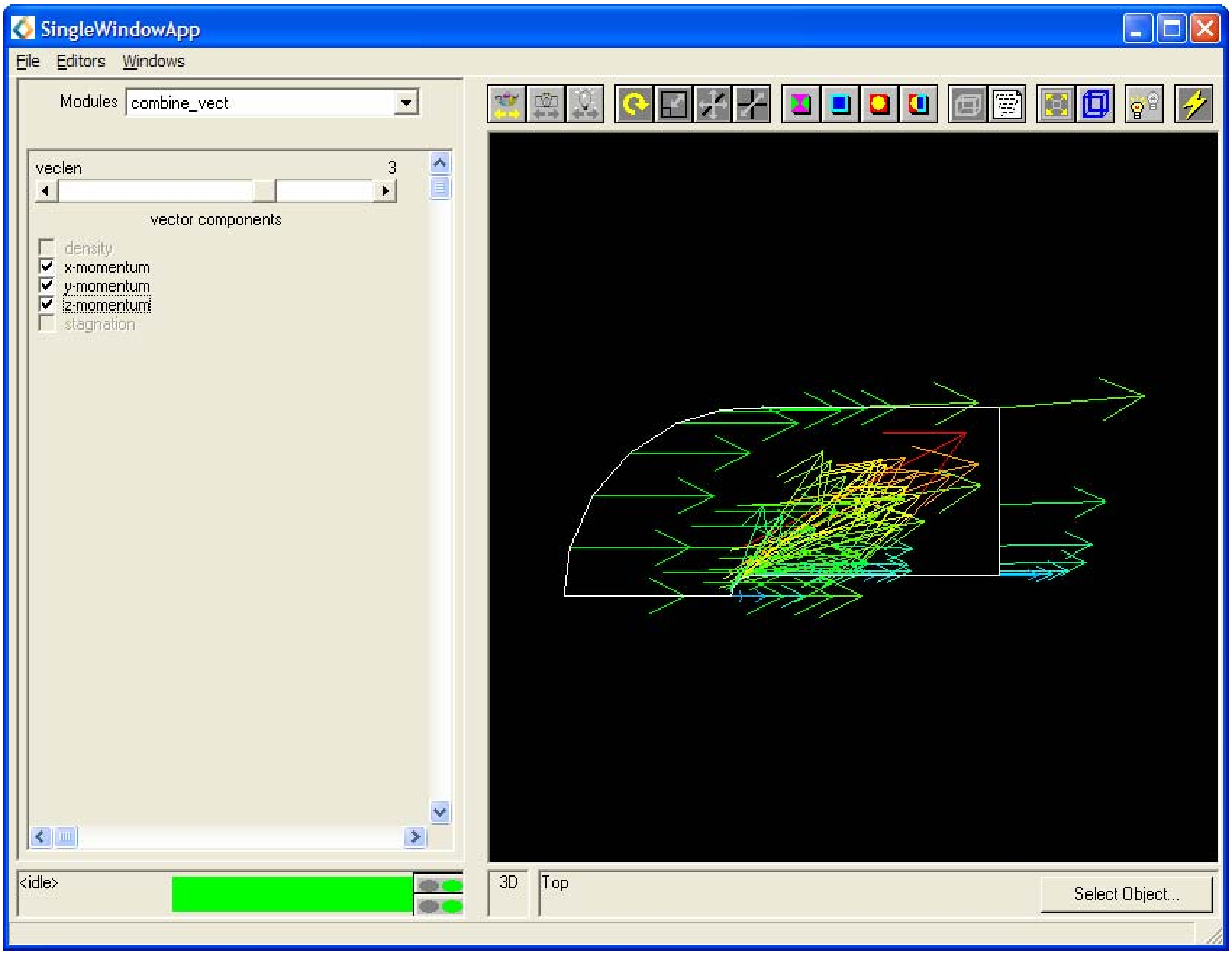

Figure 4.3: `combine_vect` Editor Panel

- Select the Editor Panel for the `combine_vect` module and activate the three data components **x-momentum**, **y-momentum** and **z-momentum** as shown in Figure 40.
- Since the **veclen** is set to **3** it is currently only possible to have 3 vector components active.

This module will now pass on a 3D field with a 3-vector data element consisting of x, y and z momentum to the other modules in the AVS/Express network.

4.2.5 Using the `glyph` Module

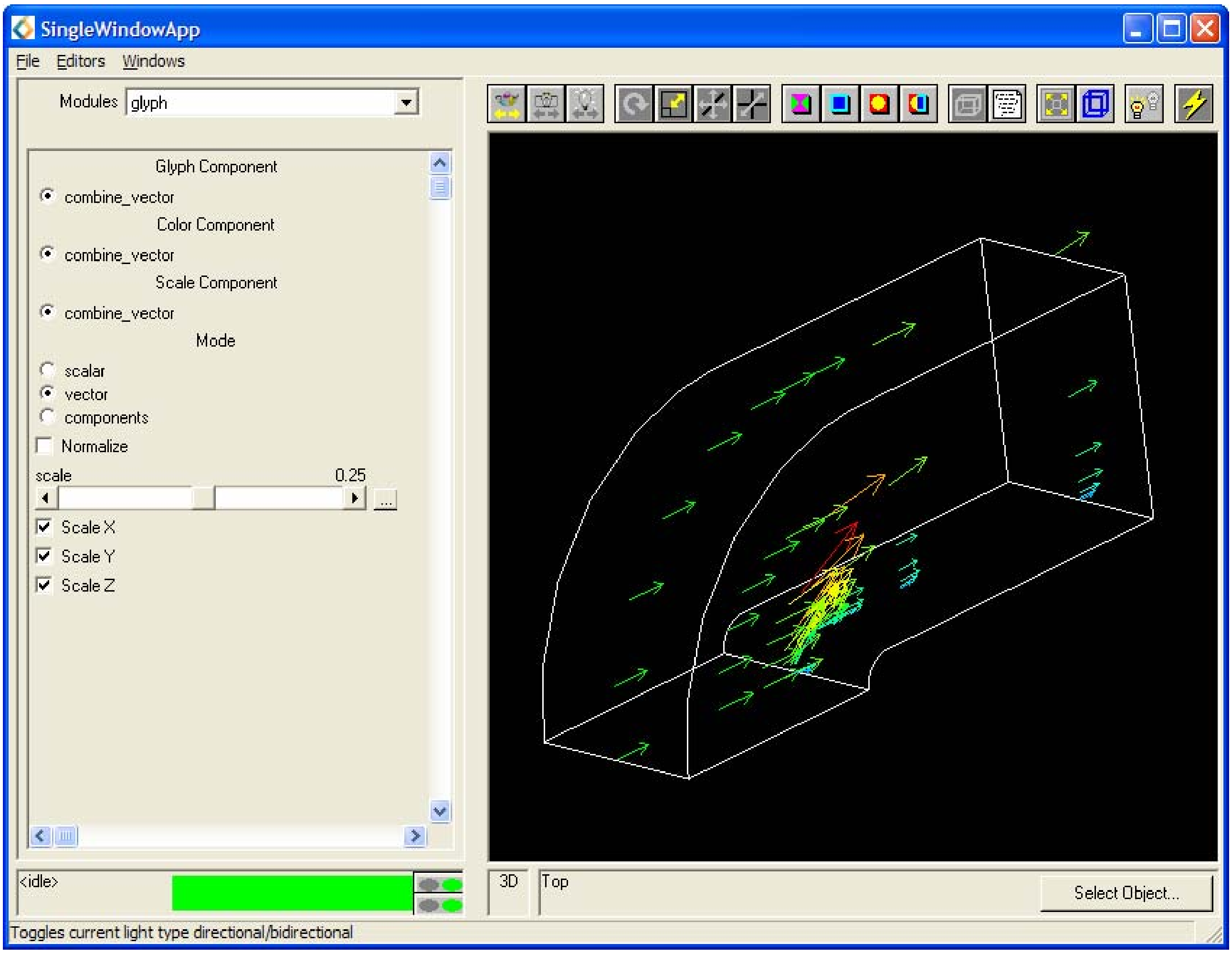

Figure 4.4: `glyph` Editor Panel

- Select the `glyph` modules control panel.
- Reduce the scale value using the **scale** slider to create arrows of a more acceptable size.
- Alter some of the other parameters – use the module help facility to understand their effect.

4.2.6 Moving the `slice_plane`

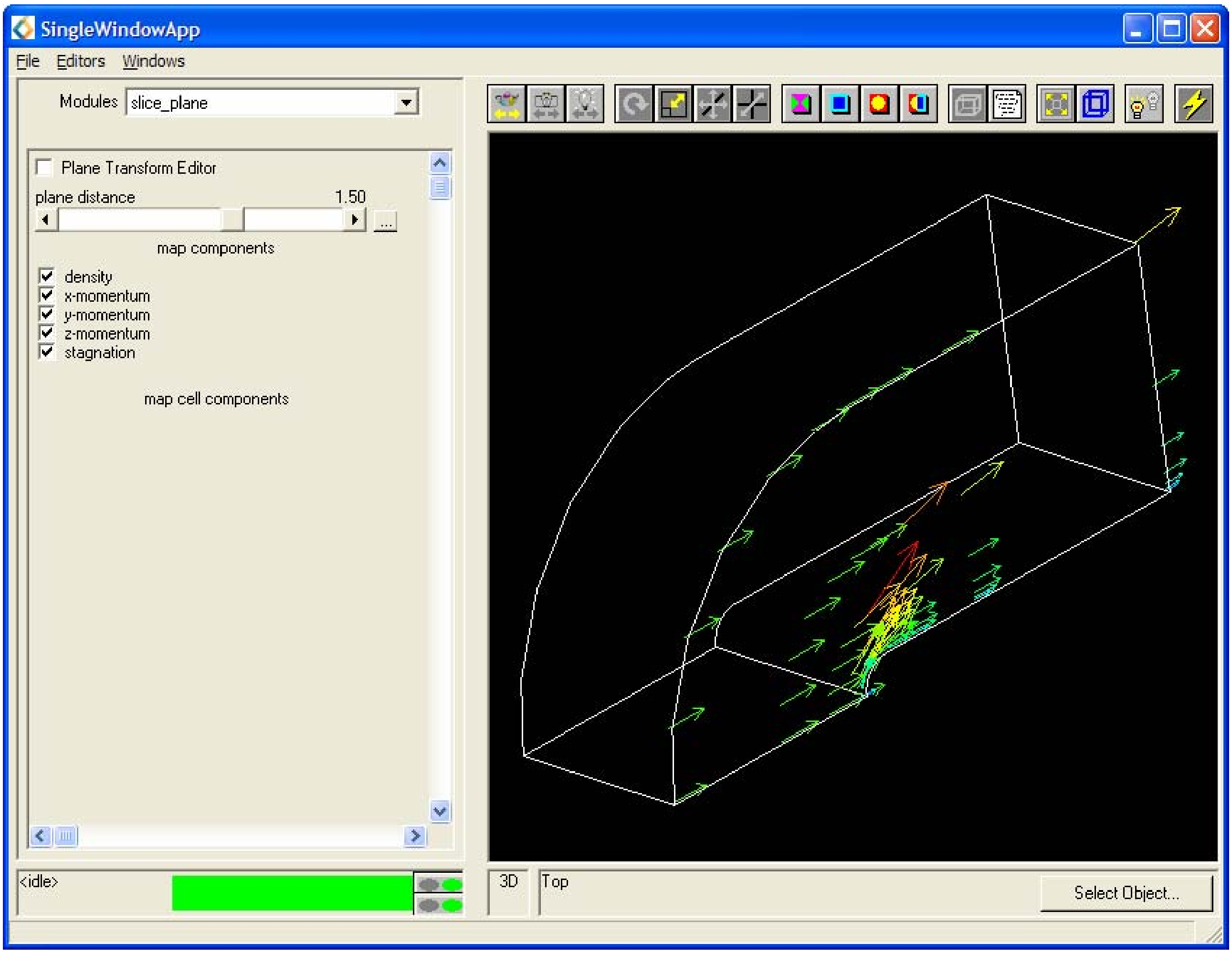

Figure 4.5: `slice_plane` Editor Panel

- Select the `slice_plane` modules Editor Panel.
- Alter the **plane distance** value to observe the value of momentum at different positions within the dataset.
- Change the orientation of the slice plane by altering the plane transformation parameters in the **Plane Transformation Editor**.

4.2.7 Visualizing the Scalar Data

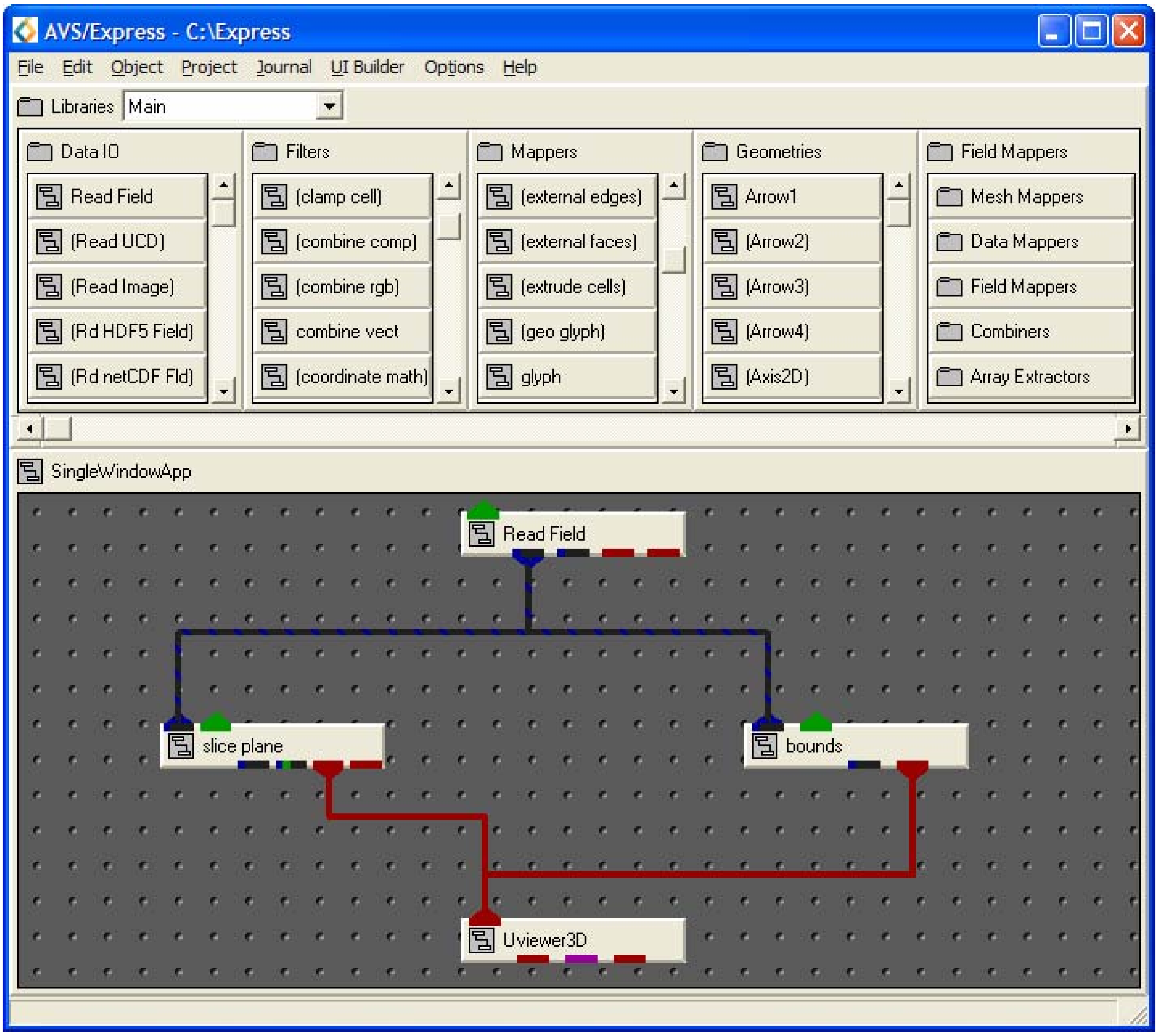

Figure 4.6: Network to View `fin.dat` as Scalar Data

Change the network to match that shown in Figure 4.6:

- `Read_Field` (Data Input): reads an AVS field file and converts it into an AVS/Express field.
- `slice_plane` (Mappers): extract a 2D slice from a 3D field with an arbitrarily positioned slice plane.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `UViewer3D` (Viewers): creates a 3D image in the DataViewer.

4.2.8 Using the `slice_plane` Module

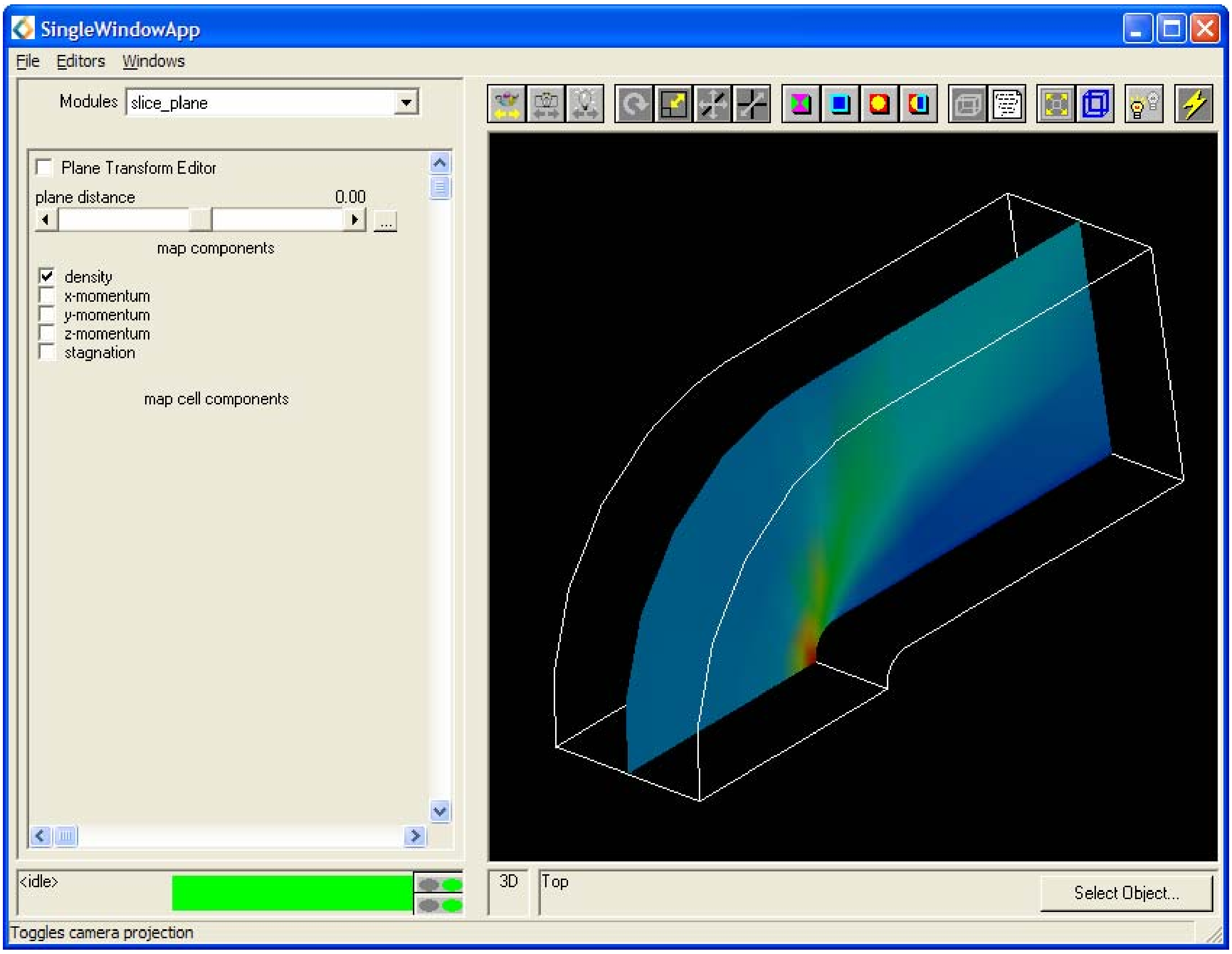

Figure 4.7: `slice_plane` Editor Panel

- Select the `Read_Field` module from the **Modules** drop down list and select the file `fin.fld` to import the data using the file browser.
- Select the Editor Panel for the `slice_plane` module and activate the **density** data component.

This module will now pass on a 3D field with a scalar data element to the other modules in the AVS/Express network.

- The Viewer window should now show an image similar to Figure 4.7.
- Use the **plane distance** slider to change the slice position.
- Extract the other scalar components using this same module.

4.3 Exercise 2

The aim of this exercise is to visualize some data obtained from a climate modelling experiment. The data contains values of:

- Temperature
- Specific humidity
- u, v, and w components of wind velocity

These values are sampled over the surface of the globe at several altitude levels. There are two sets of coordinates for the sampling: rectangular and spherical.

- Present the scalar data (temperature and specific humidity) using the spherical coordinate system.
- Present the vector data, the resultant wind velocity, using the rectangular coordinate system.
- The solutions to the field files for this exercise are given in Appendix B.

4.3.1 Importing the Climate Data

Below is a description of the data files containing the climate modelling data.

Coordinates: The data is irregularly spaced on a `49x12x40` grid. The coordinates are provided in ASCII files `ccm1.rect` which gives the coordinates in rectangular space and `ccm1.sph` which gives the coordinates in spherical space.

Both files have one line of header information and are arranged with all the coordinates of the first dimension followed by all the coordinates of the second dimension, followed by all the coordinates of the third dimension. There are eight values on each line of the files i.e.

```
x1   x2   x3   x4   x5   x6   x7   x8
x9   x10  ......................
y1   y2   y3   y4   y5   6y   y7   y8
y9   y10  ......................
y1   z2   z3   z4   z5   z6   z7    z8
y9   z10  ......................
```

Temperature and Specific Humidity: Temperature and specific humidity data are contained in binary files called `temp.dat` and `humid.dat` respectively. The data consists of a single floating point value at each grid point. There are 16 bytes of header information in each file.

Wind Velocities: The wind velocity data is contained in three binary files of floating point values for the u, v and w components. The files are called `bi.u.49`, `bi.v.49` and `bi.w.49`, respectively. Each file contains 16 bytes of header information.

World Map: A simple map of the world showing land/ocean outlines is contained in the file `world.asc`. The file is in ASCII format and irregularly spaced on the same grid as the other files. There is one line of header information.

In order to visualize this information it is necessary to write three field files:

- Create a field file called `climscal.fld` to describe the scalar data (temperature and specific humidity) using the spherical coordinate system.
- Create a field file called `climvect.fld` to describe the vector data (wind velocity) using the rectangular coordinates.
- Create a field file called `worldrec.fld` to describe the world map using the rectangular coordinates.

4.3.2 Visualizing the Scalar Data

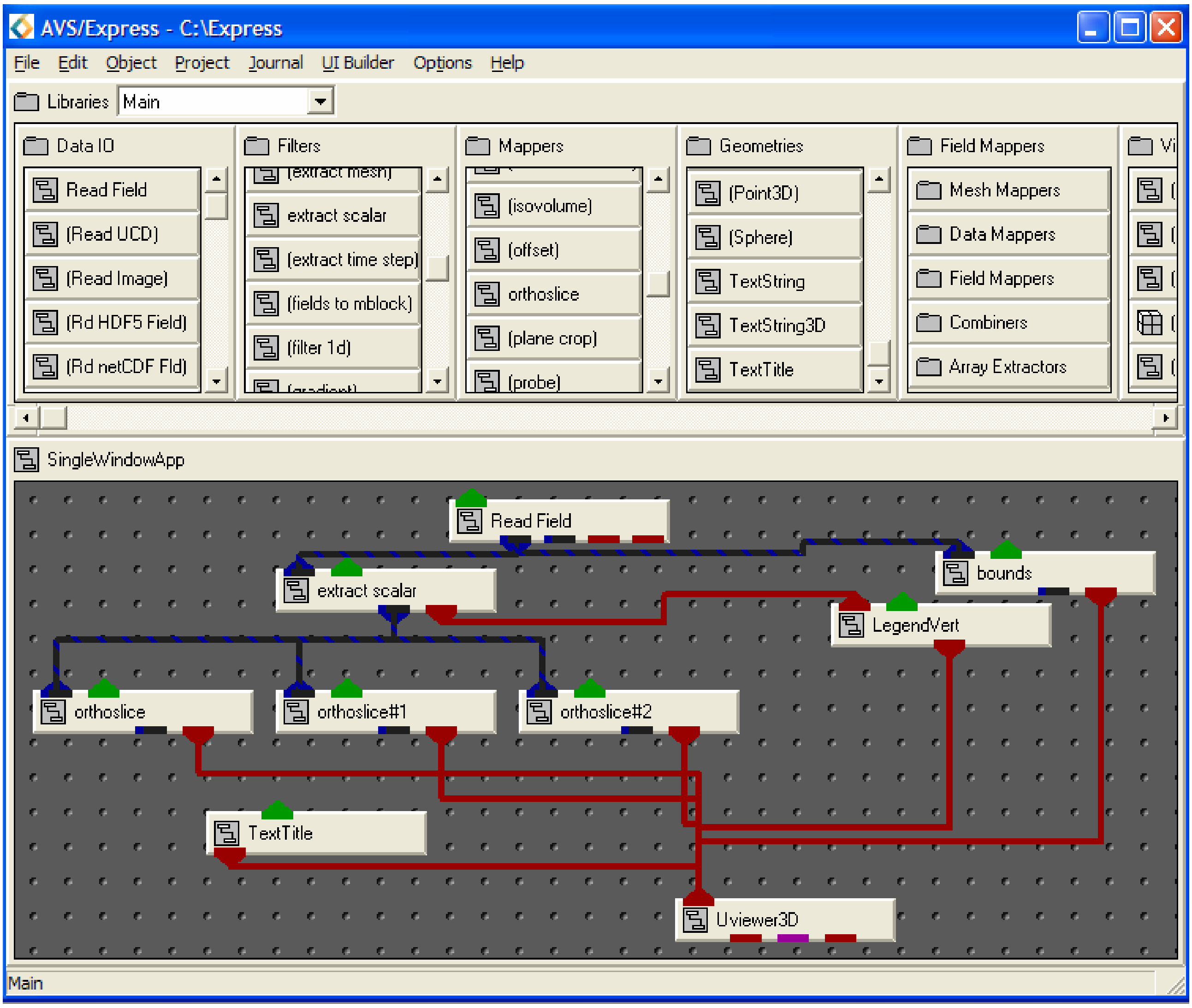

Figure 4.8: Network to Visualize the Scalar Data

Create the network shown in Figure 4.8 using the following modules:

- `Read_Field` (Data Input): reads an AVS field file and converts it into an AVS/Express field.
- `extract_scalar` (Filters): extract a single scalar data element from a field's vector component.
- `orthoslice` (Mappers): produces a slice of a structured field perpendicular to a selected coordinate axis.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `LegendVert` (Geometries): produces a colour legend which displays the data objects datamap.
- `TextTitle` (Geometries): Produces a title.
- `UViewer3D` (Viewers): creates a 3D image in the DataViewer.

4.3.3 Changing the `orthoslice` Parameters (1)

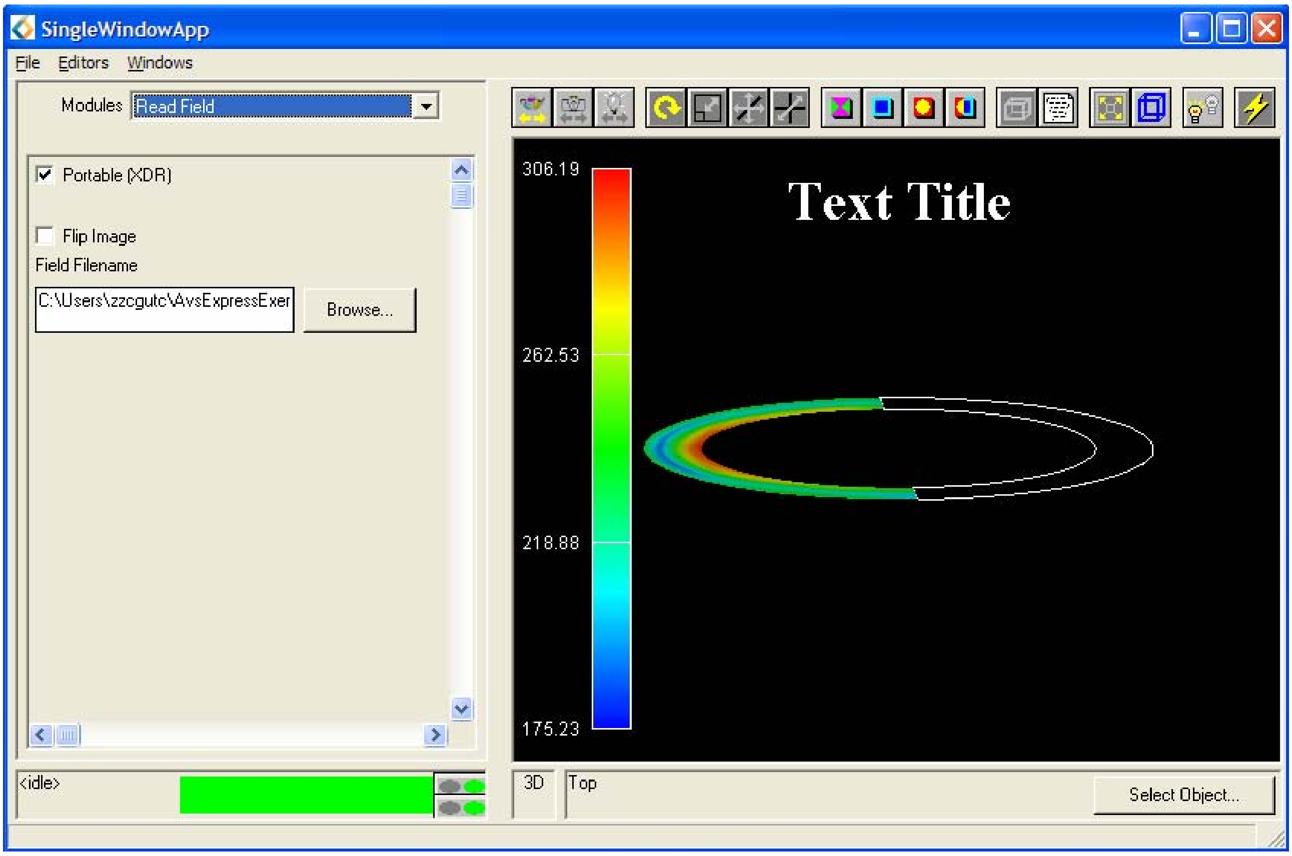

Figure 4.9: Initial Image in Data Viewer Pad

- Select `Read_Field` from the **Modules** Editor Panel and select `climscal.fld` from the file browser.
- An image similar to the one shown in Figure 4.9 should appear in the Data Viewer Pad.
- Select the `extract_scalar` module and ensure that **temperature** is selected.
- Select `orthoslice`, `orthoslice#1` and `orthoslice#2` modules in turn and set the **axis** values to 0, 1 and 2 respectively.

4.3.4 Changing the Legend and Label Parameters

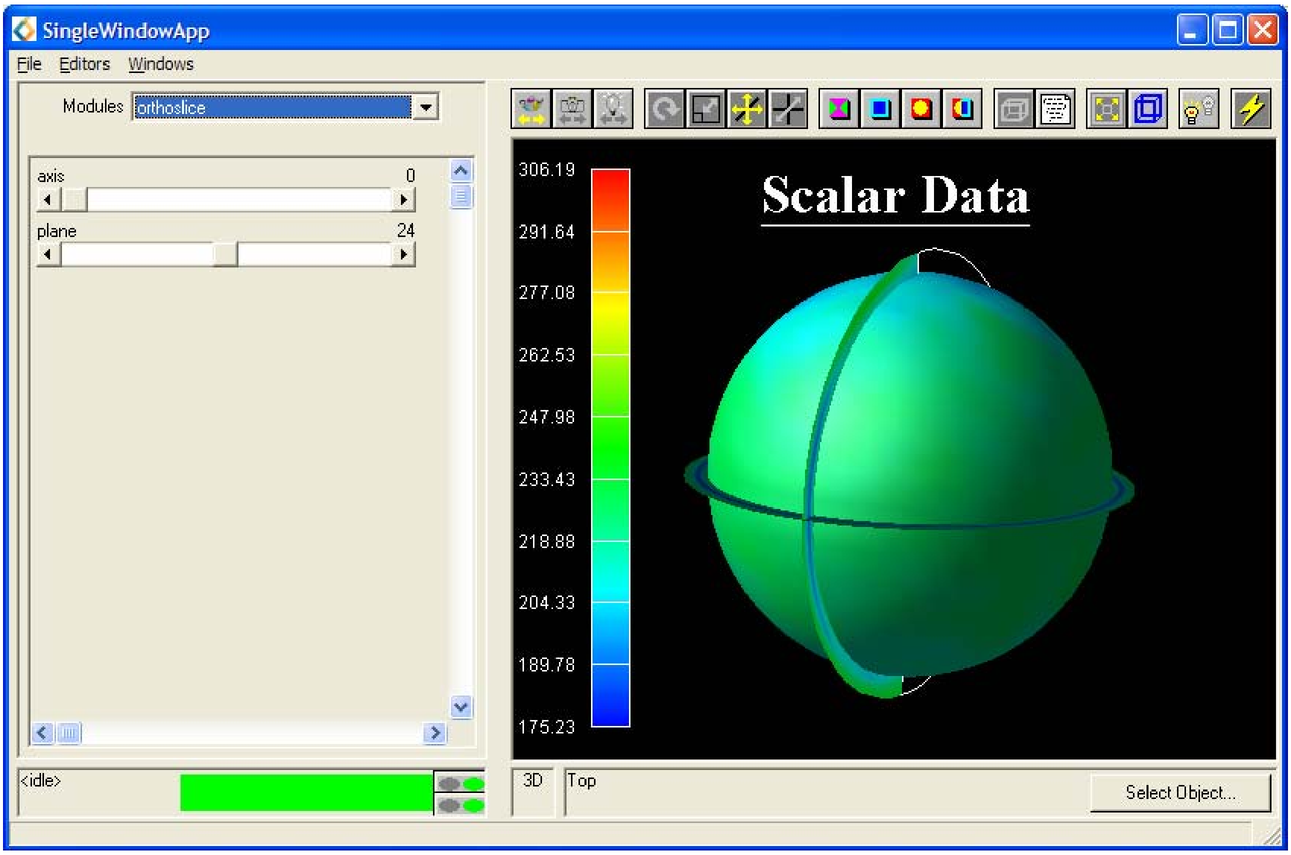

Figure 4.10: `orthoslice` Editor Panel and Transformed Image

- Rotate the image so it appears similar to that shown in Figure 4.10.
- Select the `LegendVert` module and change the parameters to produce a vertical legend similar to that shown in Figure 4.10.
- Select the `TextTitle` module and change the parameters to produce a title similar to that shown in Figure 4.10.

4.3.5 Changing the `orthoslice` Parameters (2)

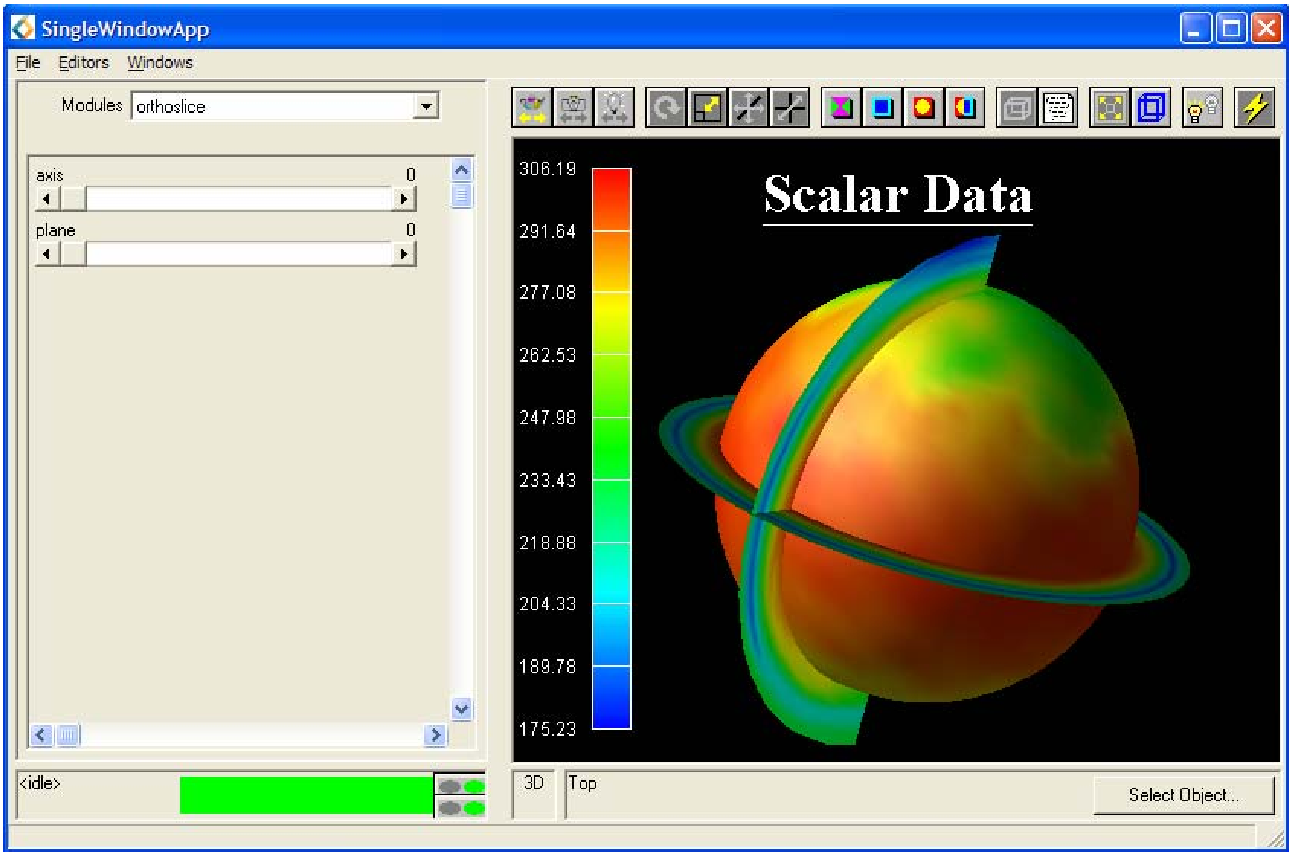

Figure 4.11: `orthoslice` Editor Panel with **plane** 0

- Select the `orthoslice` and `orthoslice#1` modules, (or the modules with axes set to 0 and 1), and reduce the **plane** sliders to 0.
- An image similar to that shown in Figure 4.11 should be obtained.
- The image can be further improved by selecting the `bounds` module and either removing the **hull** or selecting **data**. This will colour the bounds the same as the boundary value of the data (not just white).

4.3.6 Observing the Temperature Data

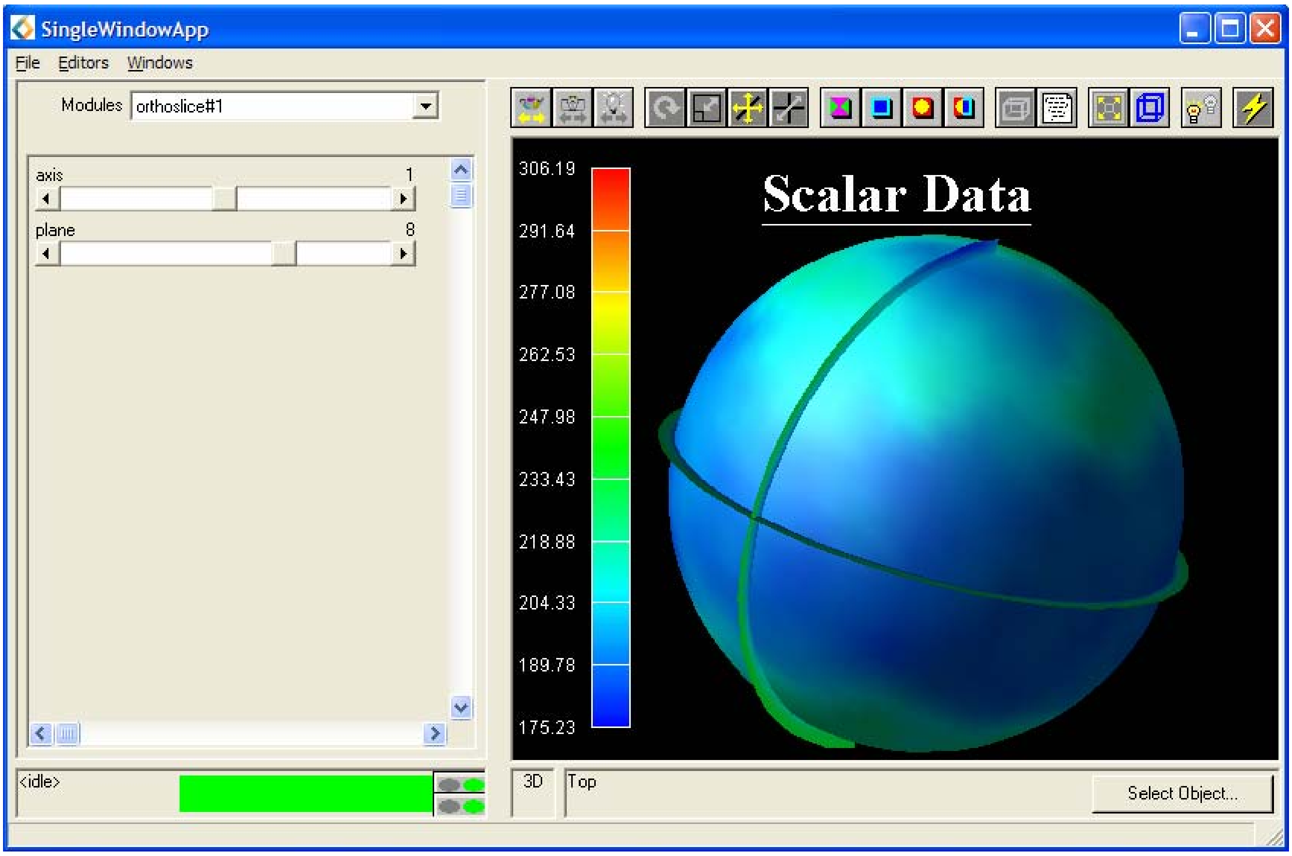

Figure 4.12: `orthoslice` Editor Panel with **plane** 5

- Select the `orthoslice#1` module whose **axis** is set to 1, and whose **plane** slider should have been set to 0.
- This orthoslice is currently illustrating the temperature around the surface of the earth.
- Increase the value of the **plane** slider and observe how the temperature of the atmosphere changes as altitude increases.

4.3.7 Observing the Humidity Data

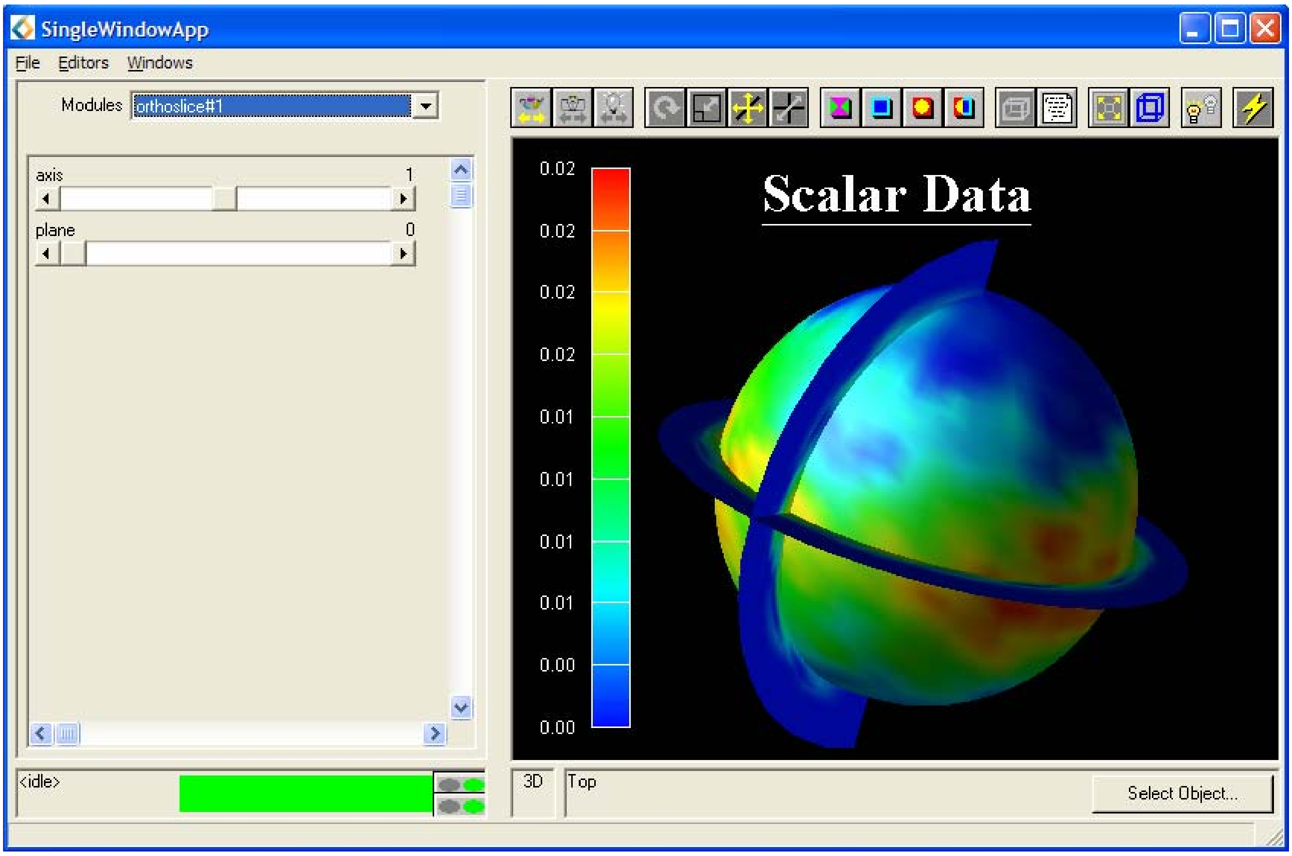

Figure 4.13: `orthoslice` Editor Panel with **plane** 0 for Humidity Data

- Select the `extract_scalar` module and select **humidity**.
- Investigate how specific humidity varies with altitude.

4.3.8 Visualizing the Vector Data

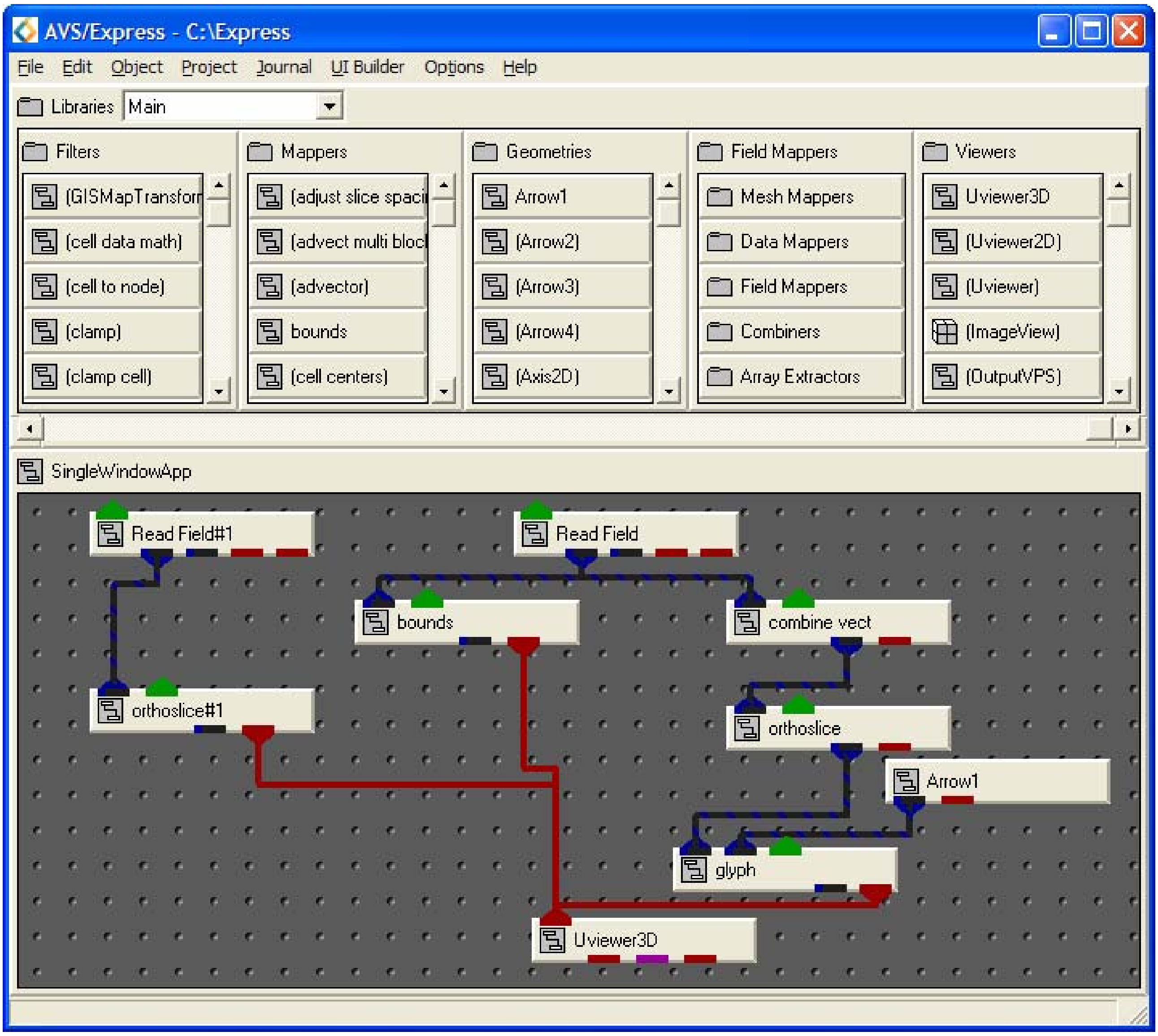

Figure 4.14: Network to Visualize the Vector Data

Create the network shown in Figure 4.14 using the following modules:

- `Read_Field` (Data Input): reads an AVS field file and converts it into an AVS/Express field.
- `orthoslice` (Mappers): produces a slice of a structured field perpendicular to a selected coordinate axis.
- `combine_vect` (Filters): takes all selected scalar data components and combines to output a single vector component.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `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.

4.3.9 Reading the World Map

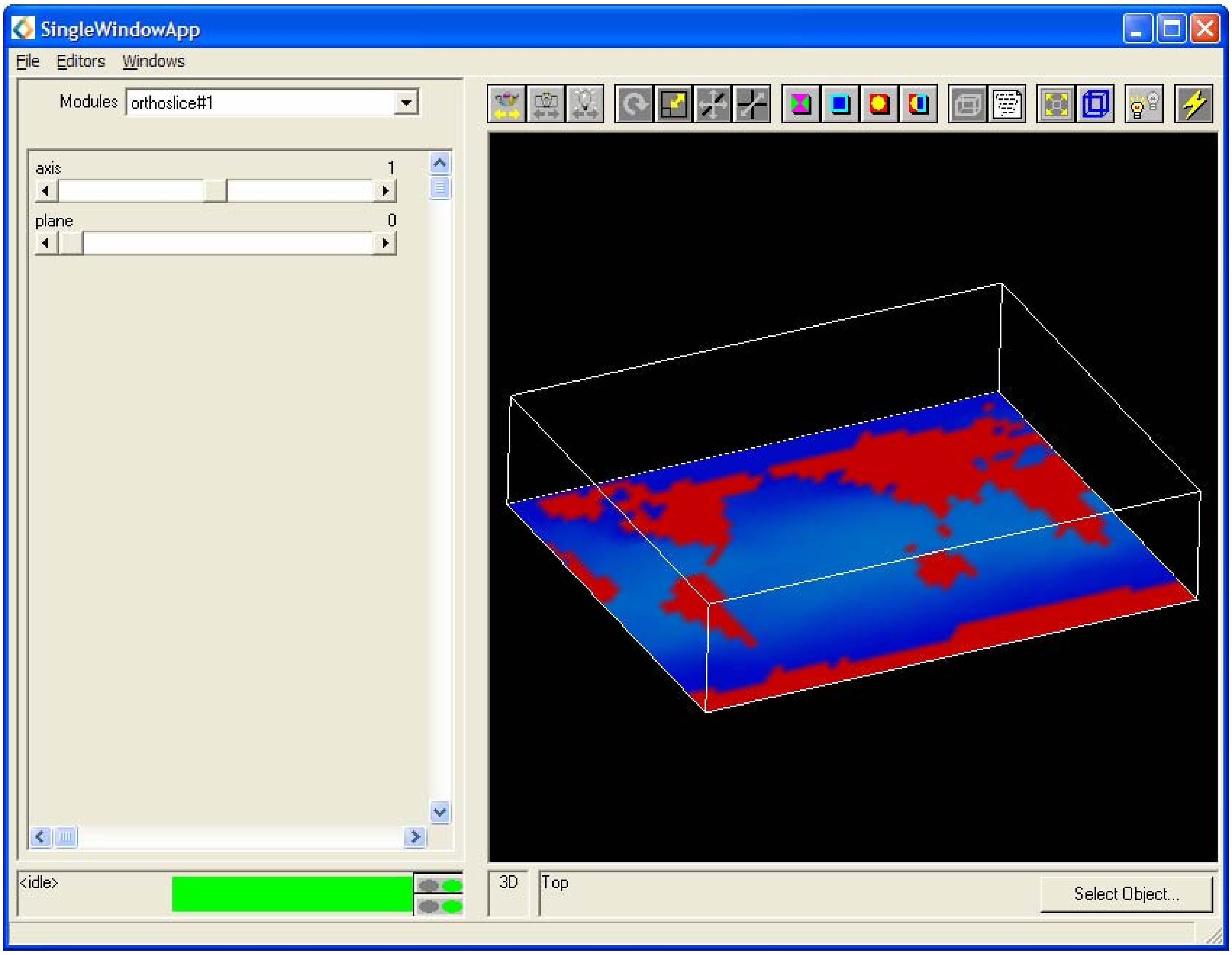

Figure 4.15: `orthoslice` Editor Panel

- Select the module `Read_Field#1` from the **Modules** drop down list, locate and select `worldrec.fld` from the file browser.
- Select the `orthoslice#1` module which is connected to the `Read_Field#1` module and set the **axis** value to 1.
- Reduce the **plane** slider to 0.
- Using the mouse, rotate the picture until it is orientated as shown in Figure 4.15.

4.3.10 Reading the Velocity Data

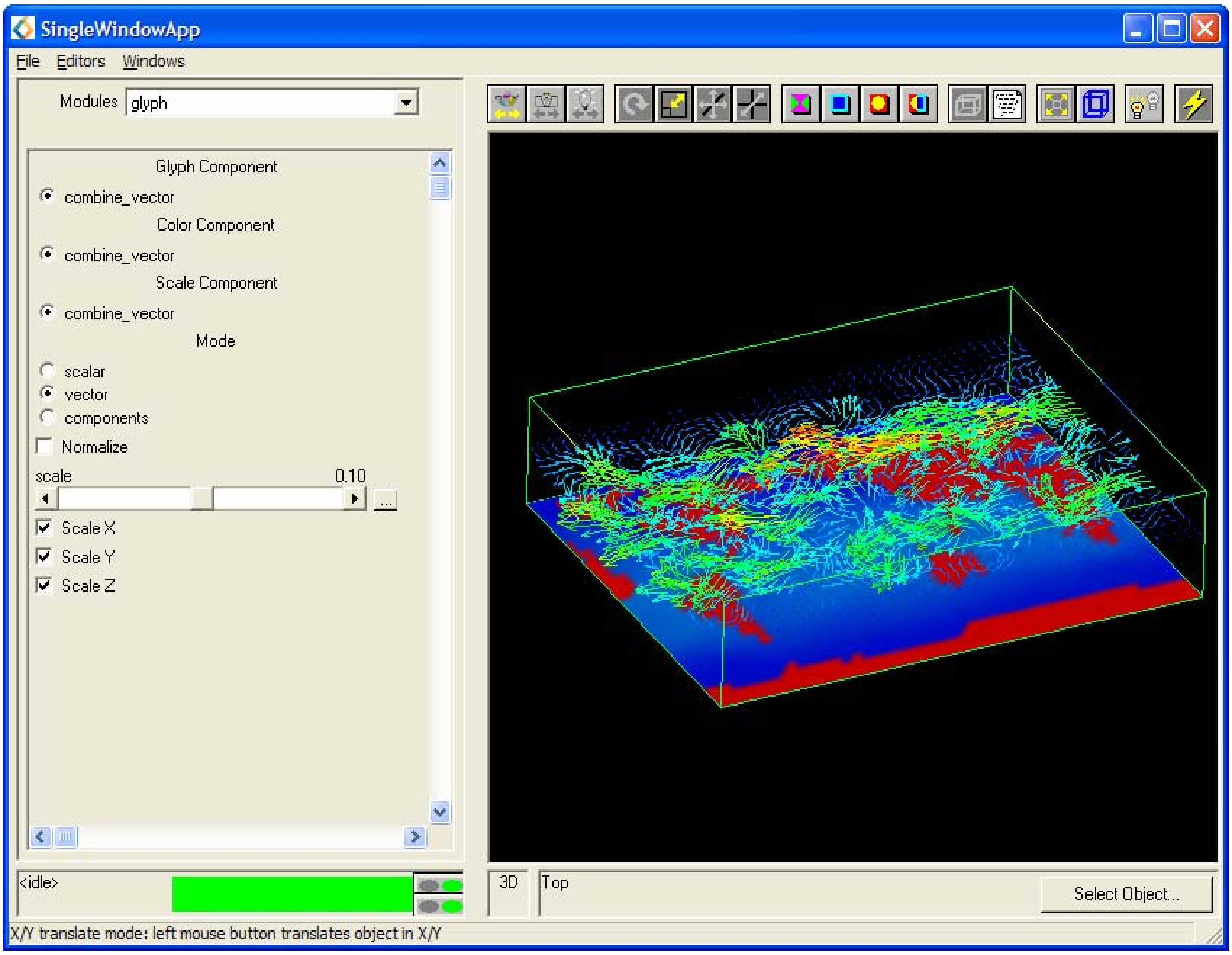

Figure 4.16: Wind Velocity Data Illustrated using the `glyph` Module

- Select the `Read_field` module and select `climvect.fld` from the file browser.
- Select the `combine_vect` module and ensure that **u-comp**, **v-comp** and **w-comp** are selected.
- Select the `orthoslice` module that is attached to `Read_Field` and set the **axis** to 1.
- Select the `bounds` module and select **data** to colour the bounds.
- Select the `glyph` module and reduce the **scale** to ~ 0.1 or until the arrows are a sensible size.
- An image similar to the one shown in Figure 4.16 should be obtained.

4.3.11 Looking at the Velocity Data

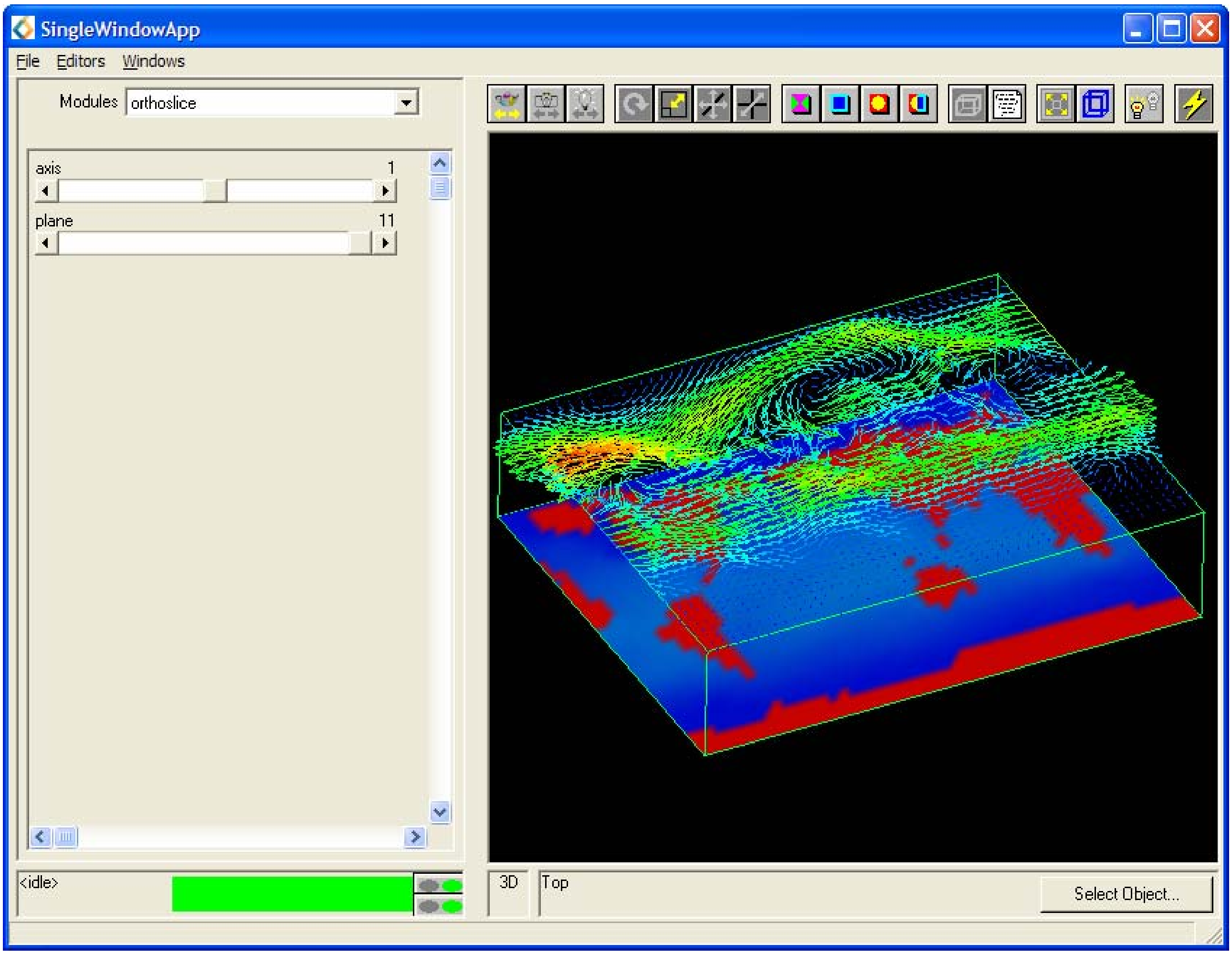

Figure 4.17: `orthoslice` Editor Panel with **Plane** 11

- Select the `orthoslice` module concerned with the velocities and set the **plane** slider value to 0.
- This shows the velocities on the surface of the earth.
- Increase the **plane** value and observe how the wind velocities change with increasing altitude.

Figure 4.17 shows the wind velocities at the highest altitude level.

4.3.12 Changing the Datamap

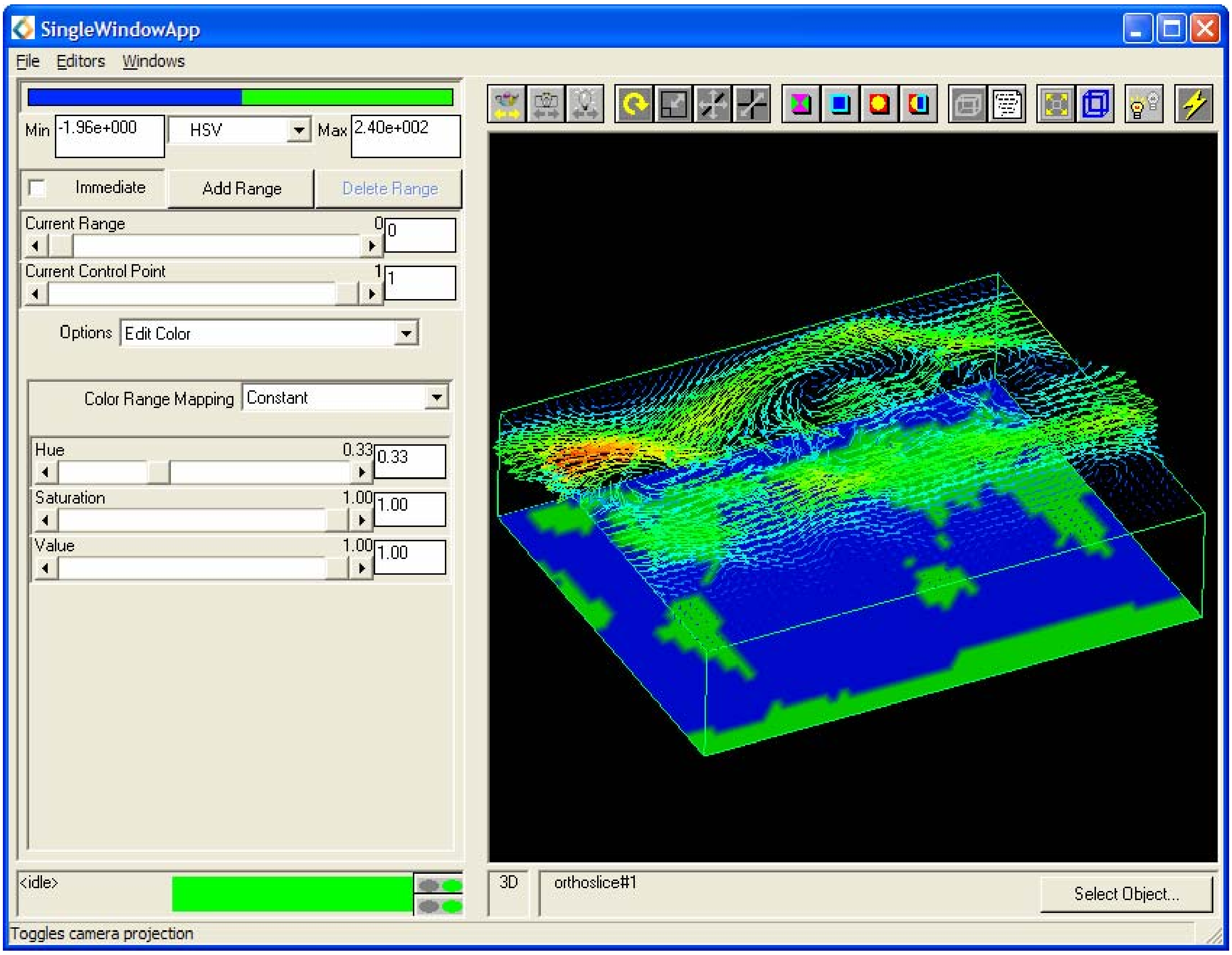

Figure 4.18: **Datamap editor**

- To improve the appearance of the image, alter the **Datamap** of the world map so as to colour the land green and the oceans blue.
- Refer back to Section 3.5.6 on page 34 in order to achieve this datamap alteration.

(Hint: Select the world map as the current object and set the **Colour Range Mapping** as **Constant**. Alter the values of the **Current Control Point** slider (Figure 4.18.))

Using the Software Renderer

The renderer option can be altered in order to alter the renderer which is used to draw the display. There are several options:

- **Software**: software renderer which uses its own graphics rendering techniques.
- **OpenGL**: uses the workstations own software and hardware graphics techniques (OpenGL).

The OpenGL/hardware renderer is used as the default. However, when using the volume render to observe 3D images, it is often necessary to use the software renderer. The software renderer is selected in the following way.

- Select the **View** editor from the **Editors** pull down menu.
- In the **Renderer** menu select Software.
- Proceed with the examples as usual.

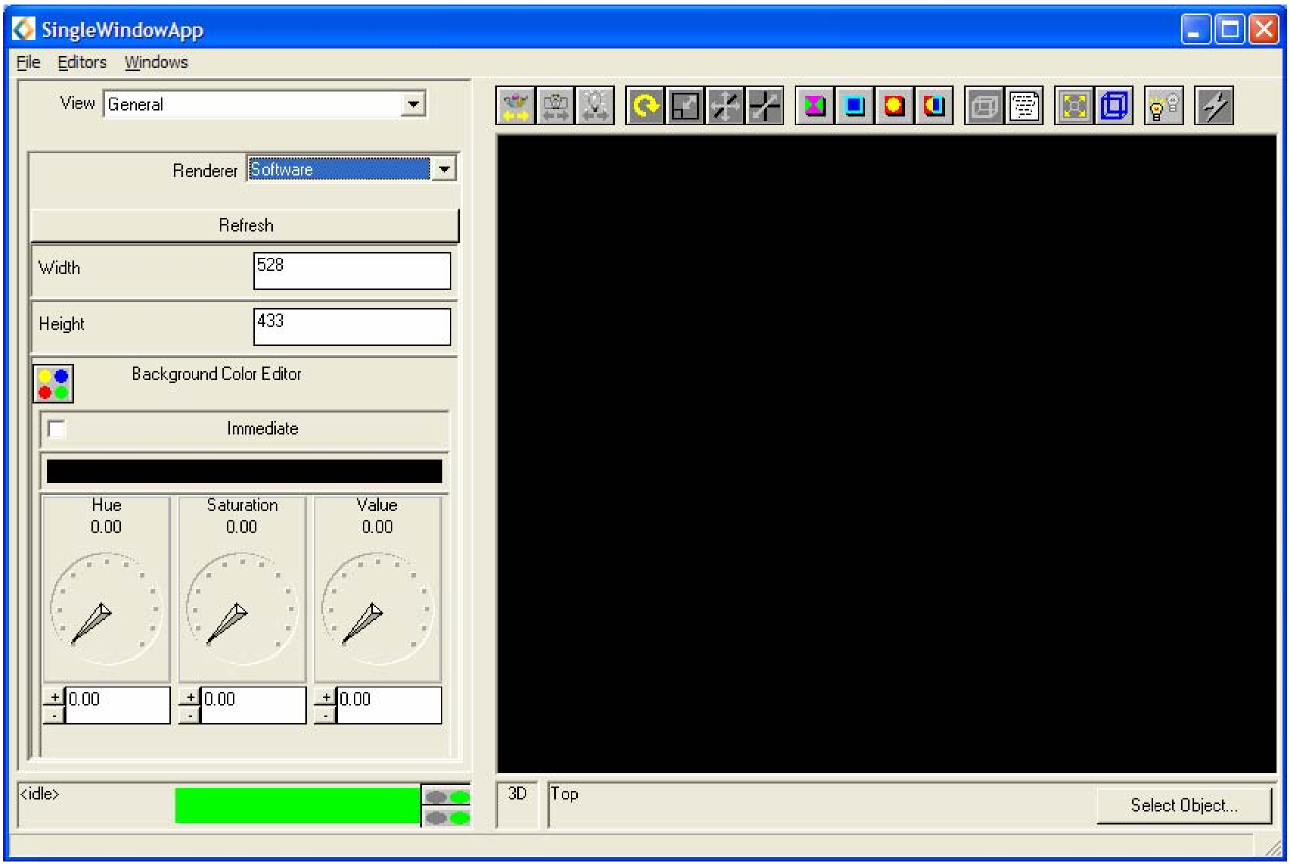

Figure 4.19: **View Editor**

The rest of the exercises in this chapter are concerned with 3D representation of various fields. The examples use the `volume_render` module and all require the use of the software renderer.

Note: Use the software renderer will significantly slow down the visualization process – but for the volume rendering this is unavoidable.

4.4 Exercise 3

The volume to be represented in this example is a 3D image of a human head.

4.4.1 Displaying the `head.fld` Data using `volume_render`

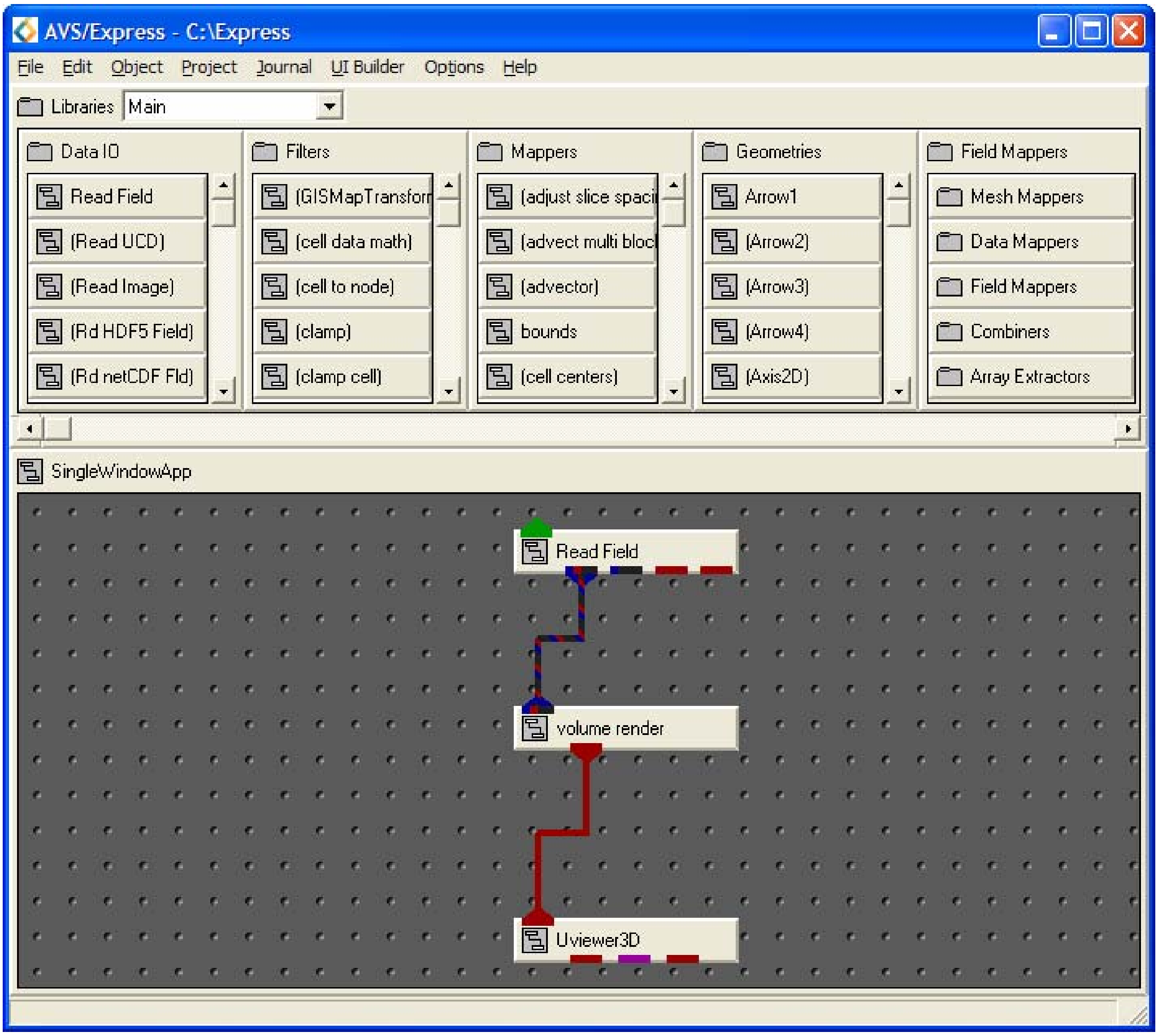

Figure 4.20: Network to Visualize `head.dat`

Create the network shown in Figure 4.20 using the following modules:

- `Read_Field` (Data IO): reads the AVS field file and converts it into an AVS/Express field.
- `volume_render` (Mappers): directly renders a volume.
- `Uviewer3D` (Viewers): creates a 3D image in the Data Viewer.

4.4.2 Normalizing the Image

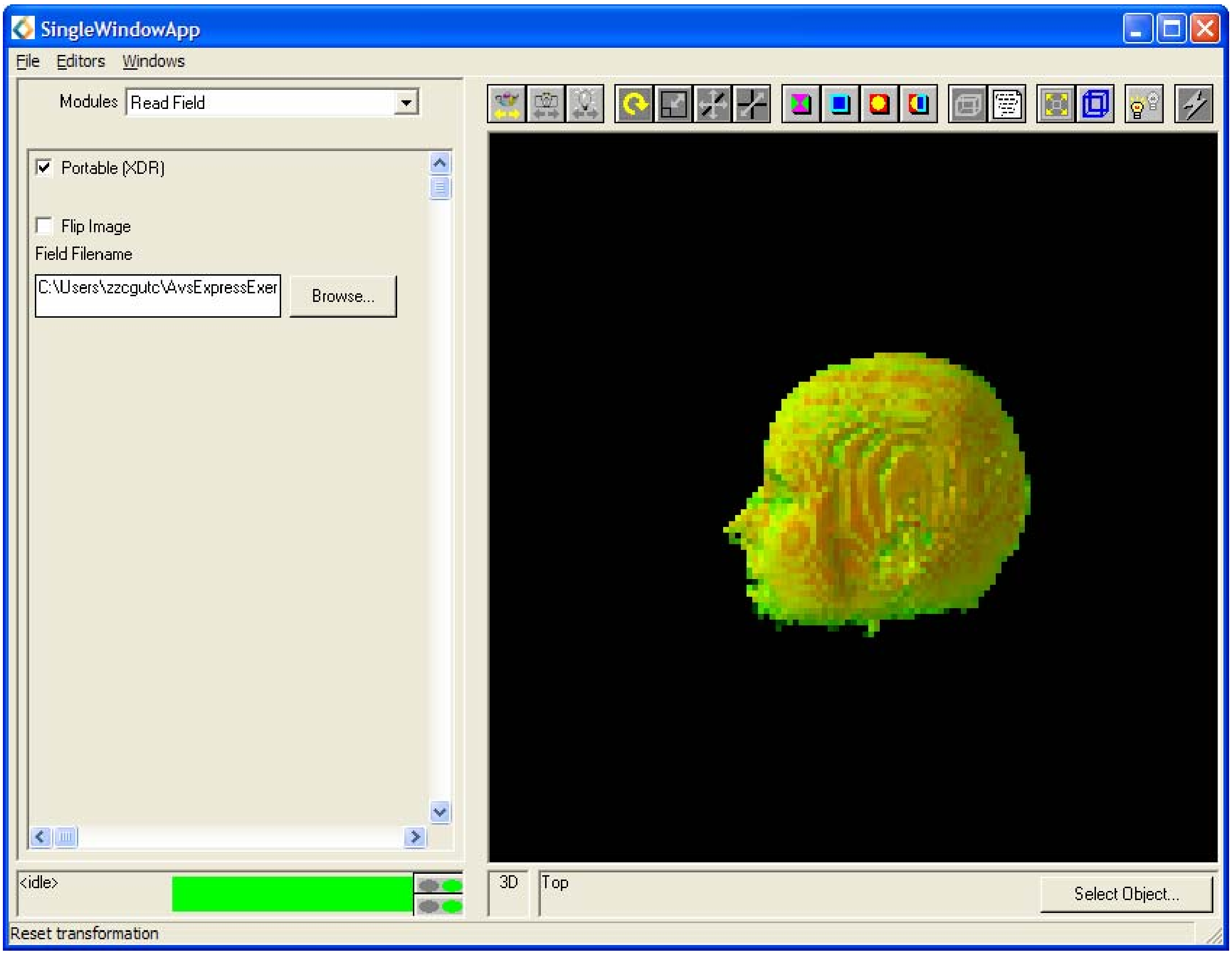

Figure 4.21: `volume_render` Editor Panel

- Select the `Read_Field` modules Editor Panel and select the file `head.fld`, from the file browser, to import the data.

Note – if an error message is reported when `Read_Field` is selected, or if no image appears, it is likely that the software renderer has not been set.

- Click the **Reset/Normalize/Center** icon in the Toolbar in order to fit the whole object in the window. The head should now be visible.
- Select the `volume_render` module Editor Panel as shown in Figure 4.21.

The steps listed on the following page will improve the appearance of the image and explain certain properties of the `volume_render` module.

4.4.3 Using `volume_render`

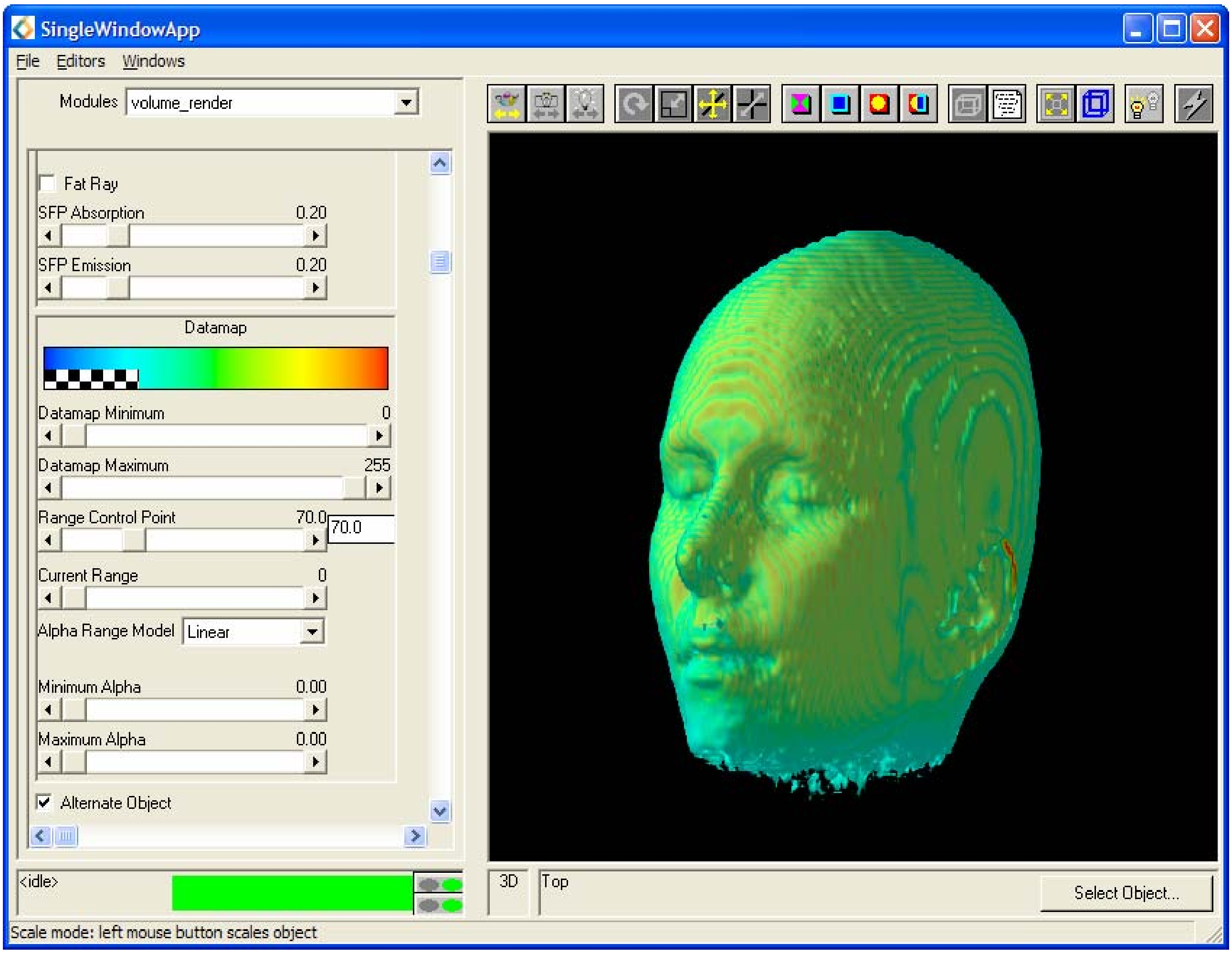

Figure 4.22: volume_render Editor Panel and Improved Image

- Switch the **Surface Mode** to **Gouraud shading**. This selection provides the highest quality rendering, includes both lighting and interpolation.
- Uncheck the **Fat Ray** option box in order to improve the resolution of the image.
- The `volume_render` module has two default data maps. The range to be edited can be selected by altering the value of **Current Range**. The **Range Control Point** value allows the mid-point of these two ranges to be altered. Set the **Range Control Point** value to approximately 70.
- Set the **Alpha Range Model** to **Linear**. The alpha component is interpolated between the maximum and minimum current data range. The other option is **constant**; the alpha component for the whole data range is set to the minimum alpha value.
- Rotate the head so that it appears similar to the image shown in Figure 4.22.
- Alter some of the alternative `volume_render` parameters and observe the effects.

4.5 Exercise 4

In this exercise the electron density of the hydrogen atom will be represented, this time in 3D.

- Use the network constructed in Exercise 3.
- Select the `Read_Volume` modules control panel and select the file `hydrogen.dat`, in the file browser, to import the data.
- Click the **Reset/Normalize/Center** icon in the Toolbar so as to fit the whole object into the window.
- Change the parameters of the `volume_render` module in order to improve the appearance of the image. Use the `volume_render` property explanations given in Exercise 3.
- Using the mouse, transform the objects.
- Finally, add a `slice_plane` module to the network and investigate the obtained results.

4.6 Exercise 5

This example will use the `excavate_brick` module to observe the 3D image of a lobster encased in a brick.

4.6.1 Displaying the `lobster.dat` data using the `excavate_brick`

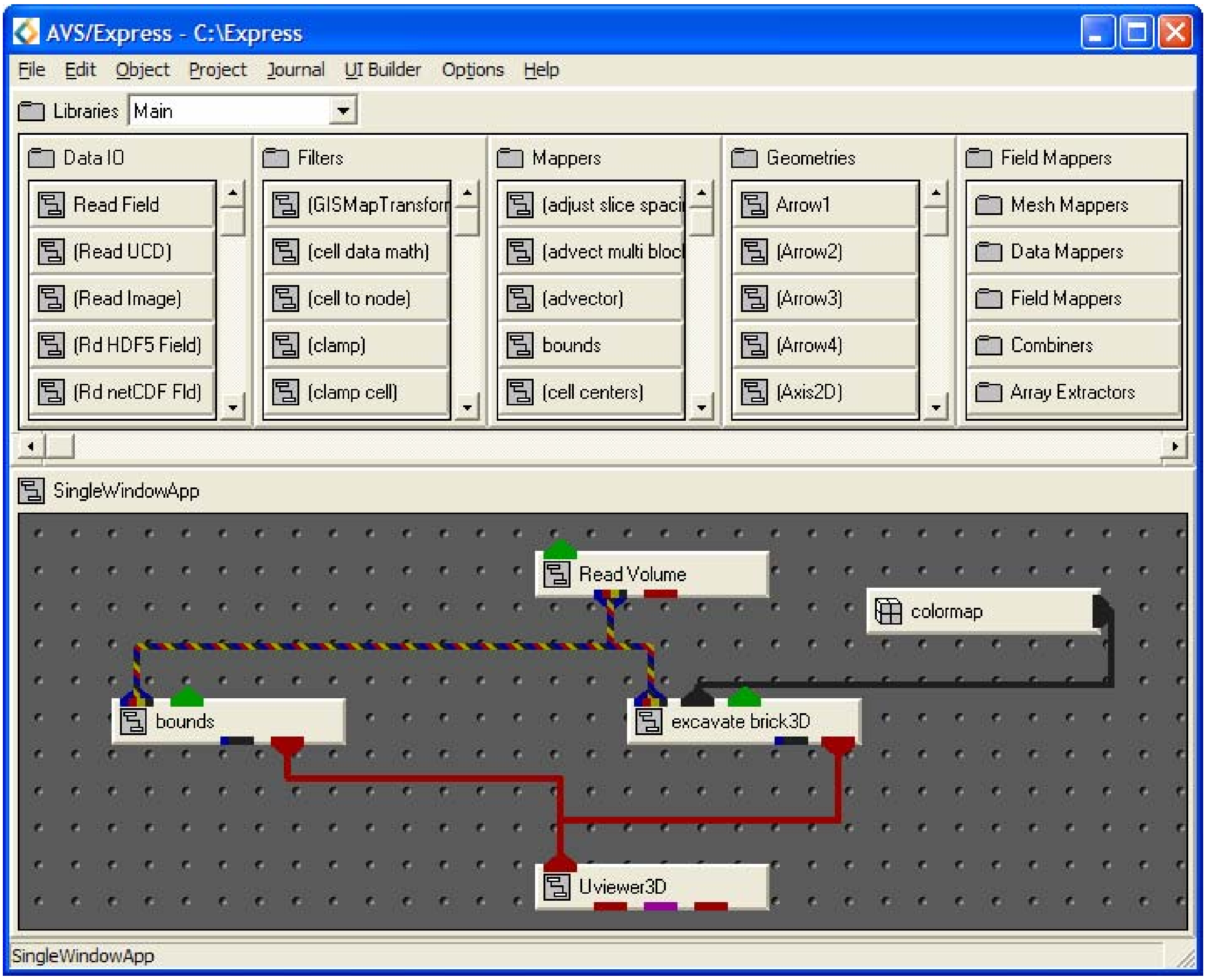

Figure 4.23: Network to Visualize `lobser.dat`

Create the network shown in Figure 4.23 using the following modules:

- `Read_Volume` (Data IO): reads a volume format file and outputs an AVS/Express field.
- `colormap` (Search for using **Object**, **Find in all libraries**): generates a coloured image.
- `Excavate_brick3D` (Mappers): technique for visualizing a 3D uniform volume. The volume is displayed with an X, Y and Z slice plane, which removes a rectangular sub-volume of the field, revealing the internal structure.
- `bounds` (Mappers): generates a bounding box of a 3D field.
- `Uviewer3D` (Viewers): produces a 3D image in the Data Viewer.

4.6.2 Normalizing the Image

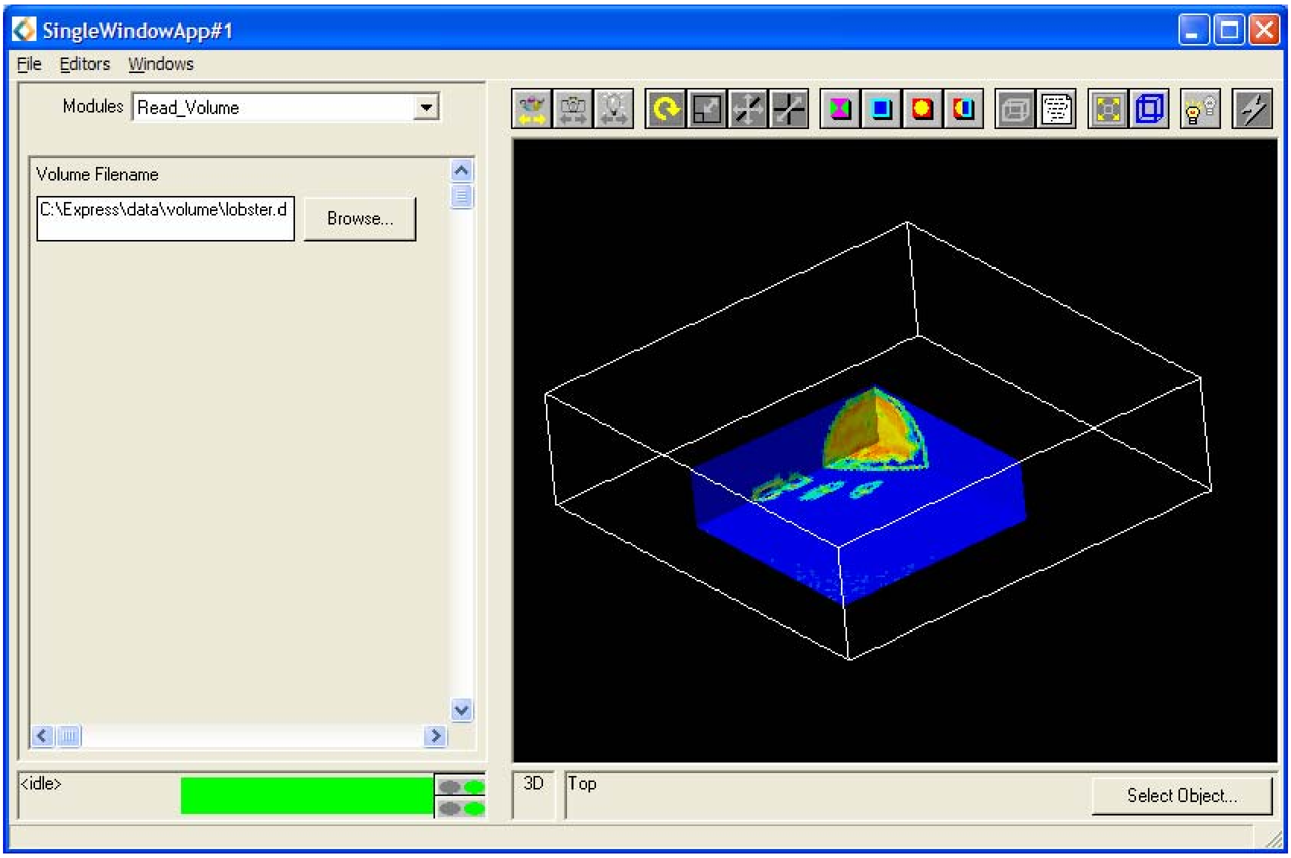

Figure 4.24: `lobser.dat` using `excavate_brick3D`

- Select the `Read_Volume` modules Editor Panel and select the file `lobster.dat` to import the data.
- Click on the **Reset/Normalize/Center** icon in the toolbar so as to fit the whole object in the window.
- Using the mouse, transform the object so it appears similar to that shown in Figure 4.24.

4.6.3 Using the `excavate_brick`

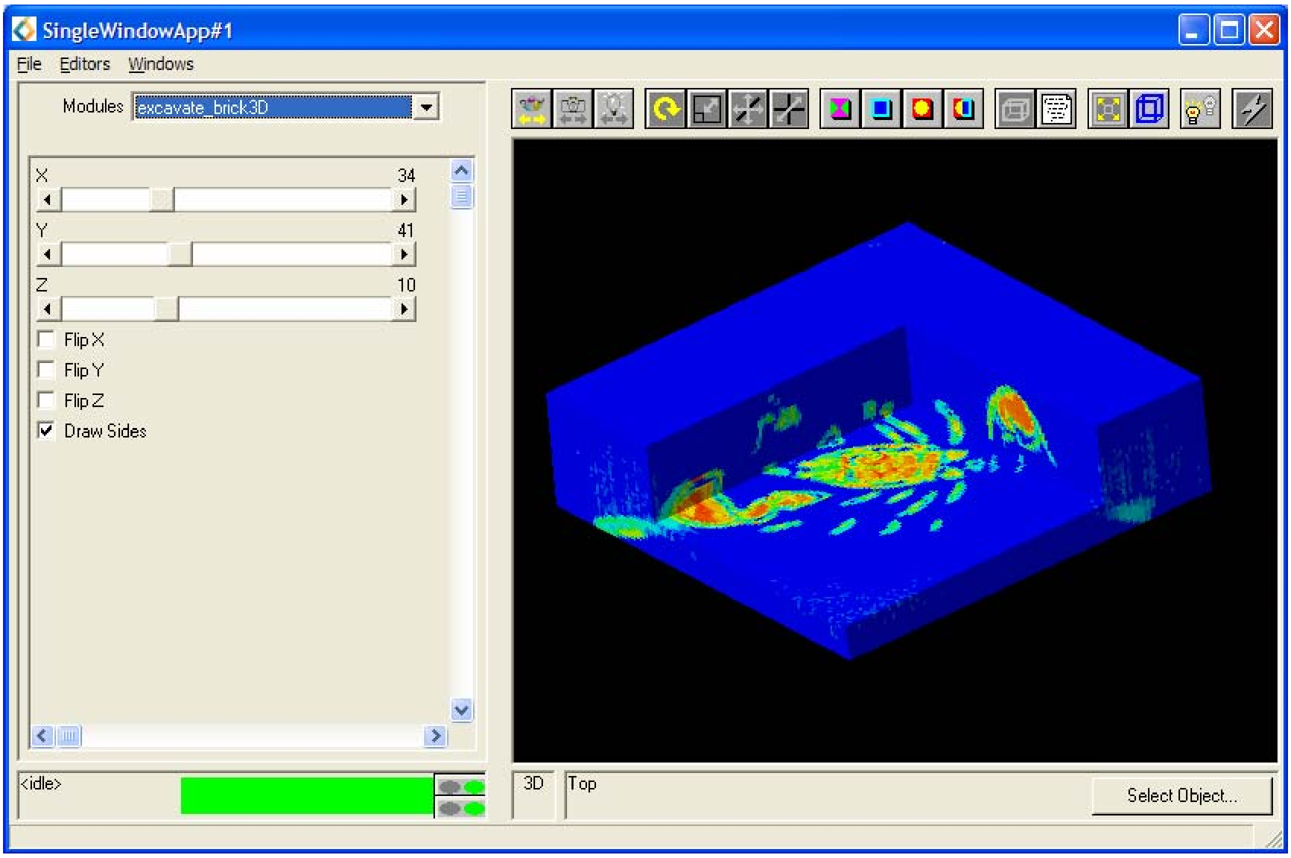

Figure 4.25: `excavate_brick3D` Editor Panel

- Select the `excavate_brick` Editor Panel from the **Modules** menu.
- Check the **Draw sides** option box, this causes the sides of the brick to be drawn.
- Alter the other parameters in order to observe their effects.
- Select the `bounds` module and uncheck the **Hull** options box.

5 Visualizing UCD Data

5.1 What is UCD Data?

UCD stands for Unstructured Cell Data. The format of UCD is a different way to read 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 a 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 shapes> nodes
<data label> <data units>
<nodeid> data
```

UCD files have a 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 this 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, 1b/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 understand the process, there are several prepared UCD formal files which can be found in the directory,

```
C:express\data\ucd
```

The exercises in this chapter are designed to demonstrate several examples of UCD visualization and cover 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 in order to demonstrate some simple networks which can be used to display UCD data.

5.3.1 Displaying the UCD Geometry (1)

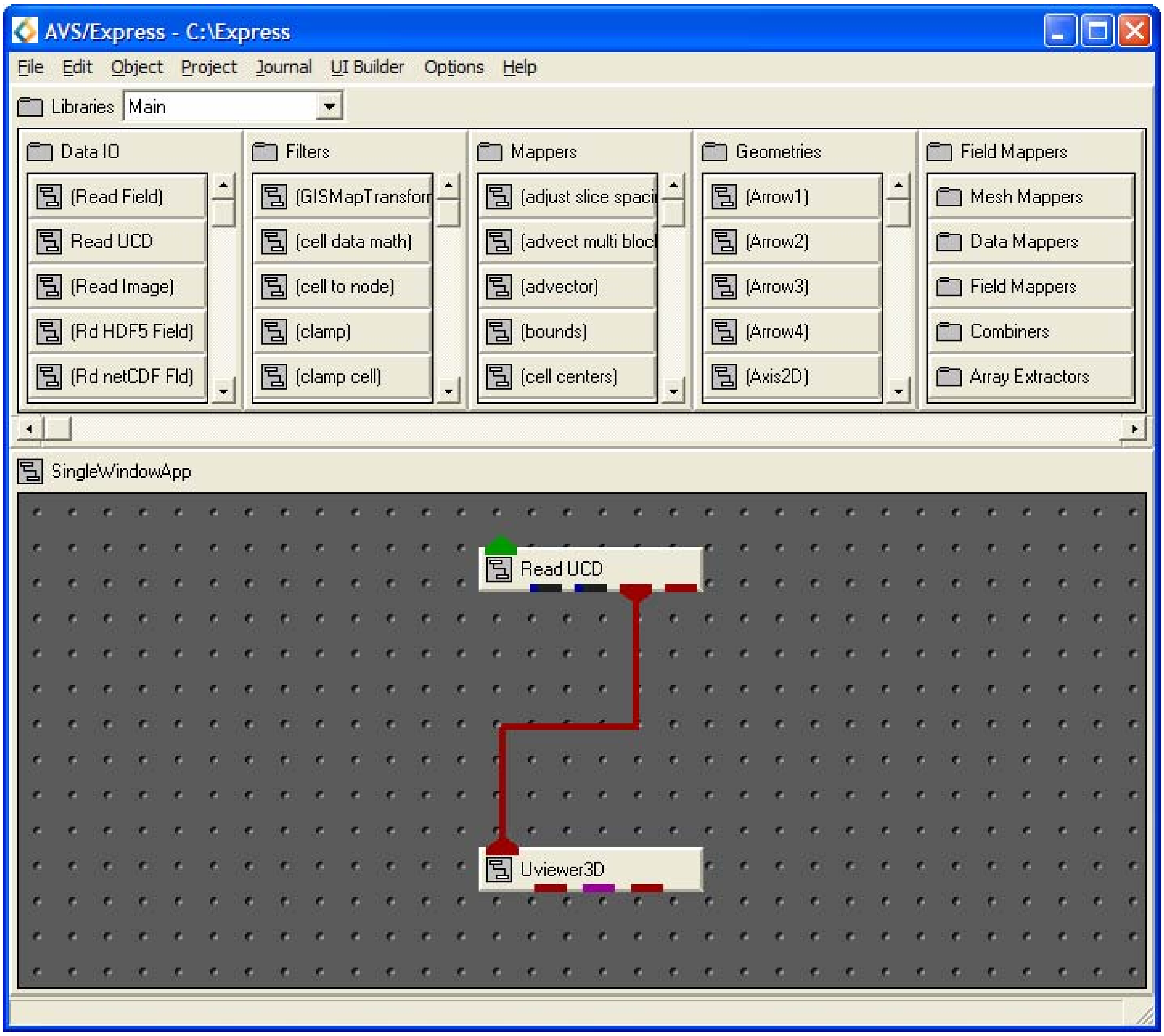

Figure 5.1: Simple Network to Visualize UCD Data

Create the network in Figure 5.1 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 Data Viewer.

5.3.2 Displaying the UCD Geometry (2)

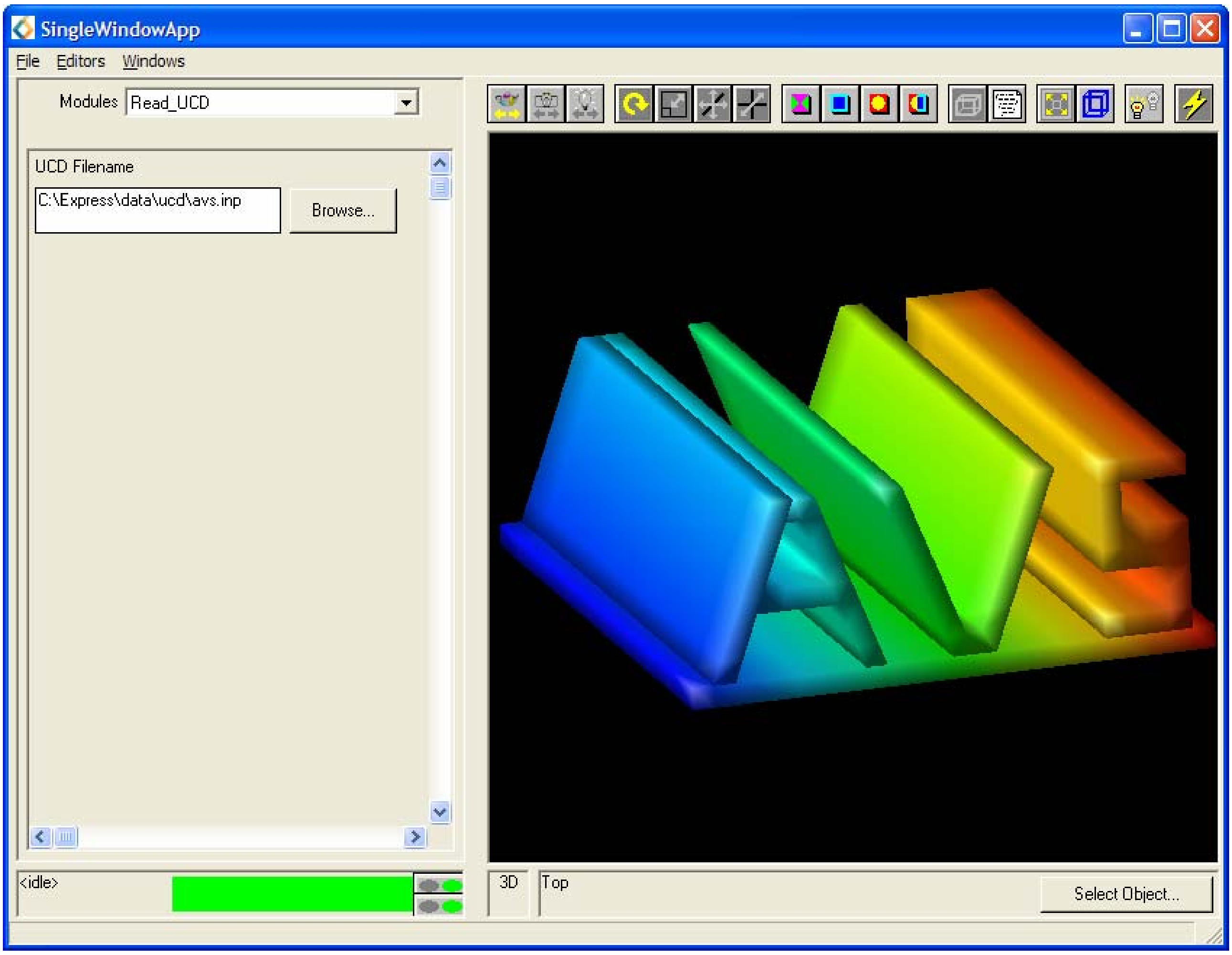

Figure 5.2: Visualization of UCD Data

- 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 image is similar to the one shown in Figure 5.2.

5.3.3 Displaying the UCD External Edges and Faces (1)

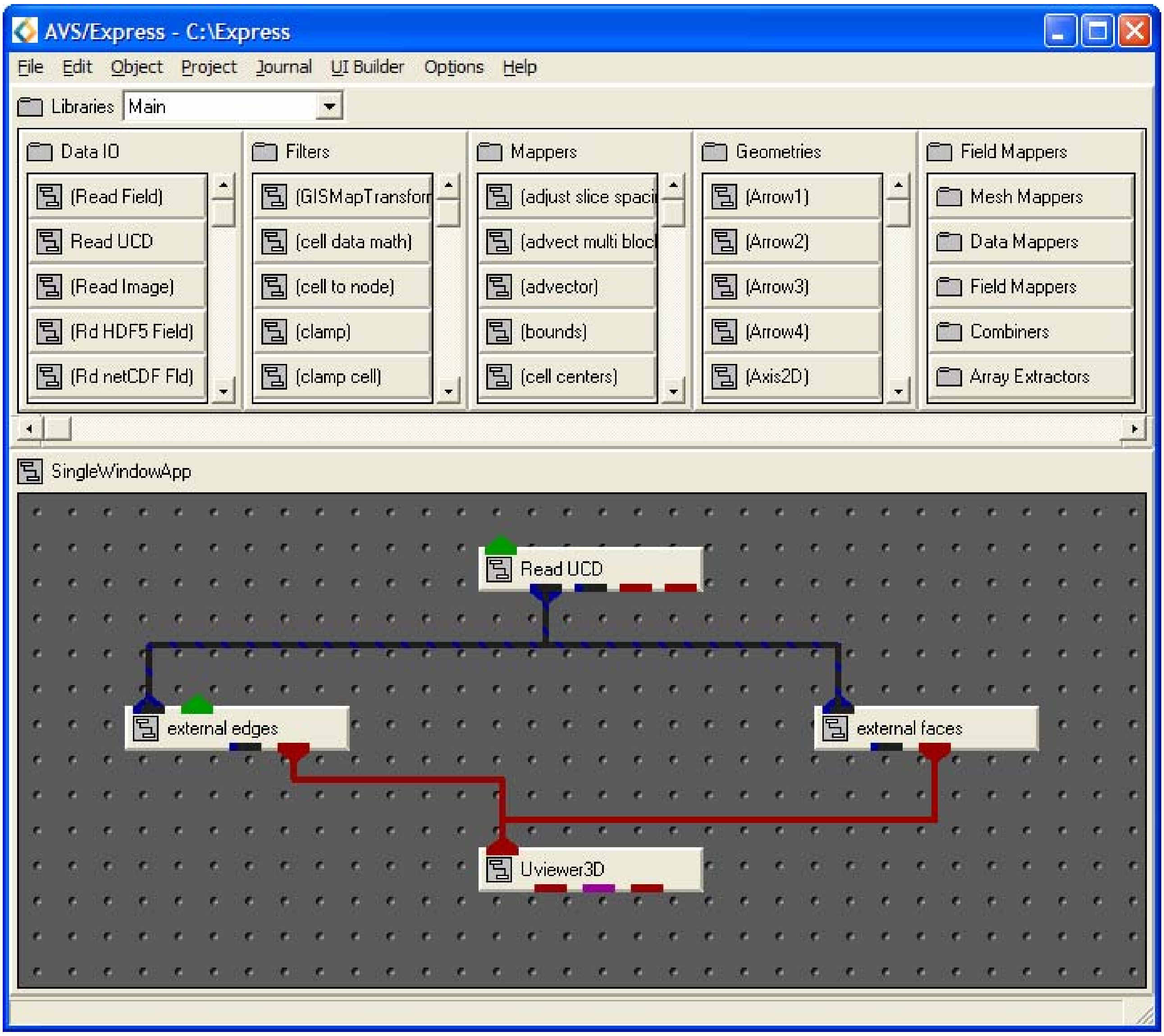

Figure 5.3: Extended Network for UCD Data

Create the network shown in Figure 5.3 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 wire frame representation of the outside of the unstructured mesh to reveal objects inside
- `external_faces` (Mappers): produce a mesh which represents the exterior, visible faces of an unstructured mesh.
- `UViewer3D` (Viewers): creates a 3D image in the Data Viewer.

5.3.4 Displaying the UCD External Edges and Faces (2)

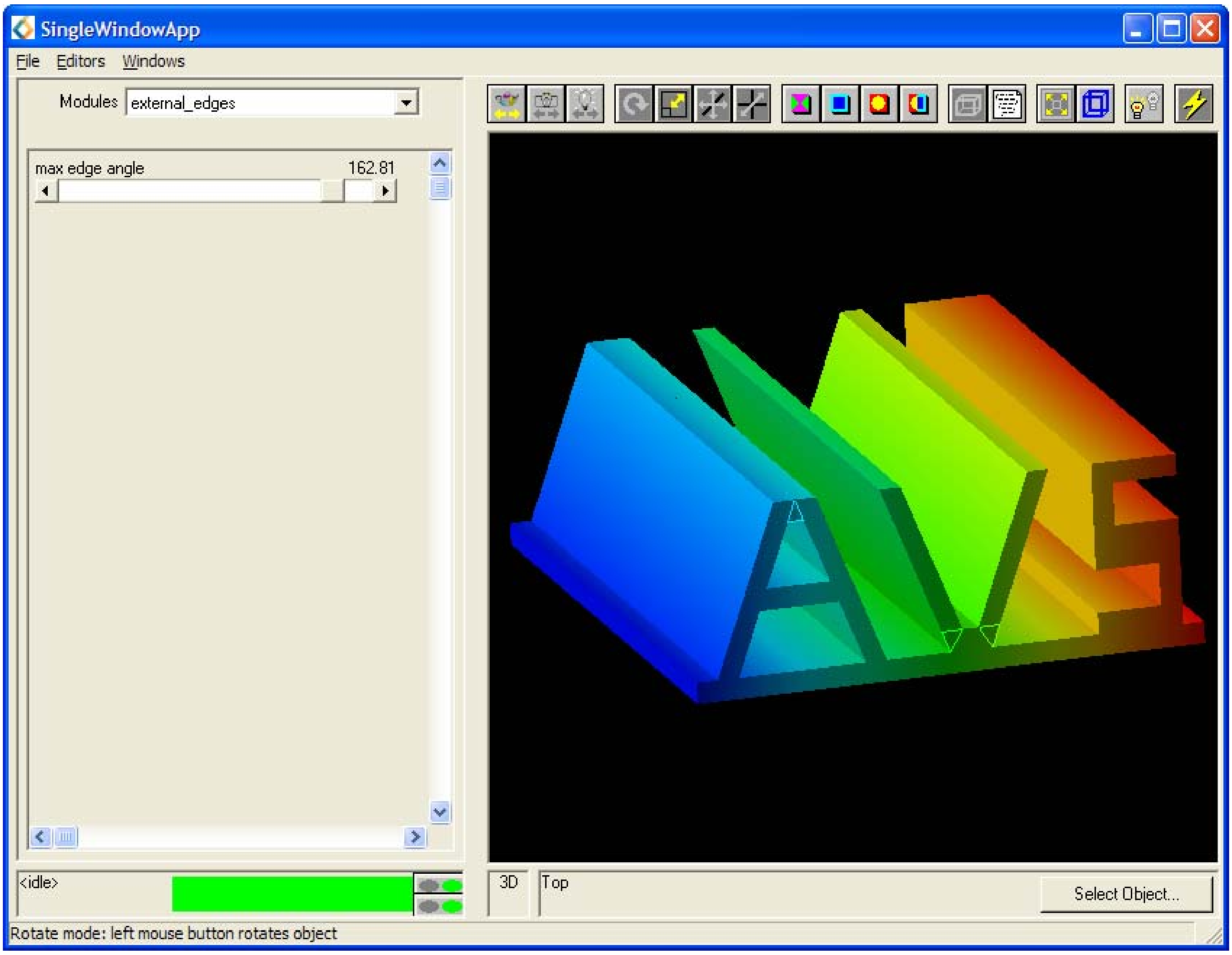

Figure 5.4: `external_edges` Editor Panel and Result

- Select the `avs.inp` UCD file using the file browser on the `Read_UCD` module.
- Select the `external_edges` modules Editor Panel.
- Alter the **max edge angle** and observe the effect.

5.3.5 Shrinking the UCD Cells (1)

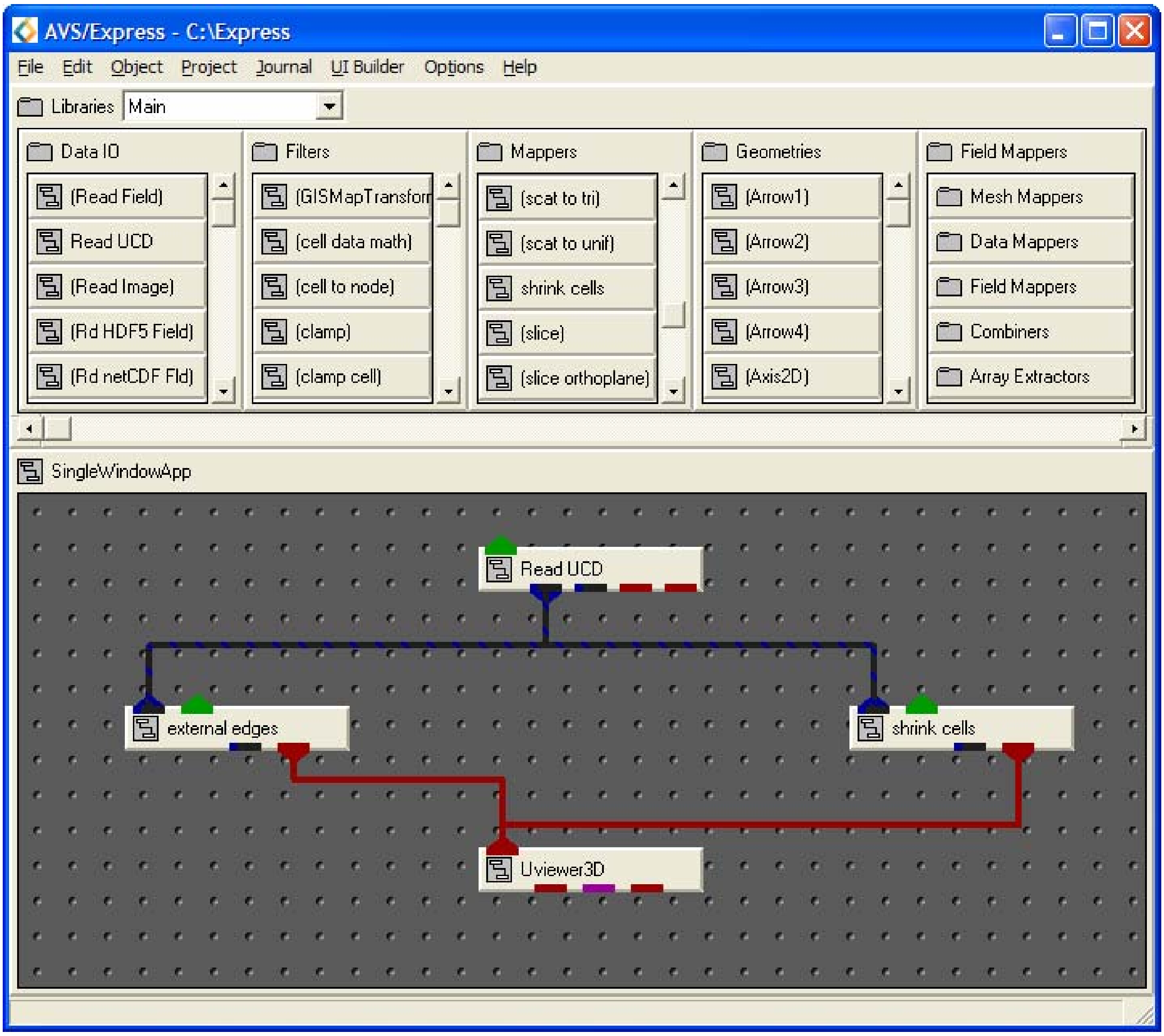

Figure 5.5: Alternative Network for UCD Data

Create the network shown in Figure 5.5 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 wire frame representation of the outside of the unstructured mesh to reveal objects inside
- `shrink_cells` (Mappers): produce a mesh with cells shrunk relative to their geometric centres.
- `UViewer3D` (Viewers): creates a 3D image in the Data Viewer.

5.3.6 Shrinking the UCD Cells (2)

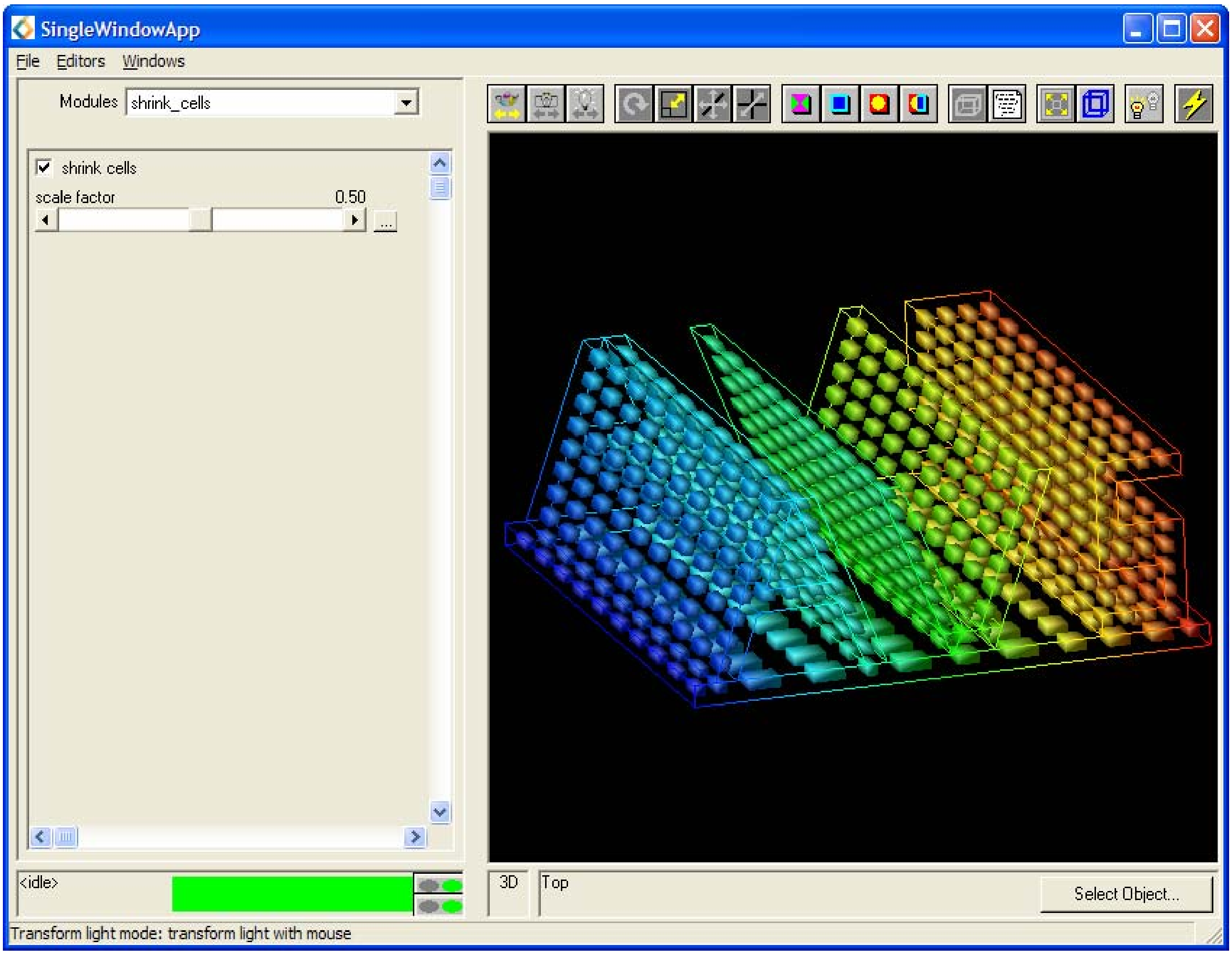

Figure 5.6: `shrink_cells` Editor Panel and its Result

- Select the `avs.inp` UCD file using the file browser on the `Read_UCD` module.
- Select the `shrink_cells` Editor Panel
- Alter the **scale factor** and observe the effect.

5.4 Further Exercises

Now that the basic UCD examples have been covered, it is recommended that the networks illustrated are used to display some alternative UCD files. Below is a list of several UCD files that are available, they can 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

This exercise contains several different networks which are able to display the bluntfin data as a UCD.

5.5.1 Displaying Scalar Data using the `contour` Module (1)

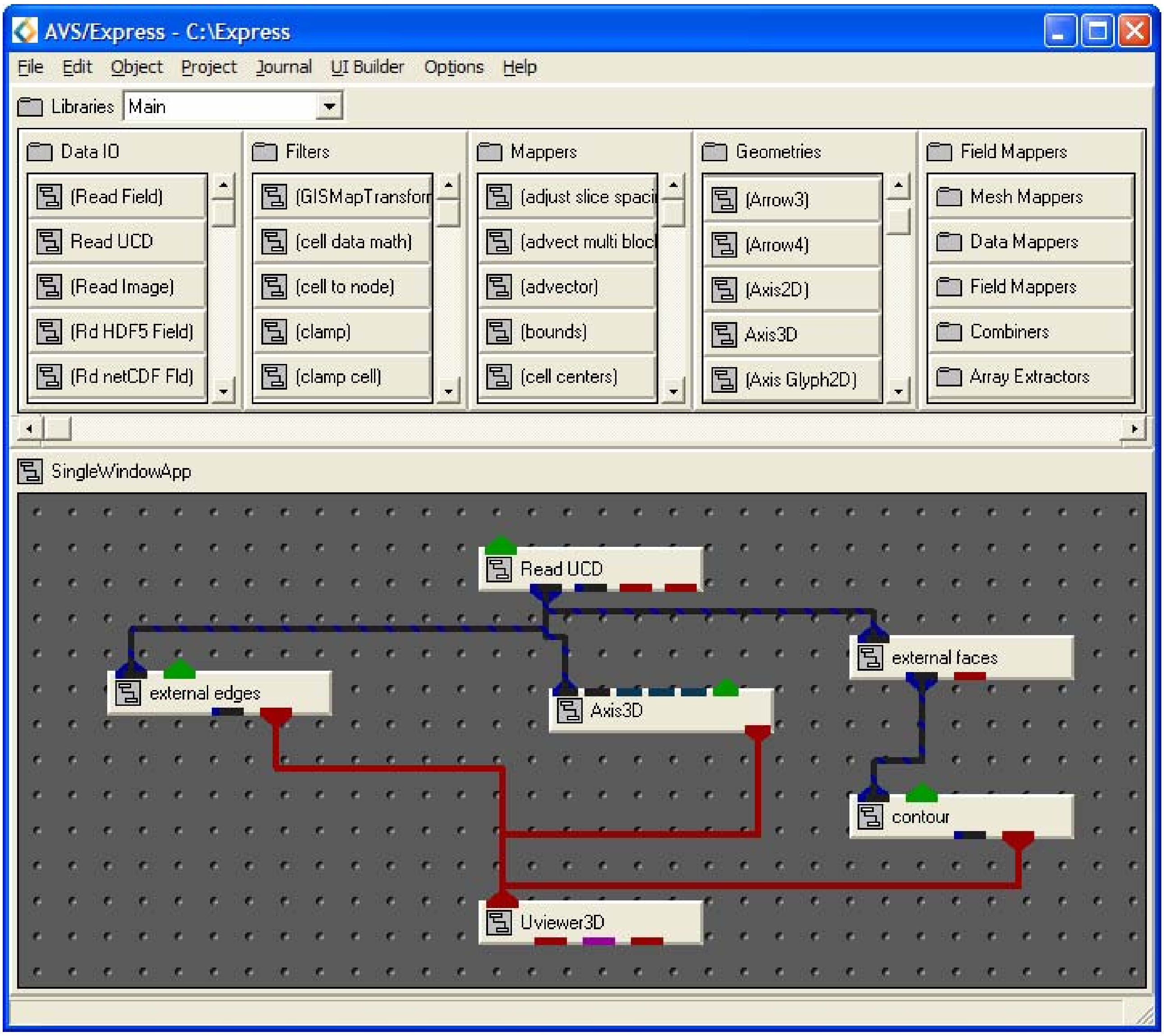

Figure 5.7: Network to Display Scalar Data

Create the network shown in Figure 5.7 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 wire frame representation of the outside of the unstructured mesh to reveal objects inside
- `external_faces` (Mappers): produce 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 Data Viewer.

5.5.2 Displaying Scalar Data using `contour` Module (2)

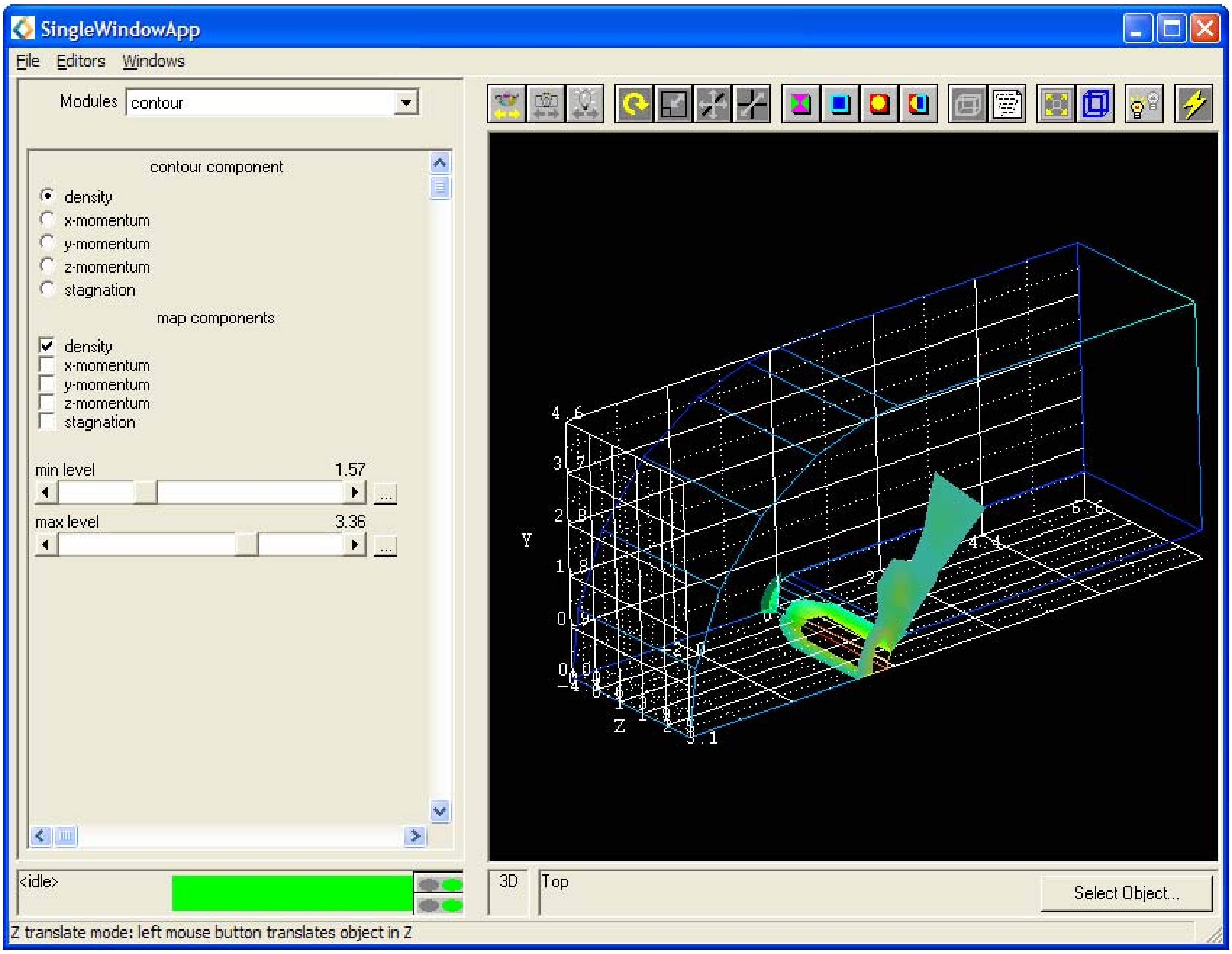

Figure 5.8: contour Editor Panel and its Result

- Select the file `bluntfin.inp` using the `Read_UCD` modules file browser.
- Using the mouse rotate the object so that similar to that in Figure 5.8.
- Select the `contour` modules Editor Panel
- Check the **density** data component option box
- Alter the **max** and **min** values to be contoured and observe the effect.

5.5.3 Using the `isoline` Module (1)

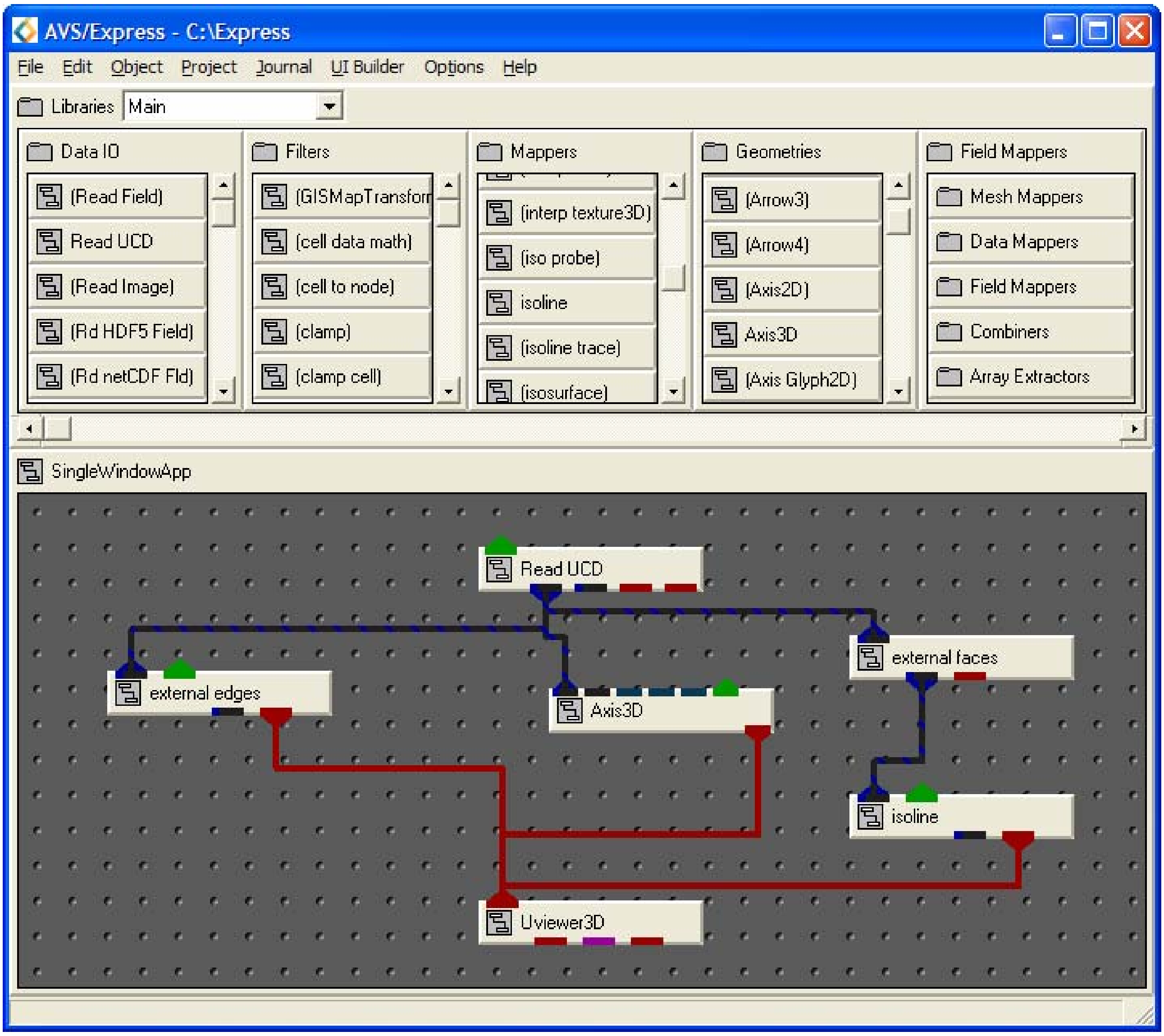

Figure 5.9: Network containing an `isoline` Module

- `Read_UCD` (Data IO): reads an AVS UCD (`.inp`) file and outputs an AVS/Express field.
- `external_edges` (Mappers): produces a wire frame representation of the outside of the unstructured mesh to reveal objects inside
- `external_faces` (Mappers): produce 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 Data Viewer

5.5.4 Using the `isoline` Module (2)

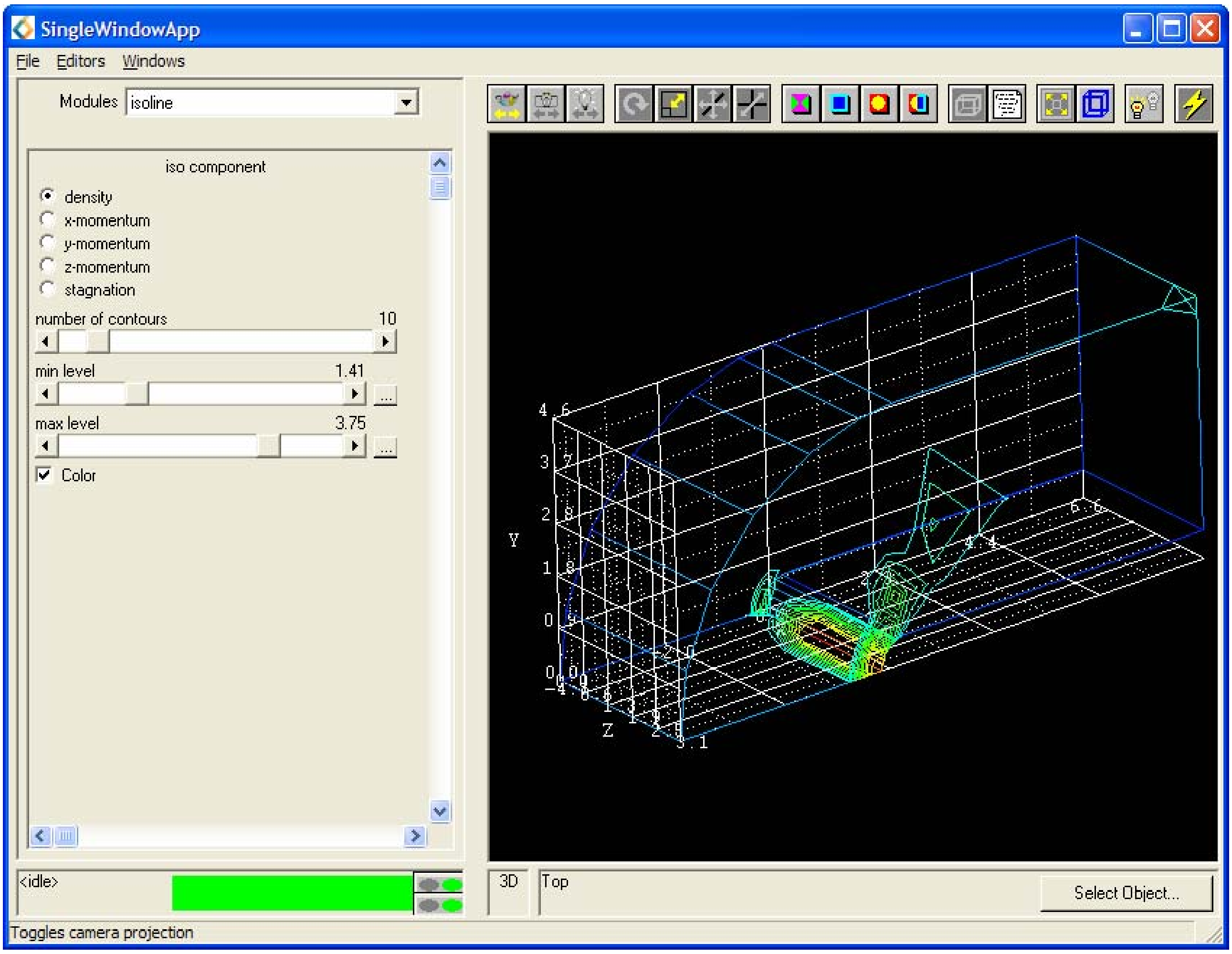

Figure 5.10: `isoline` Editor Panel and its Result

- Select the file `bluntfin.inp` using the `Read_UCD` modules file browser.
- Select the `isoline` modules Editor Panel
- Check the data component **density** option box.
- An image similar to that in Figure 5.10 should be obtained.
- Alter the **min** and **max** data values to be contoured and observe the effect.

5.5.5 Displaying Vector Data (1)

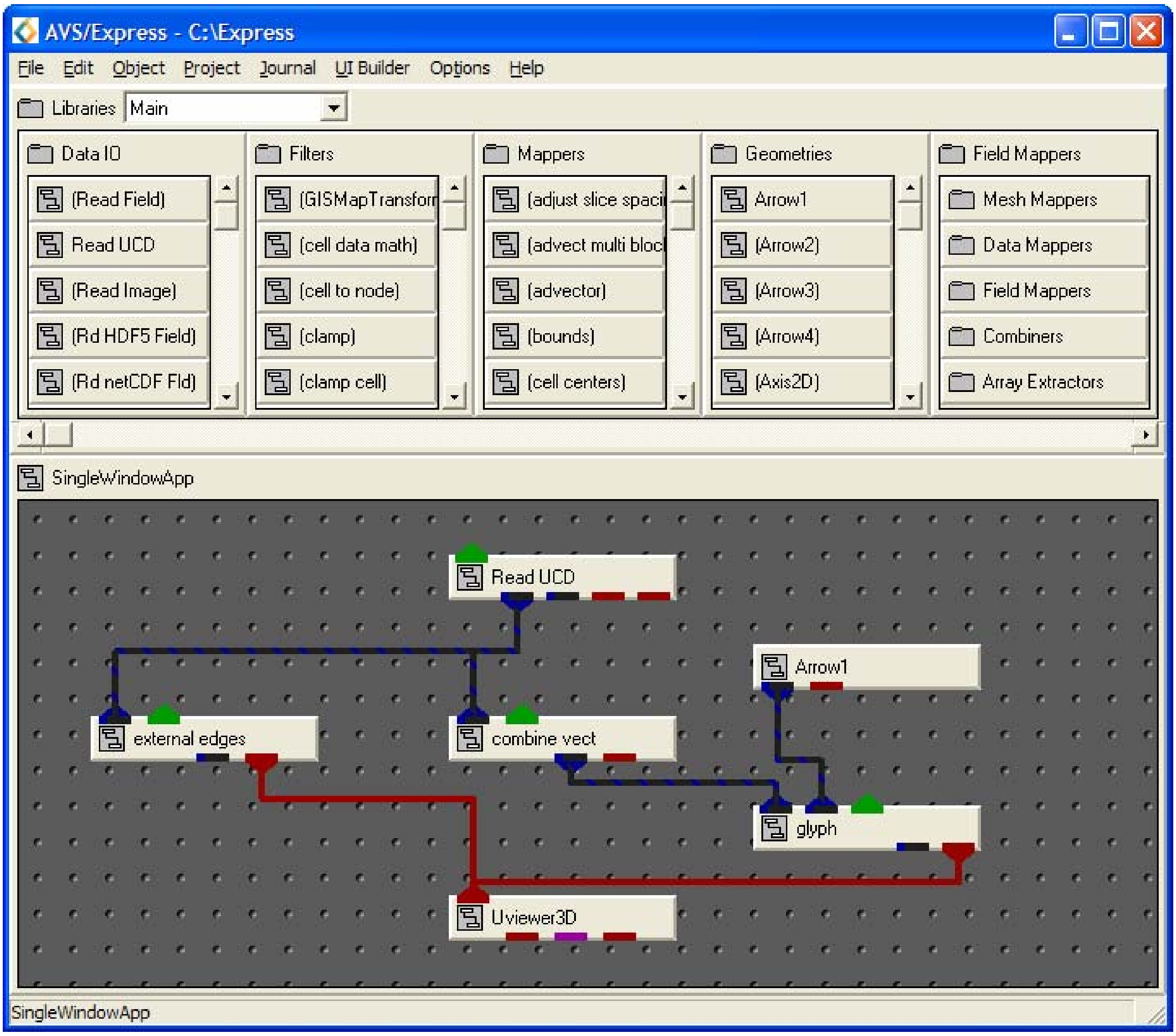

Figure 5.11: Network to Visualise the Vector Data

Create the network shown in Figure 5.11 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 a wire frame representation of the outside of the 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 Data Viewer

5.5.6 Displaying Vector Data (2)

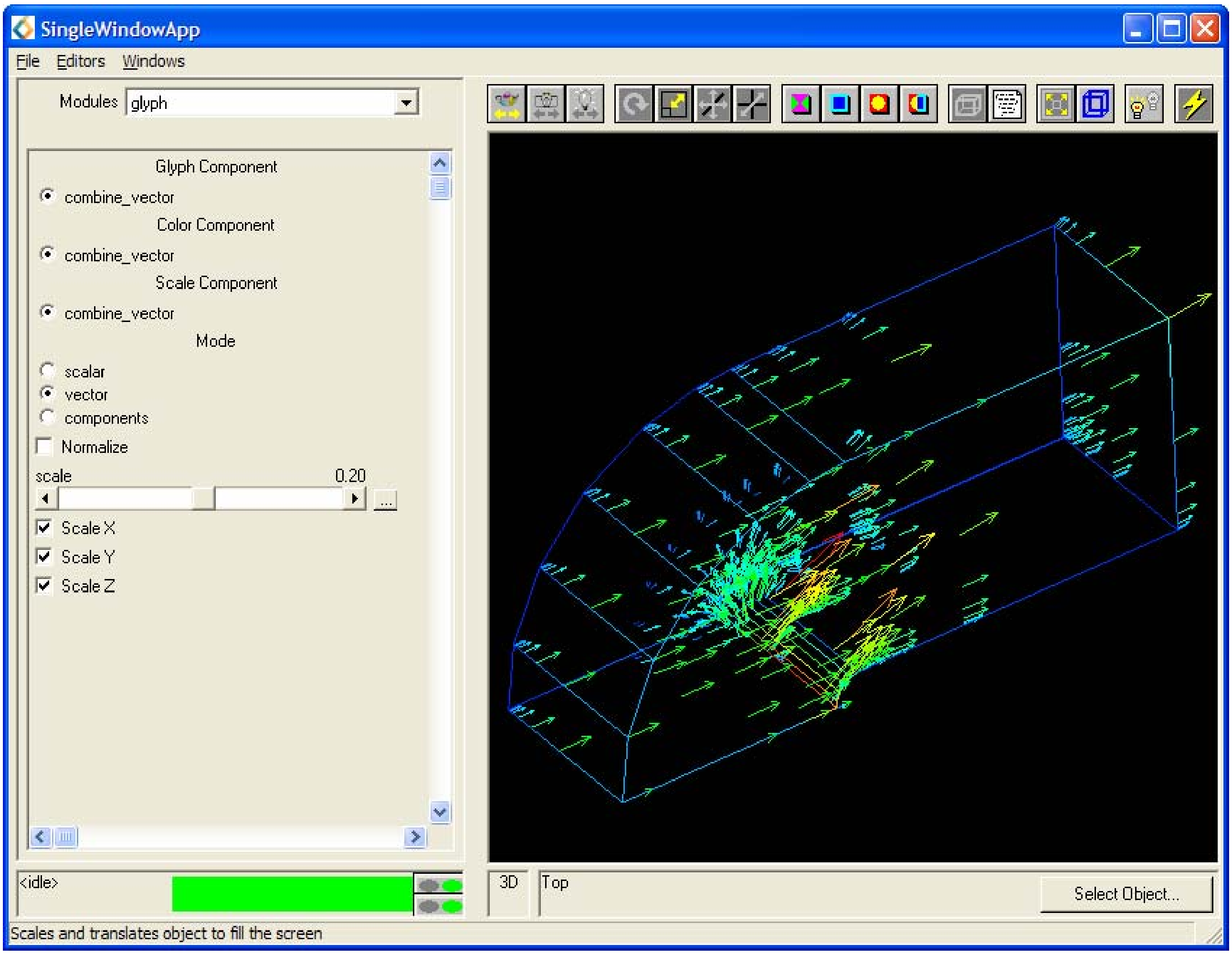

Figure 5.12: `glyph` Editor Panel and its Result

- Select the file `bluefin.inp` using the `Read_UCD` modules file browser.
- Select the `combine_vect` module and check the **x-momentum**, **y-momentum** and **z-momentum** option boxes.
- Select the `glyph` module and reduce the **scale** to ~ 0.2 or until the arrows are suitably sized.
- The image obtained should be similar to that shown in Figure 5.12.

5.5.7 Using the `streamlines` Module (1)

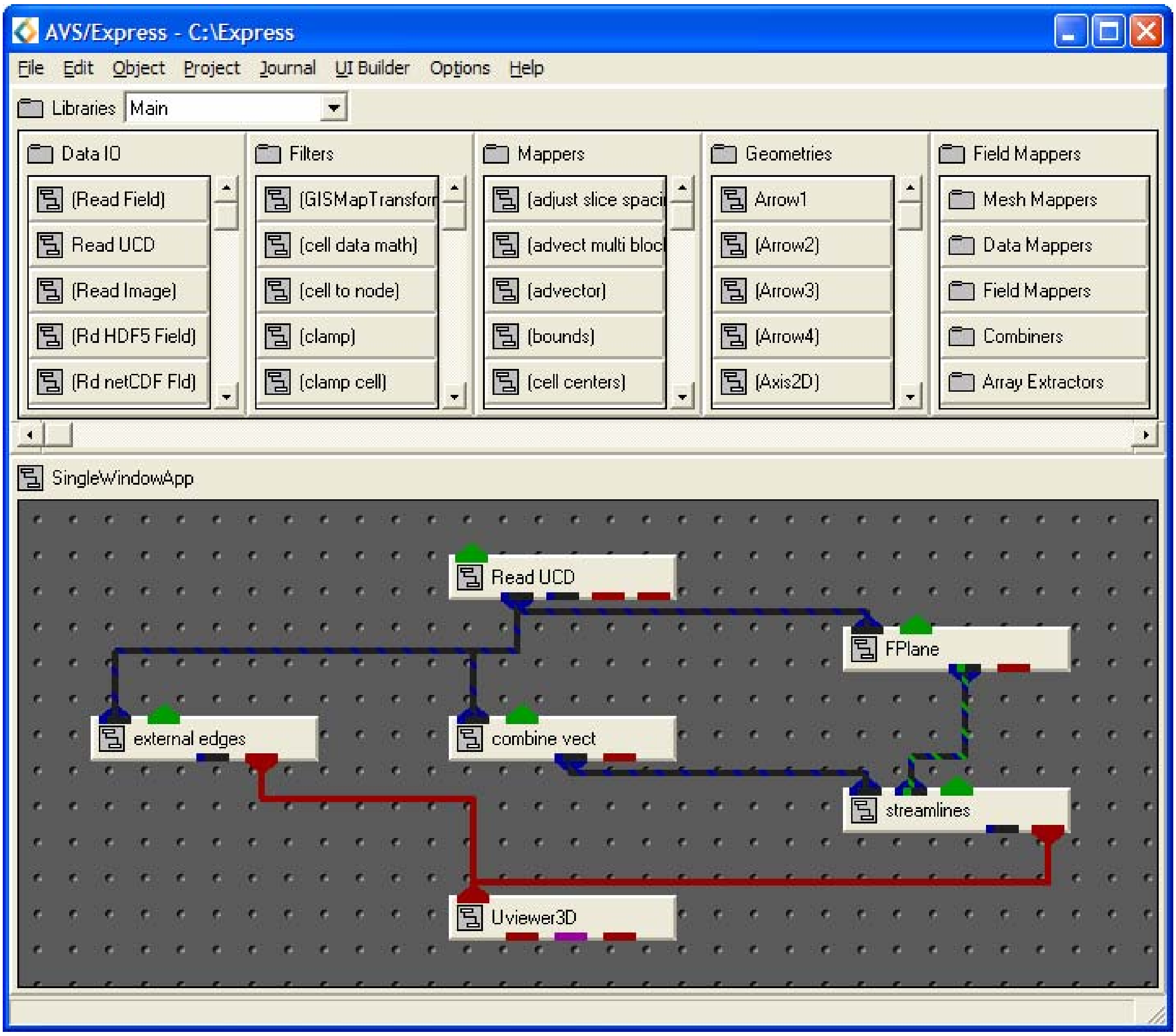

Figure 5.13: Network using the `streamline` Module to Visualize Vector Data

Create the network shown in Figure 5.13 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 wire frame representation of the outside of the unstructured mesh to reveal objects inside
- `FPlane` (Geometries): generates a 3D plane of variable size which can be transformed in three dimensions.
- `combine_vect` (Geometries): 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 one, two or three element vector components.
- `UViewer3D` (Viewers): creates a 3D image in the Data Viewer

5.5.8 Using the `streamlines` Module (2)

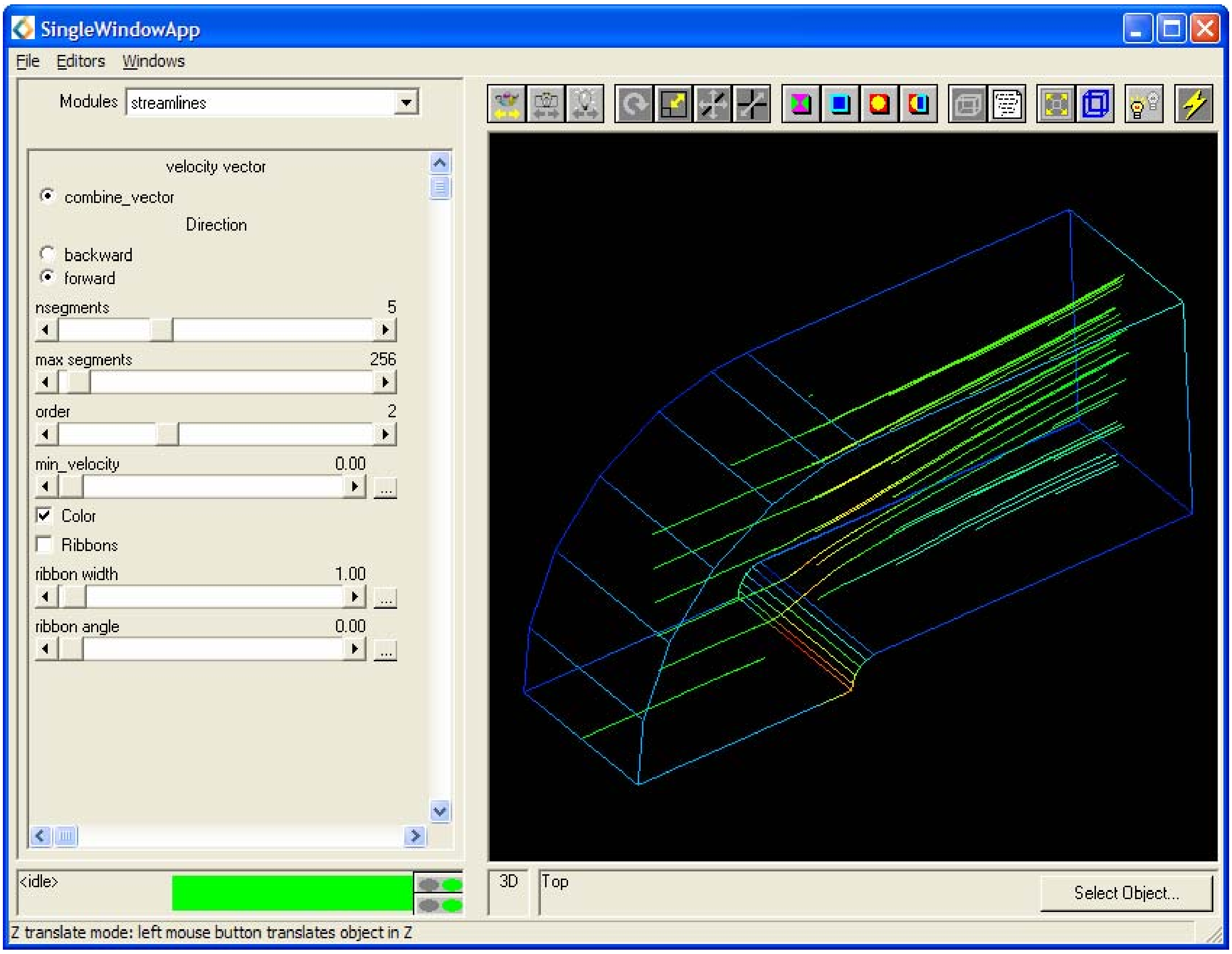

Figure 5.14: `streamlines` Editor Panel and its Result

- Select the file `bluntfin.inp` using the `Read_UCD` modules file browser.
- Select the `combine_vect` module and ensure that **x**, **y** and **z-momentum** are selected.
- Select the `streamlines` modules Editor Panel.
- Increase the **nsegment** value to increase the number of streamlines.
- Investigate the effect of changing the **min_velocity** value.
- Change the streamlines to streamribbons by selecting the **Ribbons** option box.
- Alter the `FPlane` module parameters.

6 AVS/Express Examples

6.1 Aims of the Chapter

There are numerous on-line examples included within AVS/Express in order to illustrate the operation and applications of most of the AVS/Express modules. This chapter will investigate an example network.

6.2 Exercise 1

This exercise will illustrate the use of these on-line examples using the `advector` module as a demonstration. The `advector` module releases a sample of massless particles into a field. The particles move through the field according to the magnitude and direction of the vectors at the nodes. The `advector` module is used to demonstrate fluid flow.

6.2.1 Entering the AVS/Express Examples Library

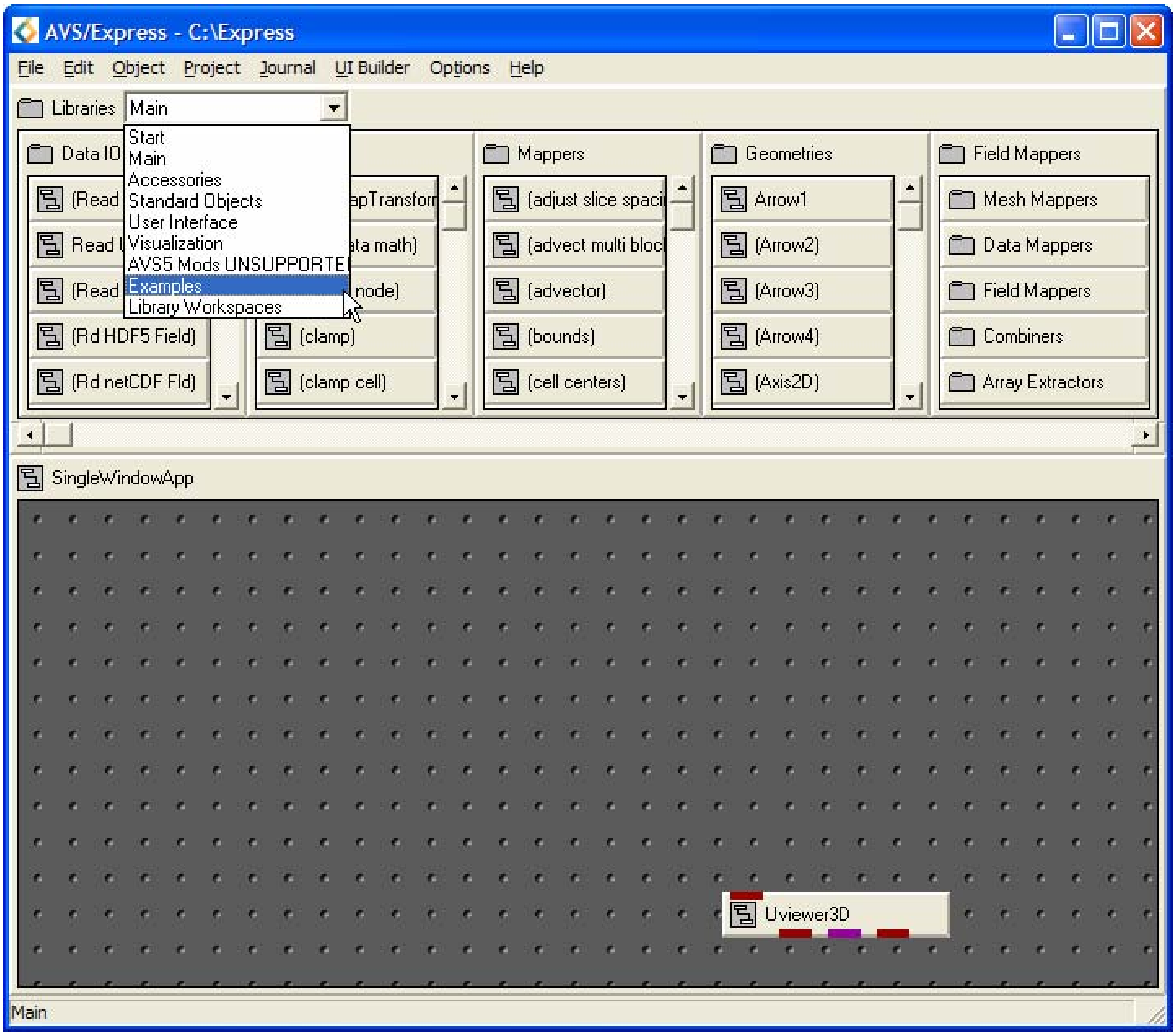

Figure 6.1: **Examples** Library in the **Libraries** Menu

To access to the AVS/Express on-line examples

- Select the **Examples** library page from the **Libraries** dropdown list on the top of the Network Editor window.

6.2.2 Starting the `advect` Demo

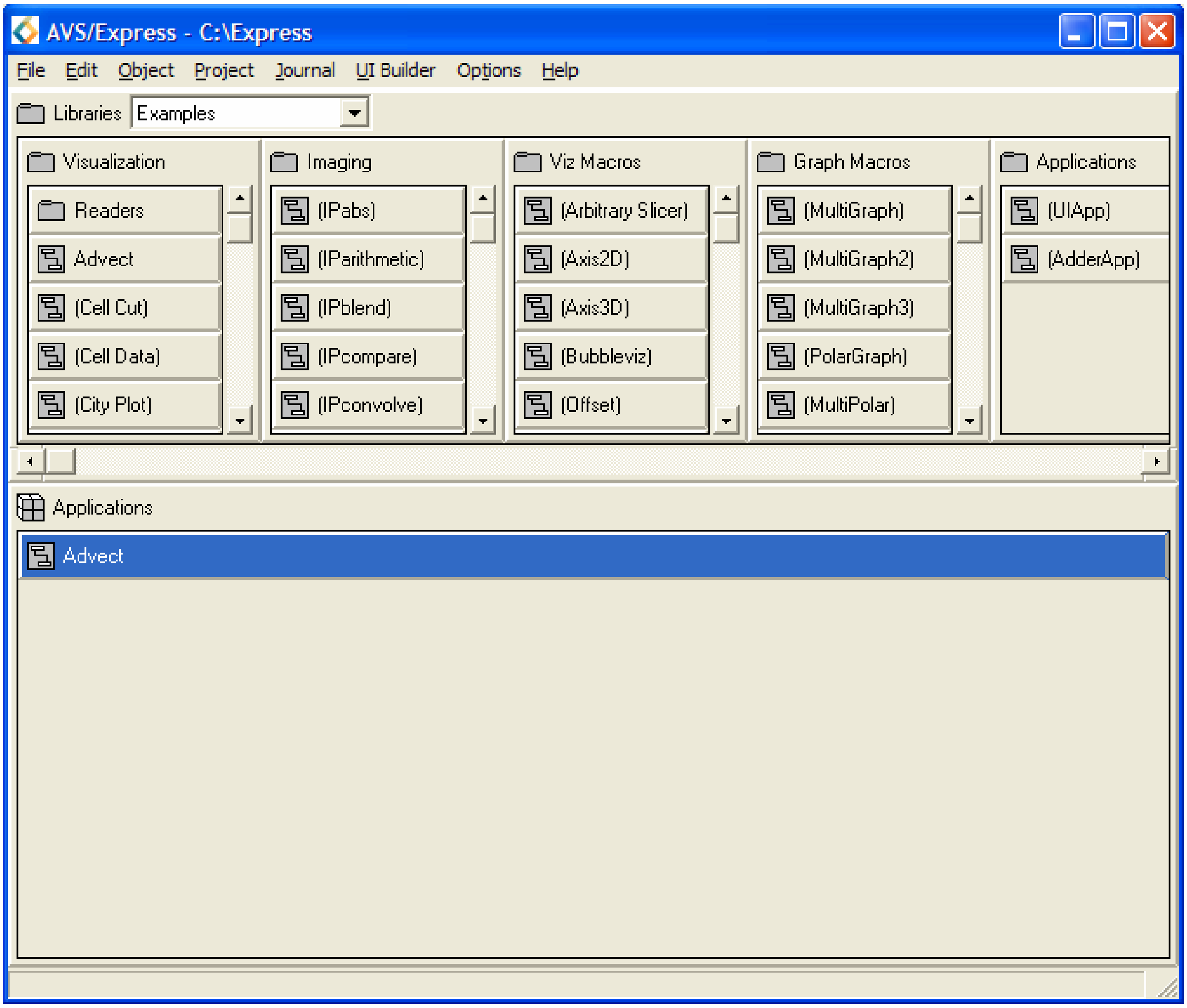

Figure 6.2: `Advect` Instanced in Application Workspace

- Double click the **Visualization** sub-library to open it. (It may already be open.)
- Close or delete all current applications, then drag and drop the `Advect` demo module into the workspace.
- The screen should appear as shown above.

6.2.3 The advect Demo

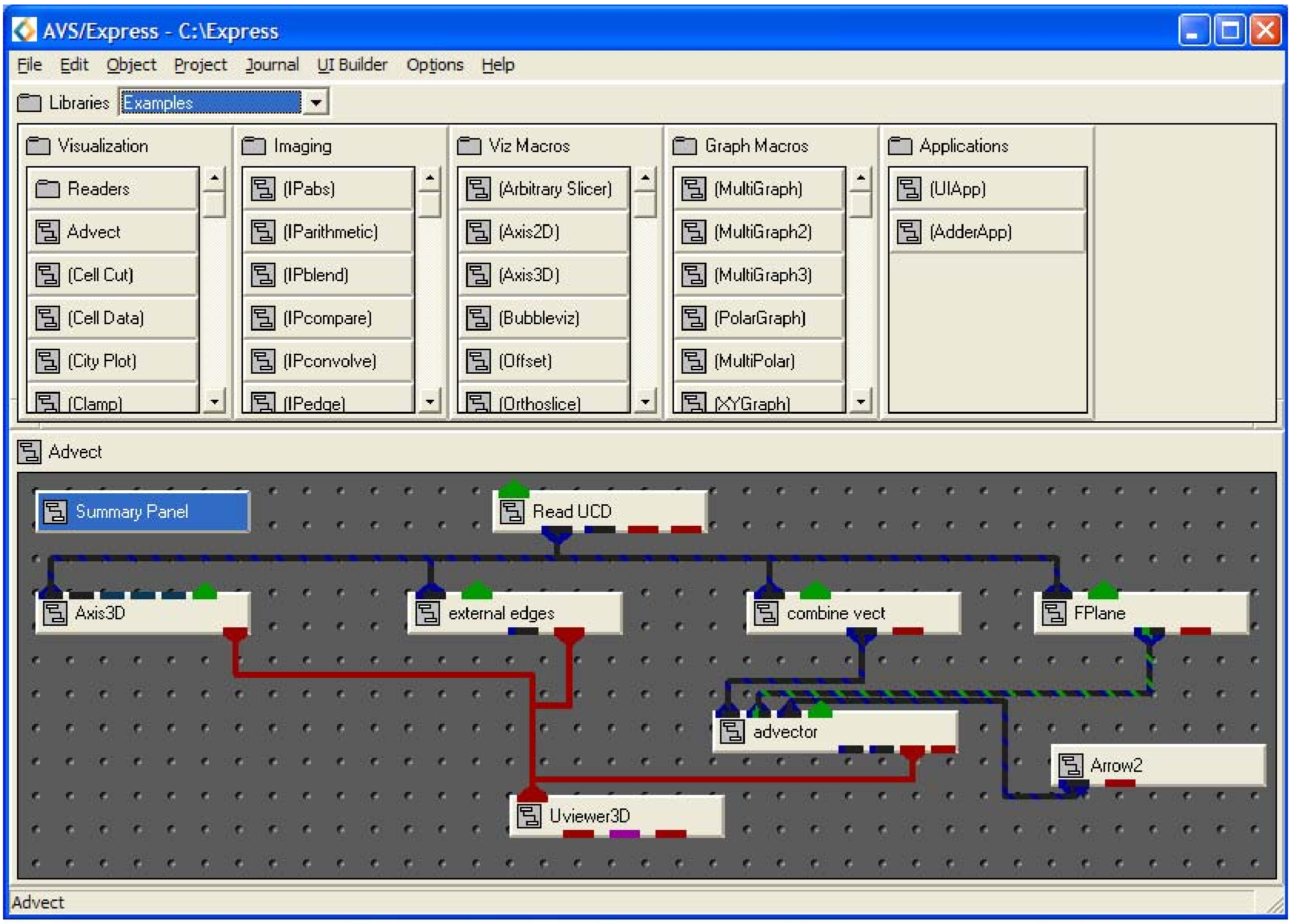

Figure 6.3: Advect Network

- Double click the `Advect` icon in the workspace in order to maximise it. The complete `Advect` network will be shown in the workspace.
- Refer to each of the modules help documentation in order to understand the way the network operates. Understanding the `advector` module is especially important.
- Rearrange the network so it appears similar to the layout shown above. Select the **Main** library page in order to select any new modules.

6.2.4 Changing the Parameters

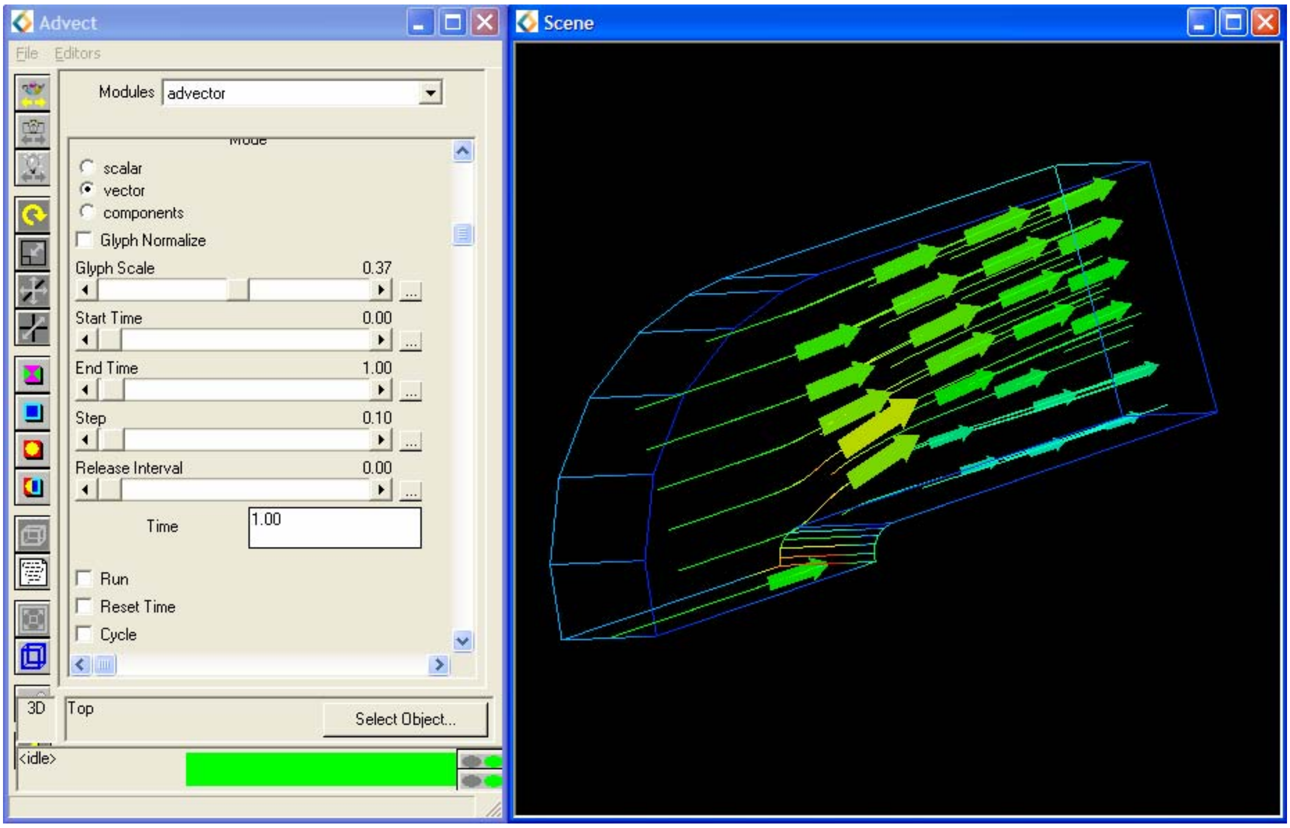

Figure 6.4: `advector` Editor Panel

- Select the `advector` module from the **Modules** dropdown menu.

The following steps will run the `Advect` demonstration:

- Before releasing the advectors select the **Reset Time** option box.
- To release the advectors select the **Run** option box.
- Select the **cycle** option box in order to run advectors on a continuous loop. To stop the cycle, unselect the **Run** option box.

Recommended extensions:

- Alter the size of the glyphs by changing the **Glyph scale** value.
- Alter other parameters in the `advector` modules control panel in order to observe the effects they have on the demonstration.

6.3 Other Exercises

Carry out further investigation of the other examples in the **Examples** library. Use the module documentation in order to help understanding in regards to their functions.

Appendix A:

A.1: Obtaining Hardcopies from AVS/Express

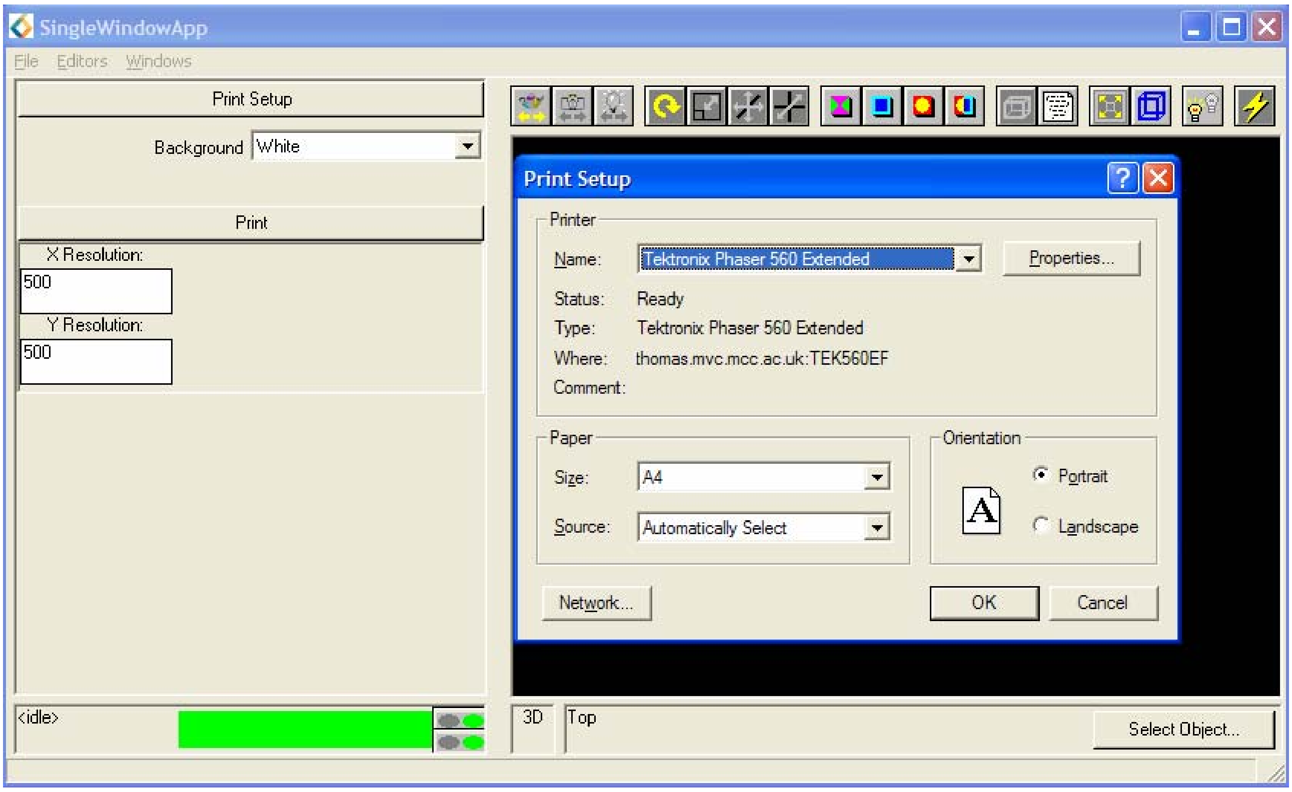

Figure A.1: **Print editor** and **Print Setup** Window

The contents of the **Viewer** window can be printed using the **Print Editor**

- Select **Print** from the **Editors** menu:
- Alter the parameters to reflect those shown above.
- Print Setup can be displayed by selecting the **Print Setup** button within the editor. Options such as **Orientation** and **Printer** selection are available here for alteration.
- Select the **Print** button in the editor in order to print the contents of the viewer.

A.2: Journaling in AVS/Express

Journaling allows users to record a set of operations, performed on an object and replay them. AVS/Express records these operations as V statements and commands within a journal file.

To record a series of Network Editor operations:

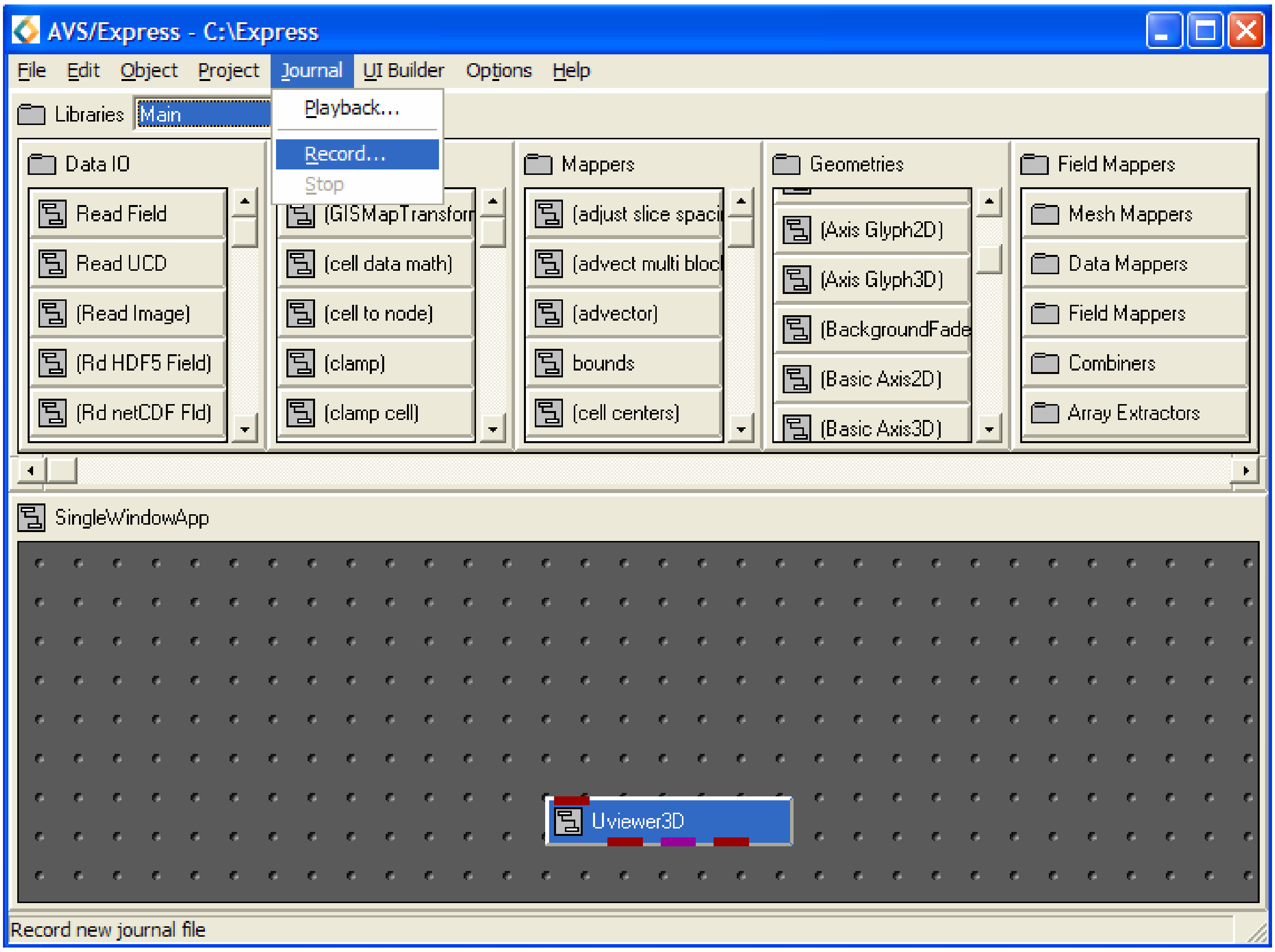

Figure A.2: **Journal** menu

- From the **Journal** menu on the Network Editor menu bar select **Record**. The **Record File** pop up browser window will be displayed as shown in Figure A.3.

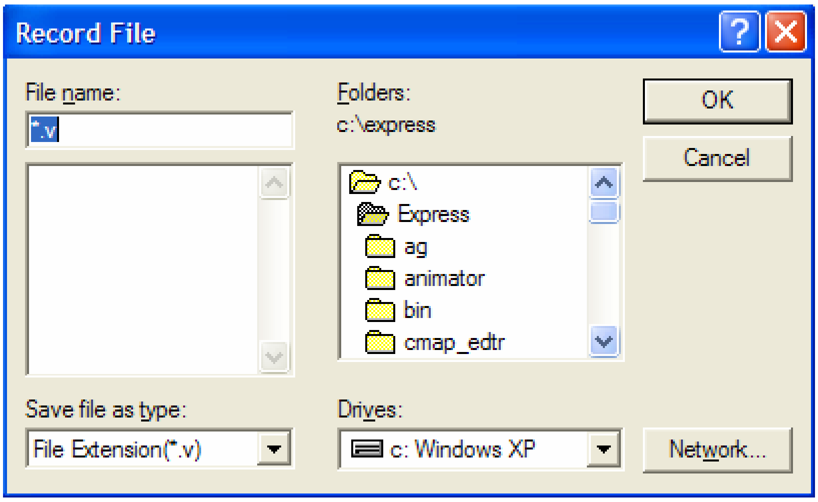

Figure A.3: **Record File** Pop up Window

- Specify a journal filename.
- Select **OK** to begin recording. Perform the series of operations, e.g. build a network, set values and manipulate the object. (Any action preformed outside the workspace will be ignored.)
- To stop recording select the **Stop** command from the **Journal** menu. The recording will stop and the resulting V file will be saved.

To play back the V file:

- Ensure that the recording has been stopped and that the network is in the same state as it was when the recording began.
- Select the **Playback** option from the **Journal** menu, Figure A.2. The **Playback File** pop up browser window will be displayed.
- Select the journal file and select **OK**. The recording will be executed and cannot be stopped part way through.

Appendix B: Field Files

B.1: Solutions to Chapter 3

B.1.1: Field Descriptor for Exercise 1

```
#   AVS field file
#   Image
ndim=2
dim1=512
dim2=256
nspace=2
veclen=1
data=integer
field=uniform
variable 1 file=image.dat filetype=ascii skip=0 offset=0 stride=1
```

B.1.2: Field Descriptor for Exercise 2

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

B.1.3: Field Descriptor for Exercise 3

```
#   AVS field file
# Flow
ndim=2
dim1=5
dim2=10
nspace=2
veclen=2
data=float
field=irregular
label=temp press
units=k kg/m**2
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 skip0 offset=1 stride=2
```

B.2: Solutions to Chapter 3

B.2.1: Field Descriptor for Exercise 1

```
#   AVS field file
#   MRI
ndim=2
dim1=512
dim2=512
nspace=2
veclen=1
data=byte
field=uniform
variable 1 file=mri.dat filetype=binary skip=0 stride=1
```

B.2.2: Field Descriptor for Exercise 2

```
#   AVS field file
#   Temp
ndim=2
dim1=31
dim2=31
nspace=2
veclen=1
data=float
field=uniform
variable 1 file=temps.dat filetype=ascii skip=1 stride=1
```

B.2.3: Field Descriptor for Exercise 3

```
#   AVS field file
#   Hydrogen
ndim=2
dim1=64
dim2=64
nspace=2
veclen=1
data=byte
field=uniform
variable 1 file=hslice.dat filetype=binary skip=0 stride=1
```

B.3: Solutions to Chapter 4

B.3.1: Field Descriptor for Exercise 1

```
#   AVS field file
#   Bluntfin
ndim=3
dim1=10
dim2=8
dim3=8
nspace=3
veclen=5
data=float
field=irregular
label = density x-momentum y-momentum z-momentum stagnation
variable 1 file=fin.dat filetype=ascii skip=0 offset=0 stride=8
variable 2 file=fin.dat filetype=ascii skip=0 offset=1 stride=8
variable 3 file=fin.dat filetype=ascii skip=0 offset=2 stride=8
variable 4 file=fin.dat filetype=ascii skip=0 offset=3 stride=8
variable 5 file=fin.dat filetype=ascii skip=0 offset=4 stride=8
coord 1 file=fin.dat filetype=ascii skip=0 offset=5 stride=8
coord 2 file=fin.dat filetype=ascii skip=0 offset=6 stride=8
coord 3 file=fin.dat filetype=ascii skip=0 offset=7 stride=8
```

B.3.2: Field Descriptor for Exercise 2

B.3.2.1: Scalar Data

```
#   AVS field file
#   Climate Modelling – scalar data
ndim=3
dim1=49
dim2=12
dim3=40
nspace=3
veclen=2
data=float
field=irregular
label = temperature Humidity
variable 1 file=temp.dat filetype=binary skip=16 offset=0 stride=1
variable 2 file=humid.dat filetype=binary skip=16 offset=1 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 3 file=ccm1.sph filetype=ascii skip=5881 offset=0 stride=1
```

B.3.2.2: Vector Data

```
#   AVS field file
#   Climate Modelling – vector data
ndim=3
dim1=49
dim2=12
dim3=40
nspace=3
veclen=3
data=float
field=irregular
label = u-comp v-comp w-comp
variable 1 file=bi.u.49 filetype=binary skip=16 offset=0 stride=1
variable 2 file=bi.v.49 filetype=binary skip=16 offset=0 stride=1
variable 3 file=bi.w.49 filetype=binary skip=16 offset=0 stride=1
coord 1 file=ccm1.rect filetype=ascii skip=1 offset=0 stride=1
coord 2 file=ccm1.rect filetype=ascii skip=2941 offset=0 stride=1
coord 3 file=ccm1.rect filetype=ascii skip=5881 offset=0 stride=1
```

B.3.2.3: World Map

```
#   AVS field file
#   World Map
ndim=3
dim1=49
dim2=12
dim3=40
nspace=3
veclen=1
data=float
field=irregular
variable 1 file=world.asc filetype=ascii skip=1 offset=0 stride=1
coord 1 file=ccm1.rect filetype=ascii skip=1 offset=0 stride=1
coord 2 file=ccm1.rect filetype=ascii skip=2941 offset=0 stride=1
coord 3 file=ccm1.rect filetype=ascii skip=5881 offset=0 stride=1
```

(Hint: For coordinates there are 49x12x40 values with 8 values on each line.)

Appendix C: Using the Wizards

C.1: Aims of this Appendix

The exercises in this appendix will illustrate the purpose of and way in which the Wizards can be employed. Exercise 1 is a simple example designed to illustrate the fundamental idea behind Wizards. Exercise 2 is a similar example to be attempted without guidance. Exercise 3 is a more advanced example designed to illustrate how data of a type unknown to AVS/Express can be imported using Wizards. In all cases, it is unnecessary to create a field file in order to read the data. The creation of field files was covered in Chapter 2.

C.2: Exercise 1

In this example the electron density within a hydrogen atom is the subject of the visualization. The aim is to take a slice through the atom and observe differing electron densities throughout the atom.

There are two Wizards within AVS/Express, designed to help new AVS/Express users to get started, by guiding them through the application process. These two Wizards, the `data_import_wizard` and the `data_visualization_wizard` will be employed in order to create the network without any need to interface with the Network Editor directly.

C.2.1: Starting the AVS/Express Visualization Edition

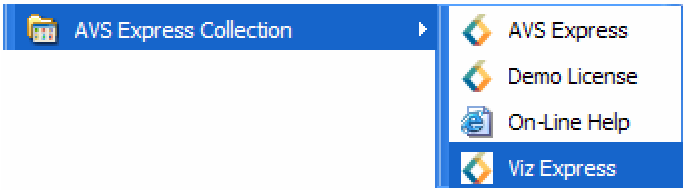

Figure C.1: Start Menu

- To start AVS/Express Visualization Edition, select AVS Express Collection folder from the Start menu and load the **Viz Express** application (Figure C.1)
- At the start-up of AVS/Express the Network Editor will be in the **Start** library and the `data_import_wizard` and `data_visualization_wizard` will be instanced in the working area (Figure C.2).

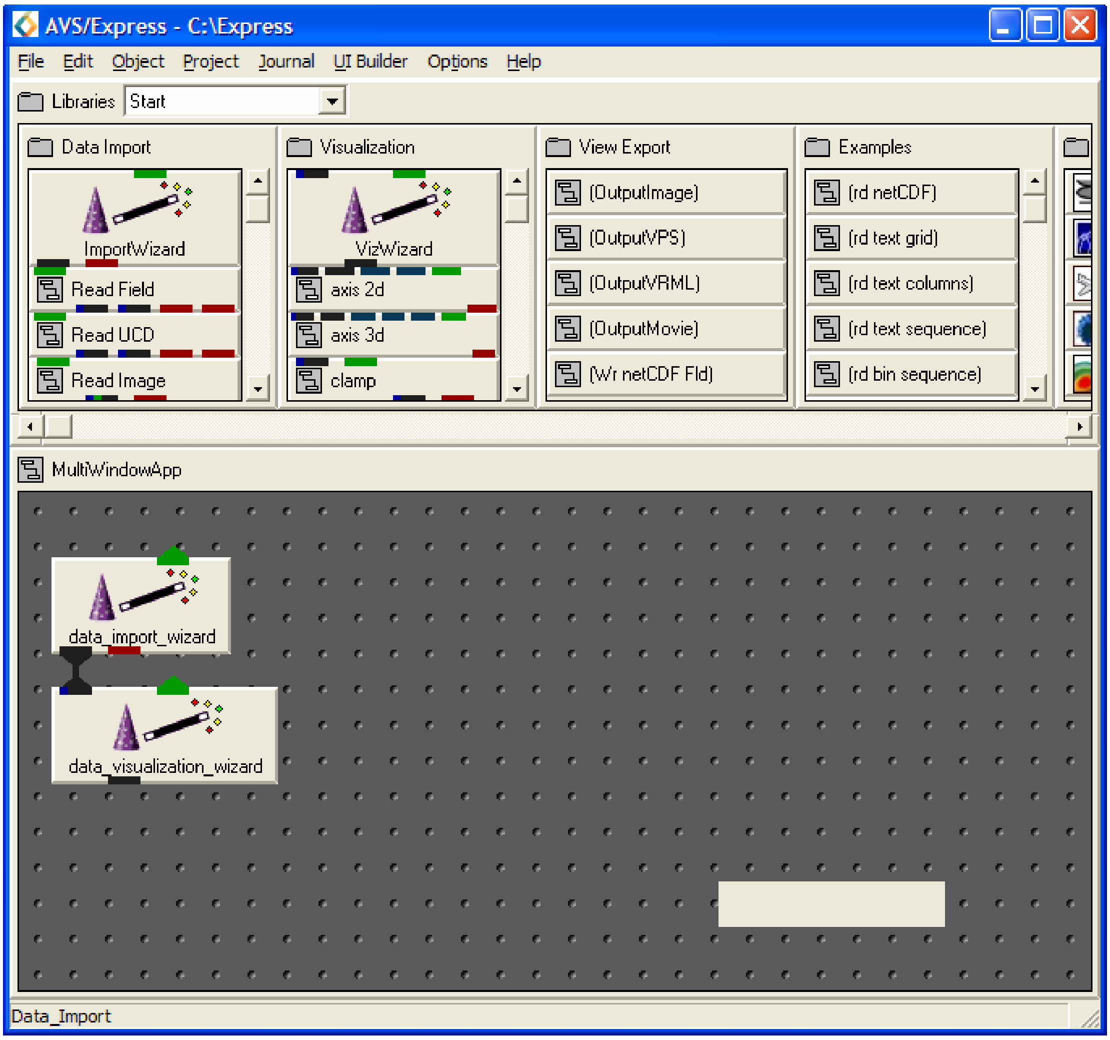

Figure C.2: Network Editor in the **Start** Library

C.2.2: Data Viewer Pad

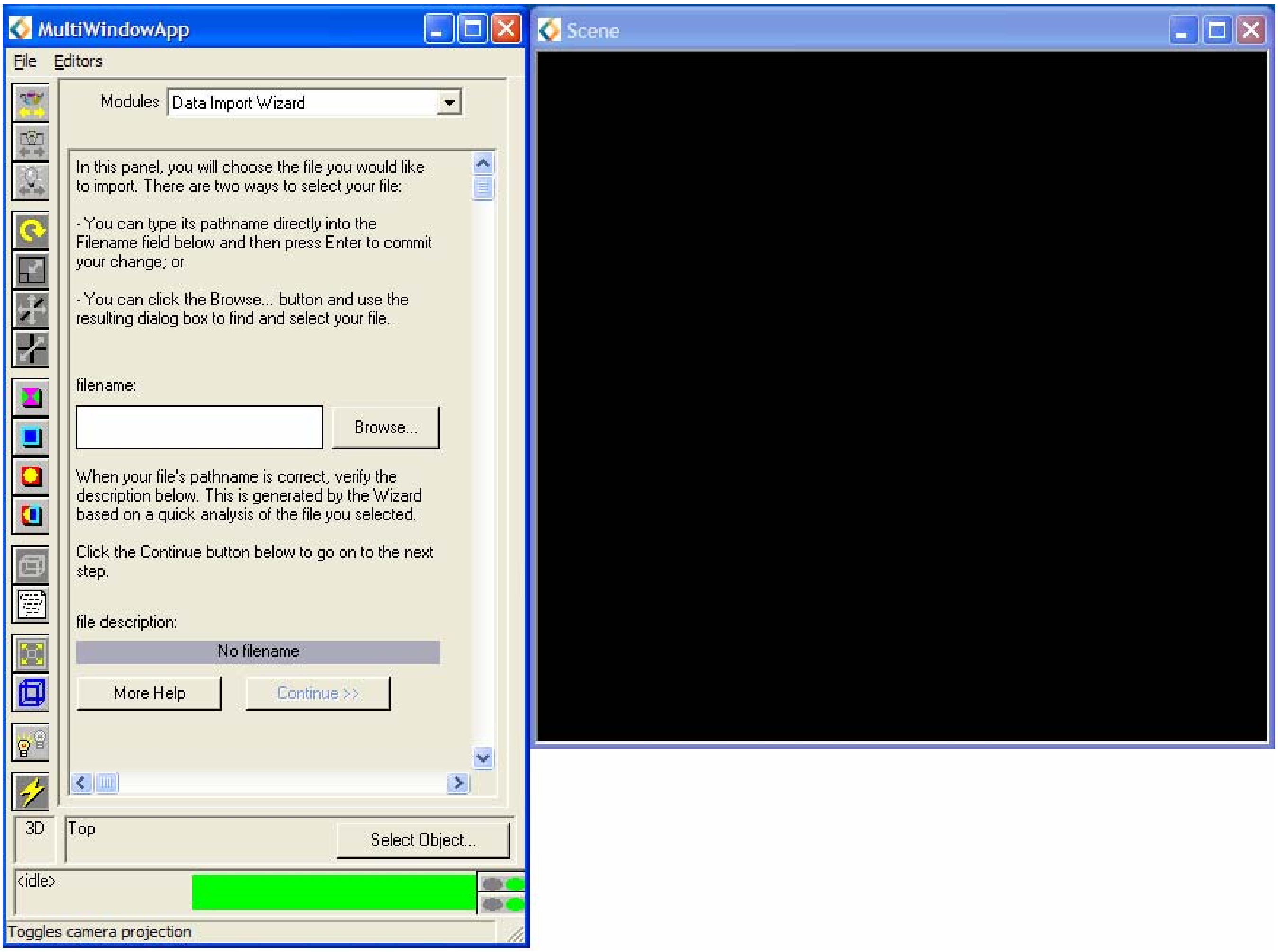

Figure C.3: **MultiWindowApp** DataViewer

By default the DataViewer will be split in two with the Editor Panel, left of Figure C.3 and the Viewer window, right of Figure C.3 as two separate windows.

This layout will not affect the visualization process or method. However, the DataViewer layout can be altered by reloading and redefining the application type. This is unnecessary but if preferred follow these steps:

- Remove the Viewer window and Editor Panel by selecting the **cross** in the right hand corner of each window.
- From the Toolbar at the top of the Network Editor select **File**, **New Application** to display the **New Application** pop up window shown in Figure C.4.
- Within the **New Application** pop up box select **Single-window DataViewer**, **3D viewer, Add Data Import Wizard to application** and **Add Data Visualization Wizard to application** (Figure C.4).
- Select **OK** to proceed.

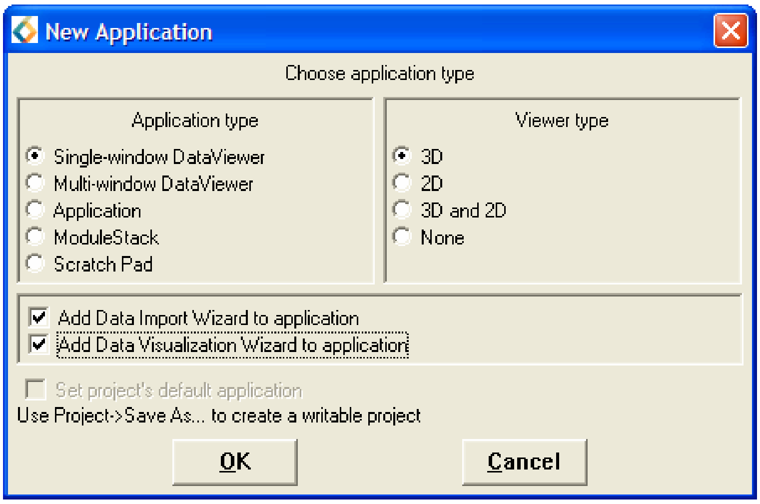

Figure C.4: **New Application** Pop up Window

- The resulting DataViewer can be seen in FigureC.5
- The Network Editor is shown in Figure C.6.

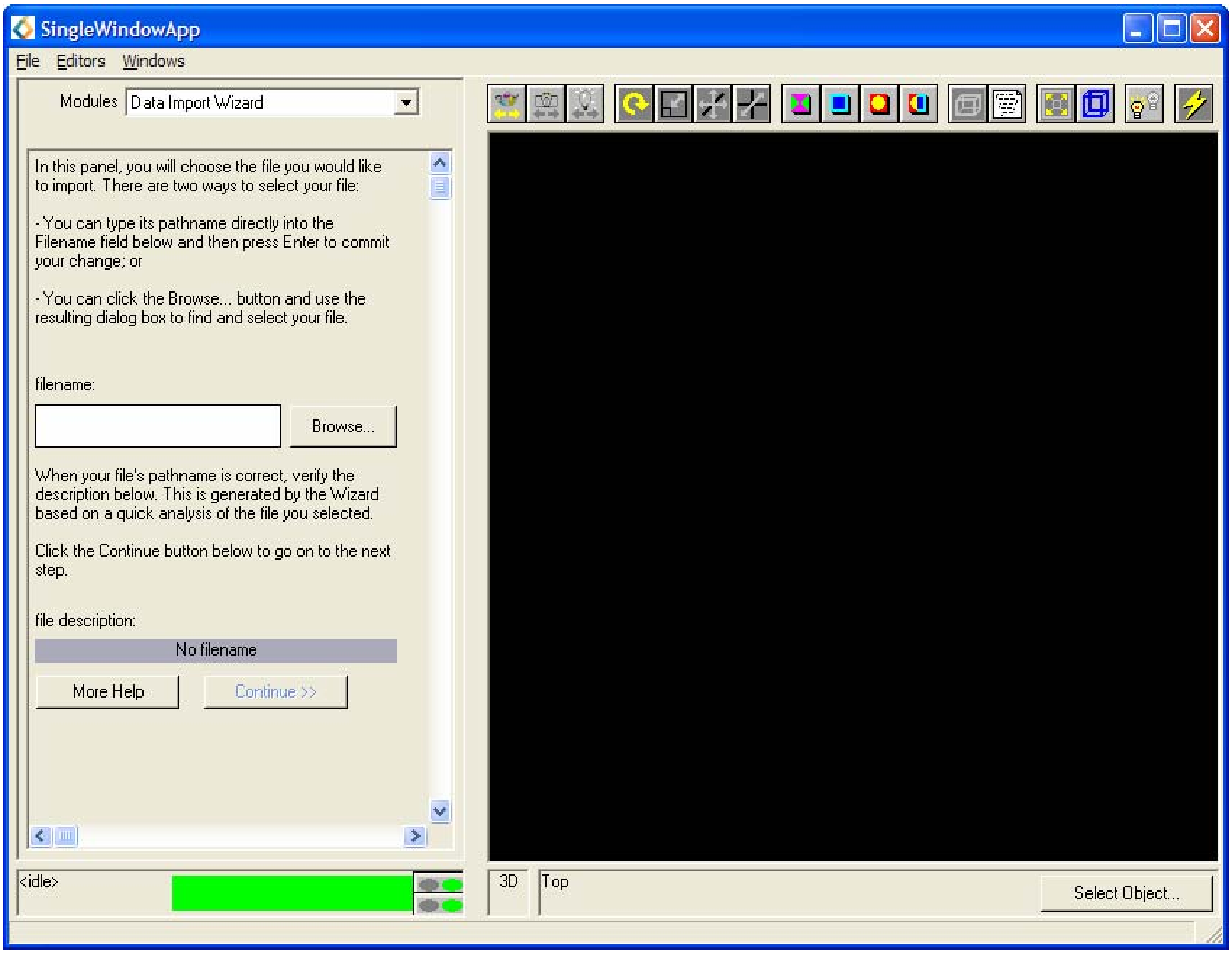

Figure C.5: **SingleWindowApp** DataViewer

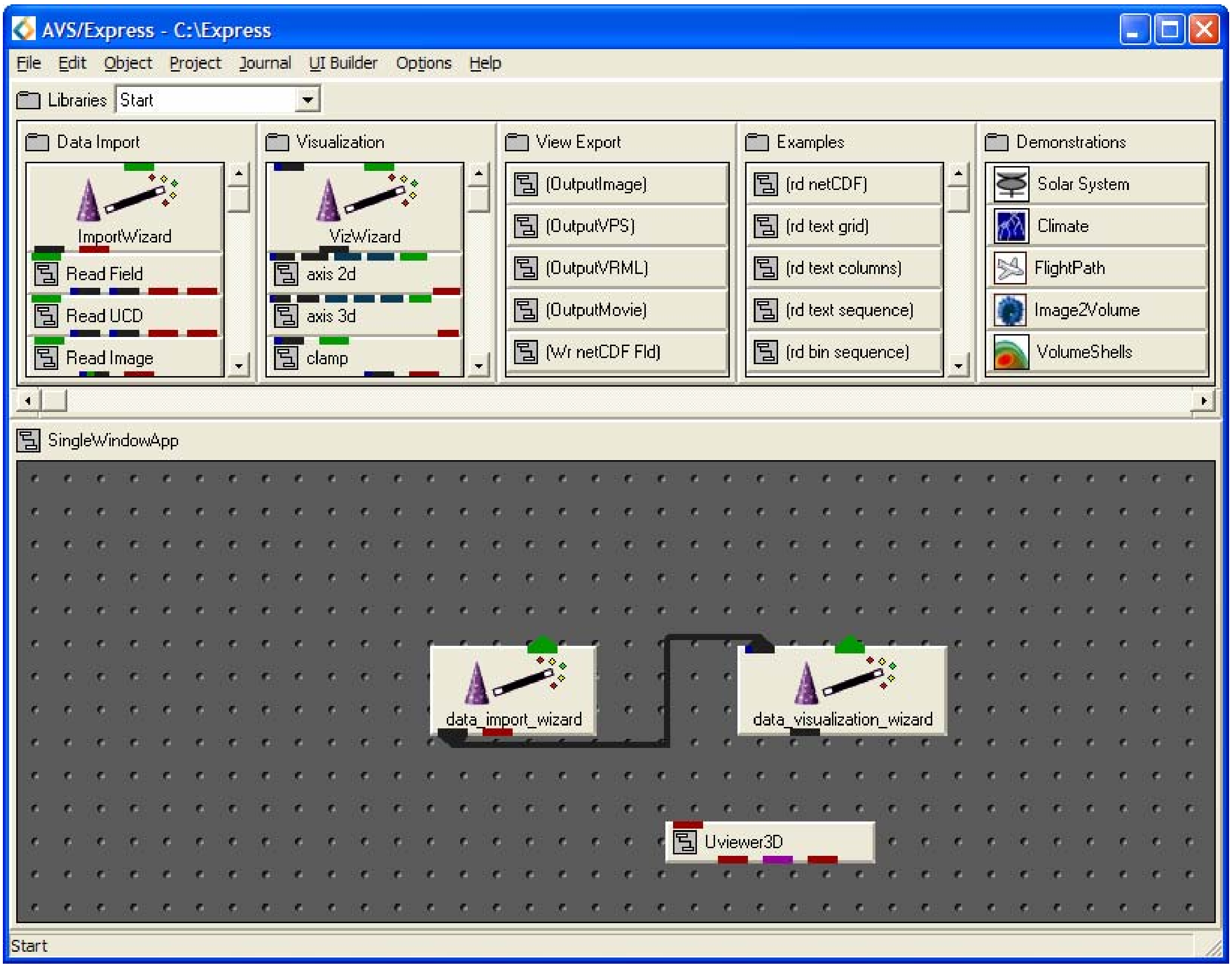

Figure C.6: Network Editor in the **Start** Library

A connection between the two Wizards will be setup automatically (Figure C.6).

C.2.3: Structure of the Network Editor

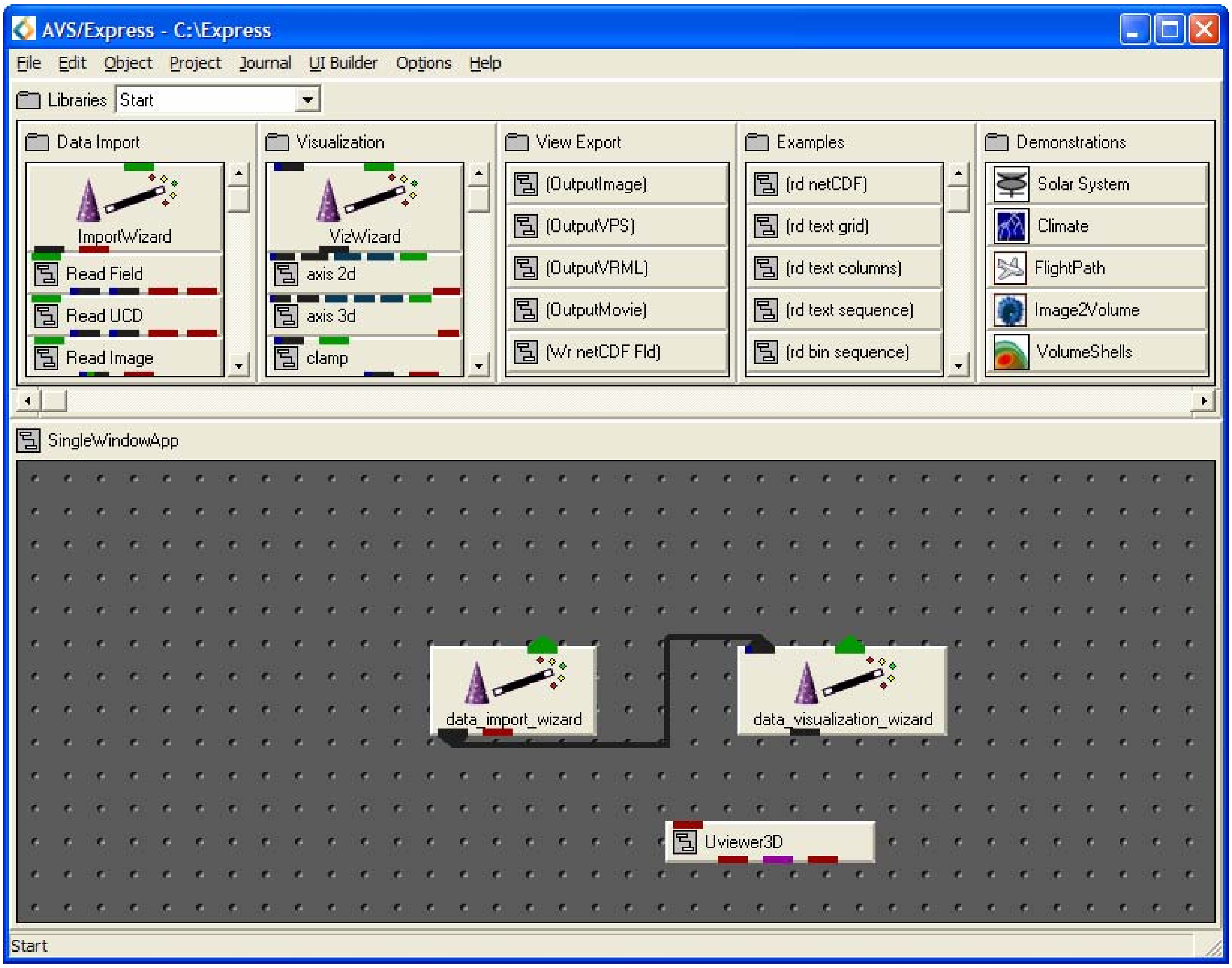

Figure C.7: Network Editor in **Start** Libarary with Wizards Instanced

- The Network Editor should have a layout similar to Figure C.7, showing all major components of the Network Editor.
- The **Start** library of modules is on view above the application workspace. This library enables new users to create and learn about AVS/Express applications quickly. Within the Start library modules are divided into sub-libraries:
  - Data Import
  - Visualization
  - View Export
  - Examples
  - Demonstrations
- **Data Import**, **Visualization**, and **View Export** provide a means of creating simple applications for users to get started on.
- **Examples** and **Demonstrations** provide a learning tool to help inexperienced users.

C.2.4: Reading a Data File

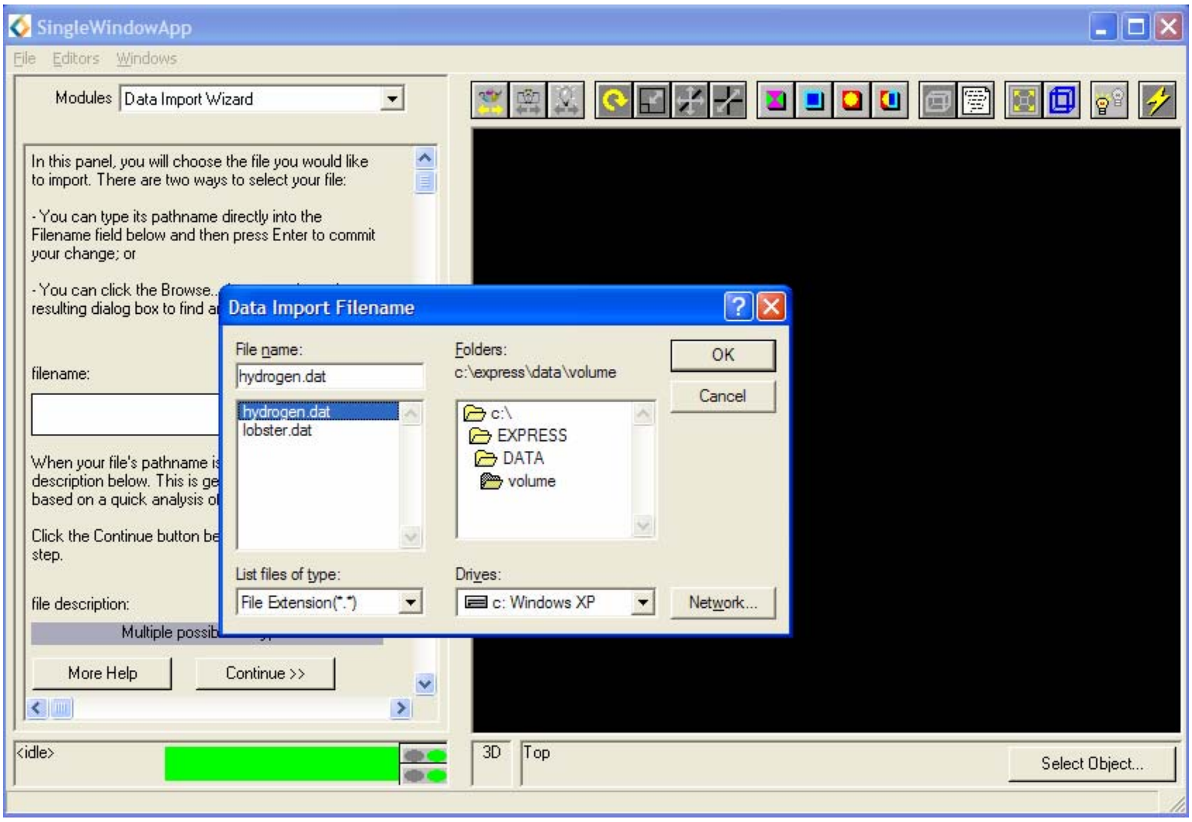

Figure C.8: **Data Import Filename** Browser

When any module including a Wizard is instanced to the workspace the controls for the module can be accessed in the Editor Panel of the DataViewer.

- Select the **browse**[3] button in the `Data Import Wizard` Editor Panel. The file browser should appear. (Figure C.8)
- Use the file browser to change the directory `c:\express\data\volume` and click on the file called `hydrogen.dat`.
- Select **Continue** and the **Multiple options file import Wizard** will appear as shown in Figure C.9.

[3] As default the `Data Import Wizard` Editor Panel should be loaded in the DataViewer. If it is not, perform the following two steps: Select **Modules** from the **Editors** drop down menu, select the `Data Import Wizard` from the **Modules** drop down list.

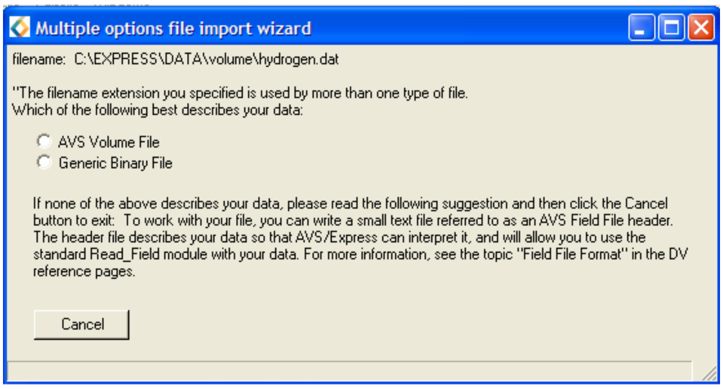

Figure C.9: **Multiple Options File Import Wizard**

As the file extension "`.dat`" is used by more than one type of file it is necessary to define the data so it can be handled appropriately. In addition to the two options provided by the Wizard there are also suggestions that can be followed if neither of the options are suitable.

- Select the **AVS Volume File** option and the Wizard will produce an image in the DataViewer window.
- The network created by the Wizard that is being executed to produce the visualization can be seen in the Network Editor as shown in Figure C.10.

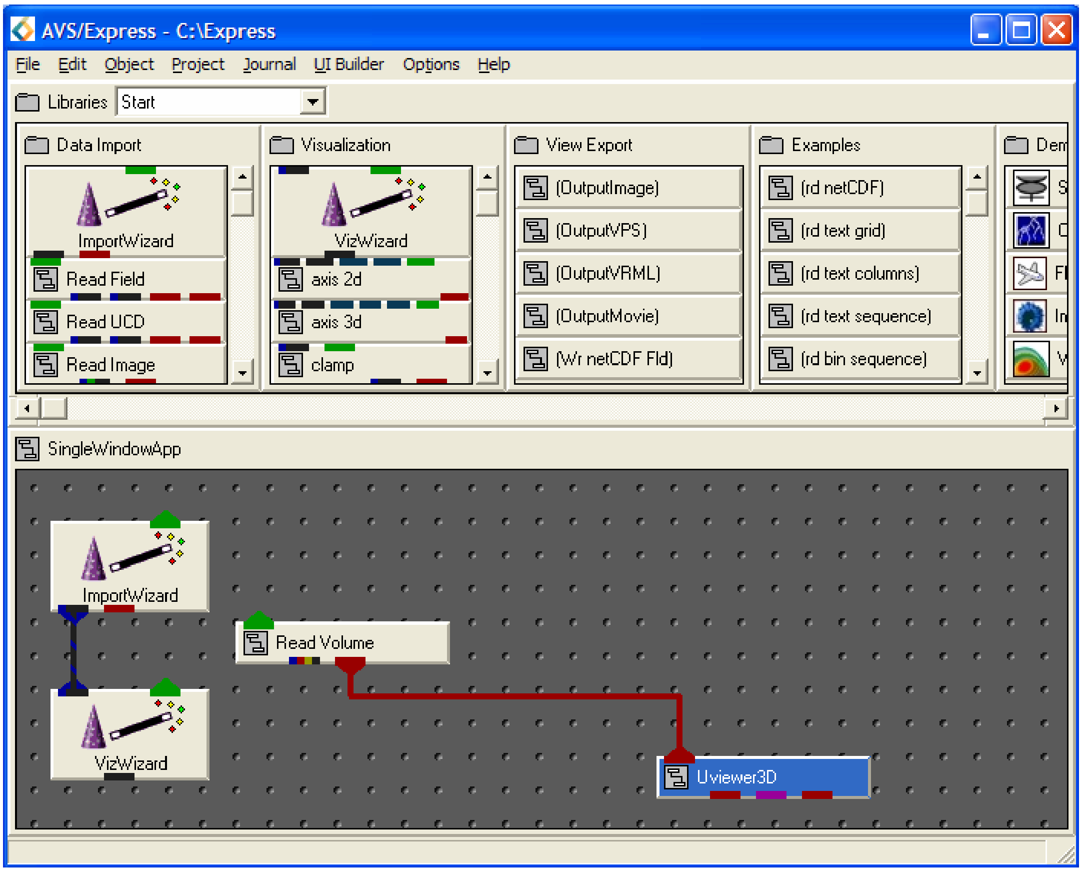

Figure C.10: Network Created by the `data_import_wizard`

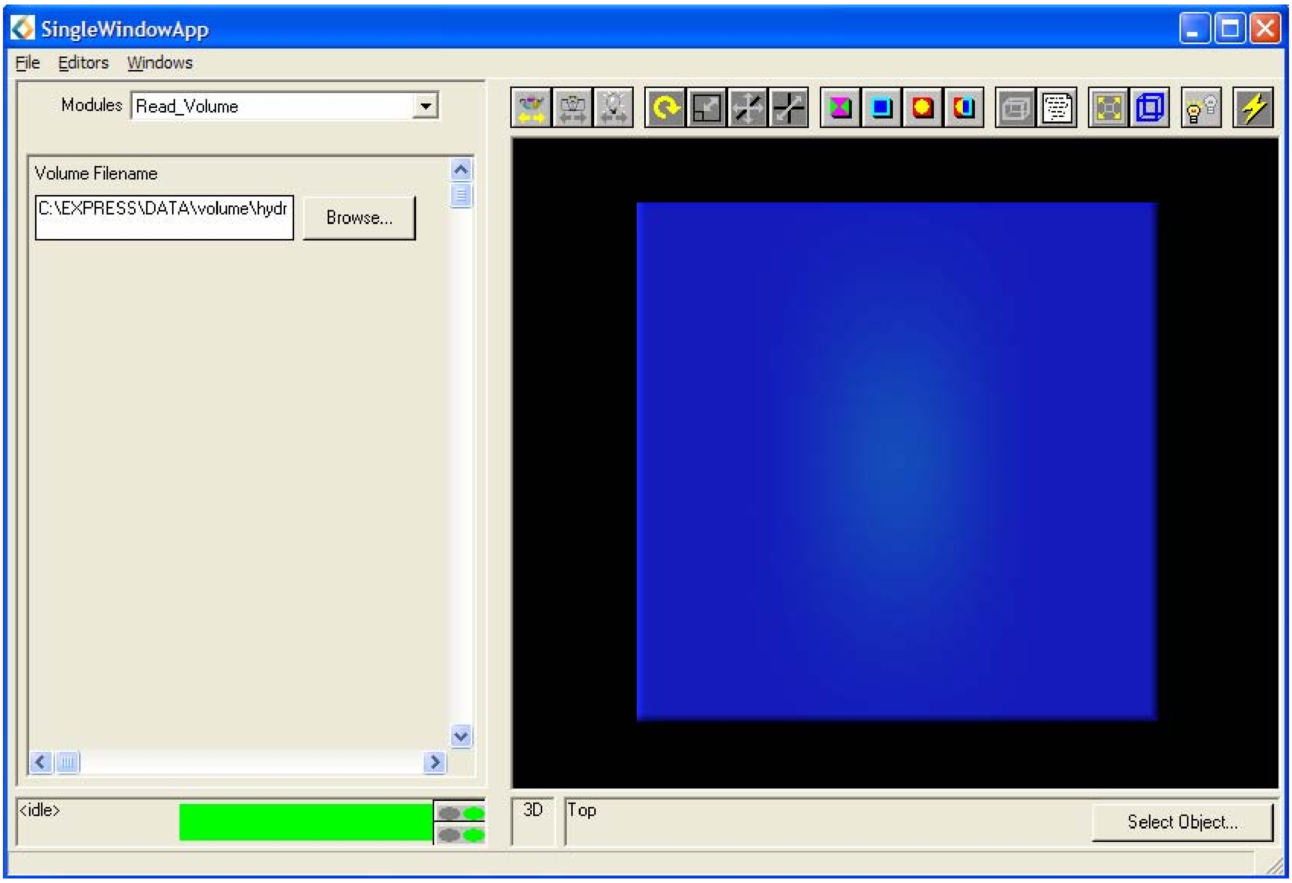

Figure C.11: Image Created by the `ImportWizard`

- The DataViewer window should display an image similar to that shown in Figure C.11.
- Now that an image has been obtained it is necessary to view it in a way that is useful and provides relevant information.

C.2.5: Visualizing the Data

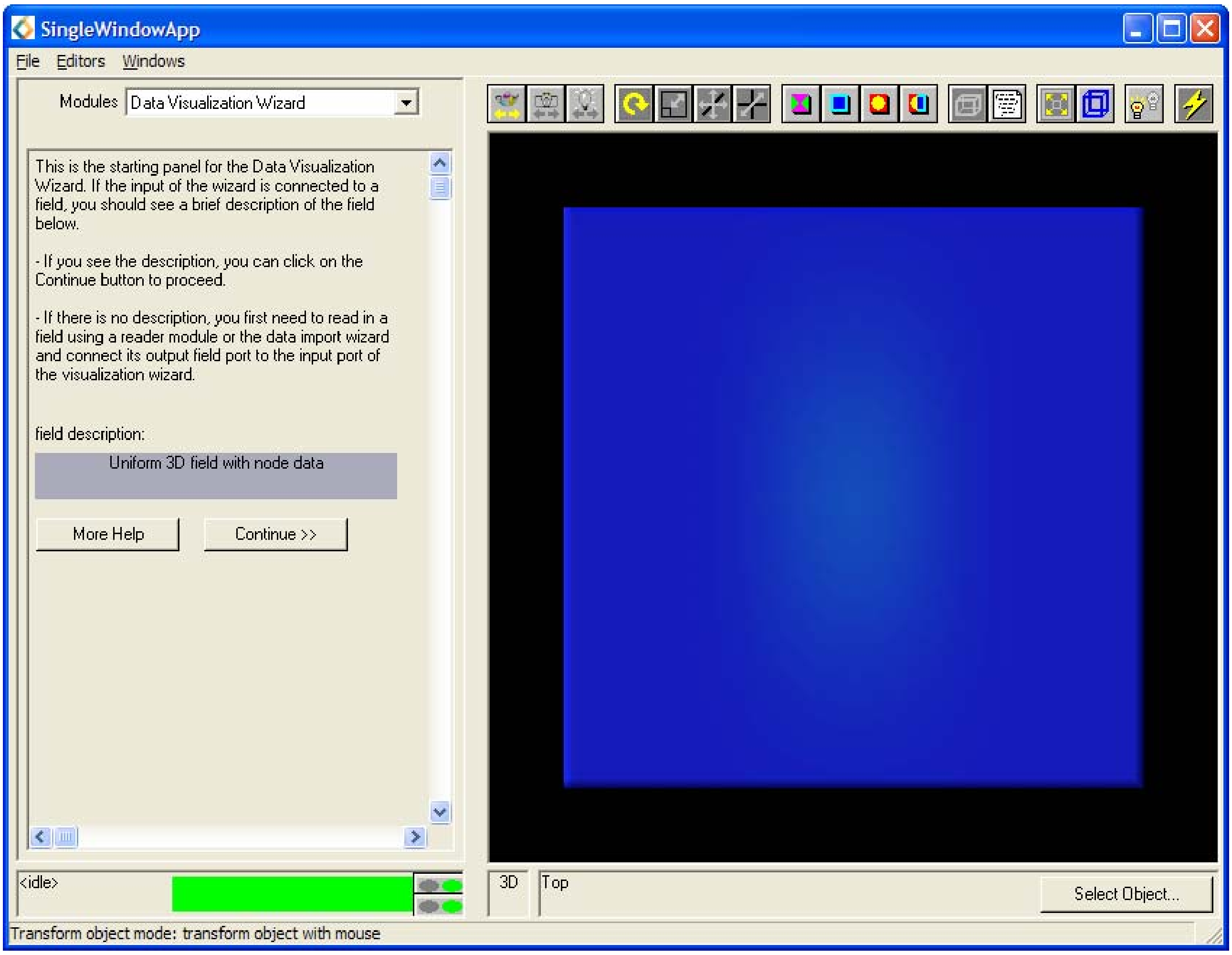

Figure C.12: `Data Visualization Wizard` Editor Panel

- Select `Data Visualization Wizard` from the **Modules** drop down list, its Editor Panel will be displayed as in Figure C.12.
- Select **Continue** >> in order to load the **Data Viz Wizard** where the visualization method can be defined.

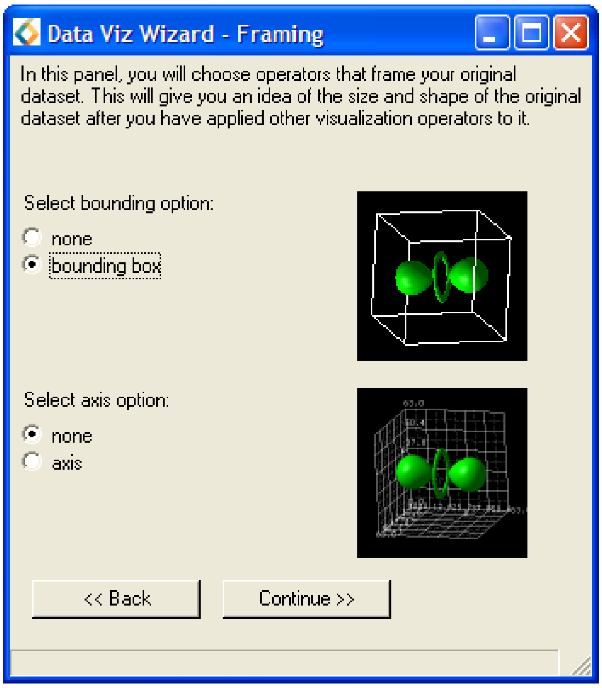

Figure C.13: **Data Viz Wizard** – Framing

- In the **Framing** window select the **bounding box** option, leave the other as default (**none**) and **Continue** >> (Figure C.13)

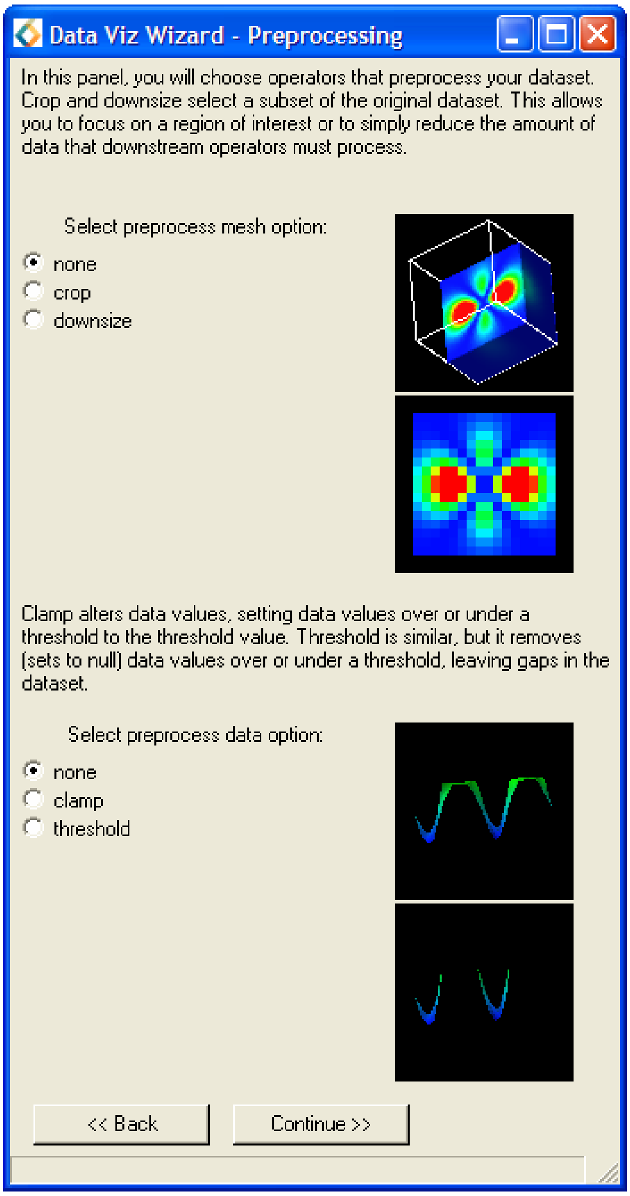

Figure C.14: **Data Viz Wizard** - Preprocessing

- In the **Preprocessing** window leave all options as default (**none**) and **Continue** >> (Figure C.14).

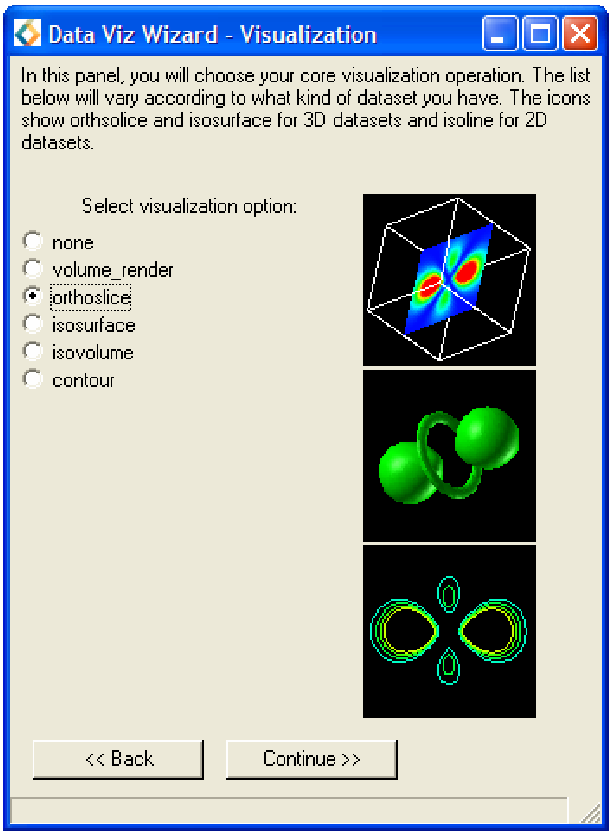

Figure C.15: **Data Viz Wizard** - Visualization

- In the **Visualization** window select **orthoslice** and **Continue** >> (Figure C.15).

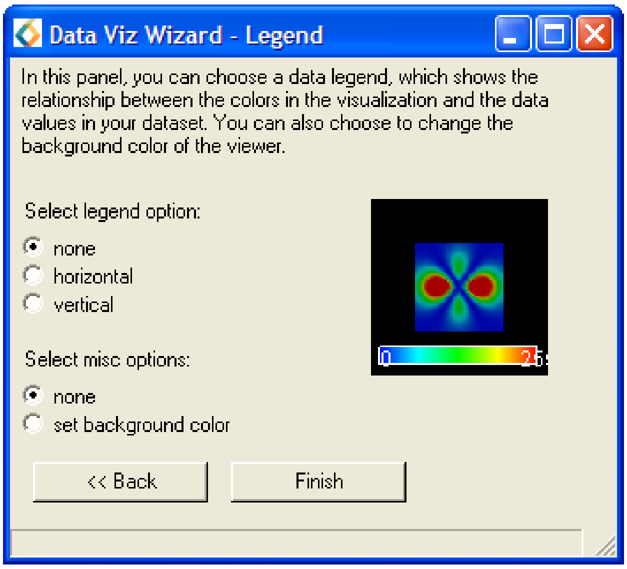

Figure C.16: **Data Viz Wizard** - Legend

- Finally in the **Legend** window leave all options as default (**none**) and **Finish** (Figure C.16).

The network created by the `VizWizard` that is being executed to produce the visualization can be seen in the Network Editor as shown in Figure C.17

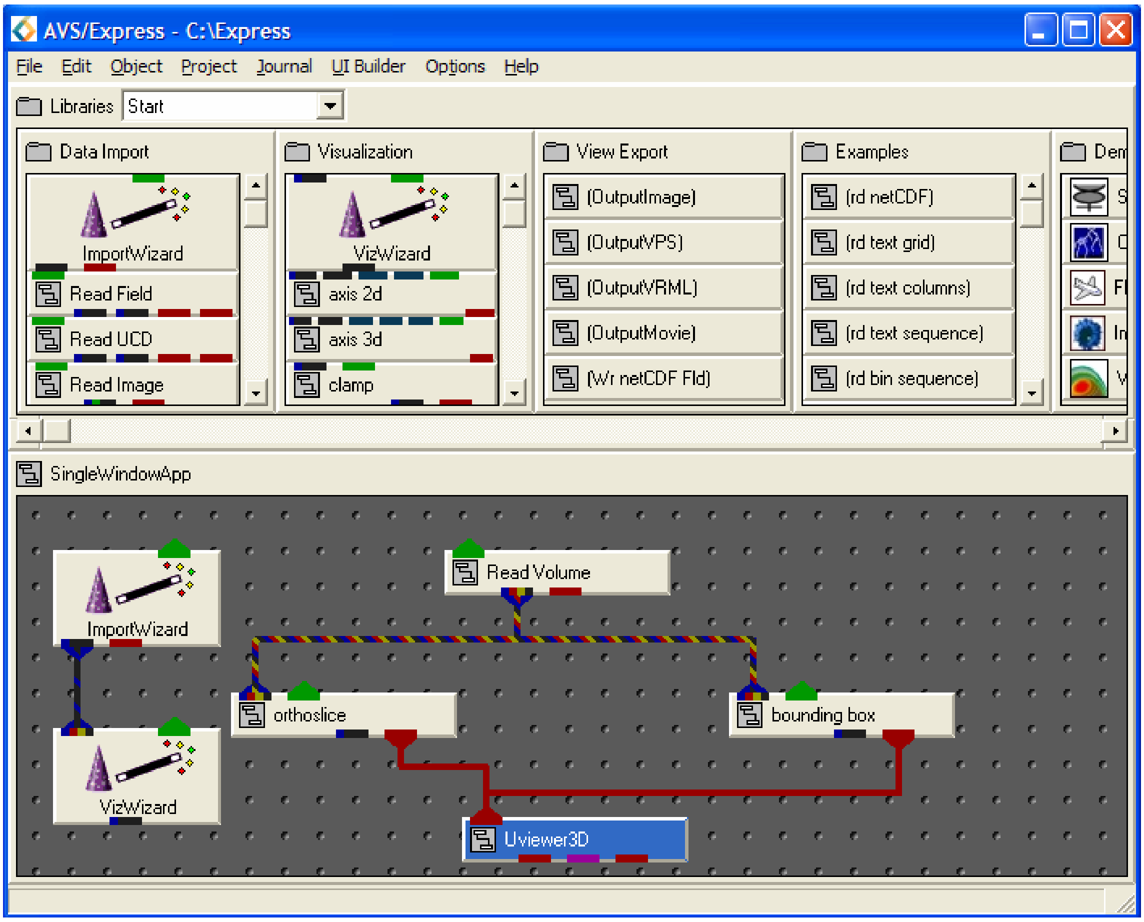

Figure C.17: Network Produced by `VizWizard`

The DataViewer window should now display an image similar to that shown in Figure C.18. The data is being visualized as was intended; as a slice through the hydrogen atom in order to view the differing electron densities.

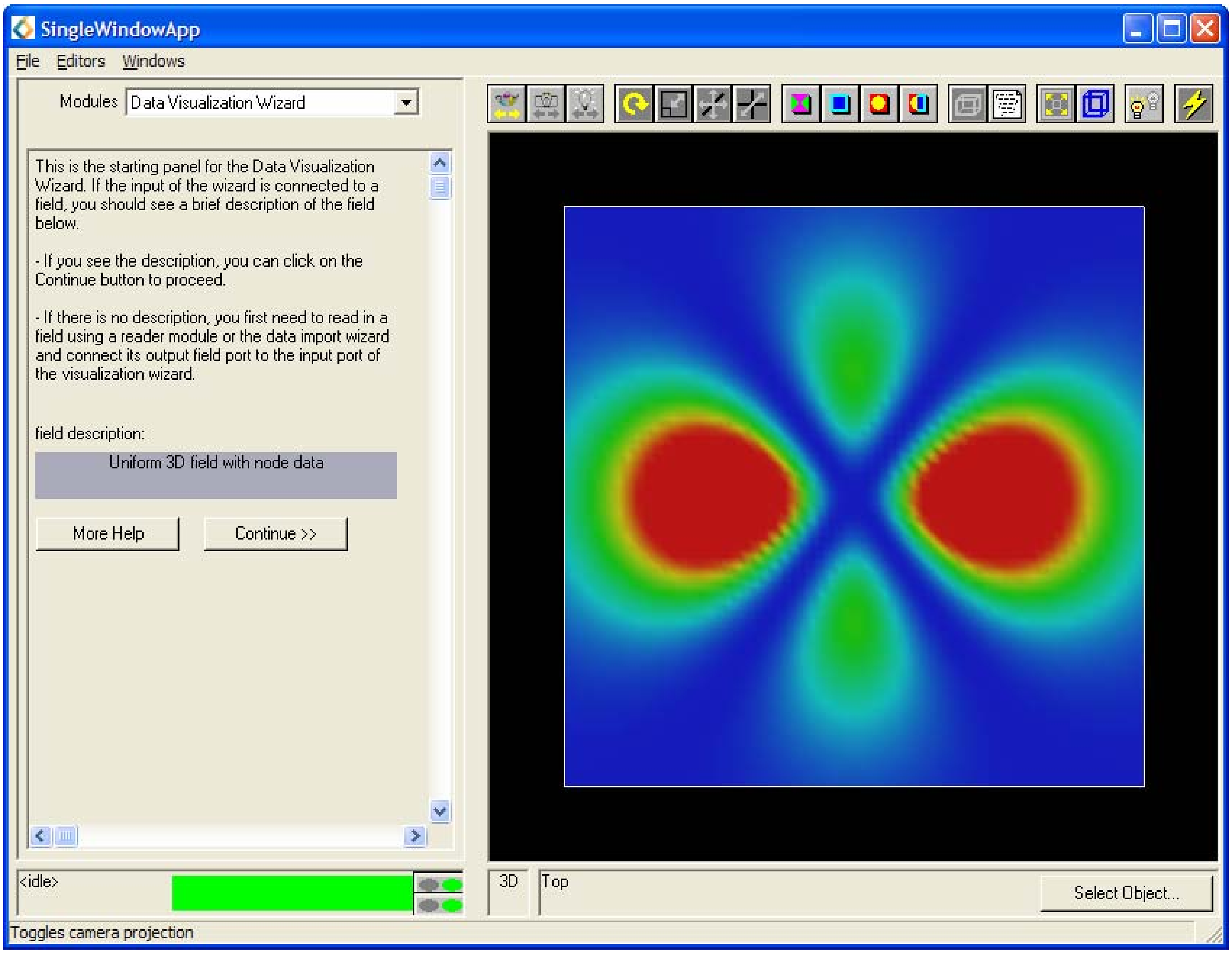

Figure C.18: `Data Visualzation Wizard` Editor Panel

Now that a suitable visualization of the image has been obtained, it is necessary to manipulate the image into the desired format.

C.2.6: Interacting with the Objects

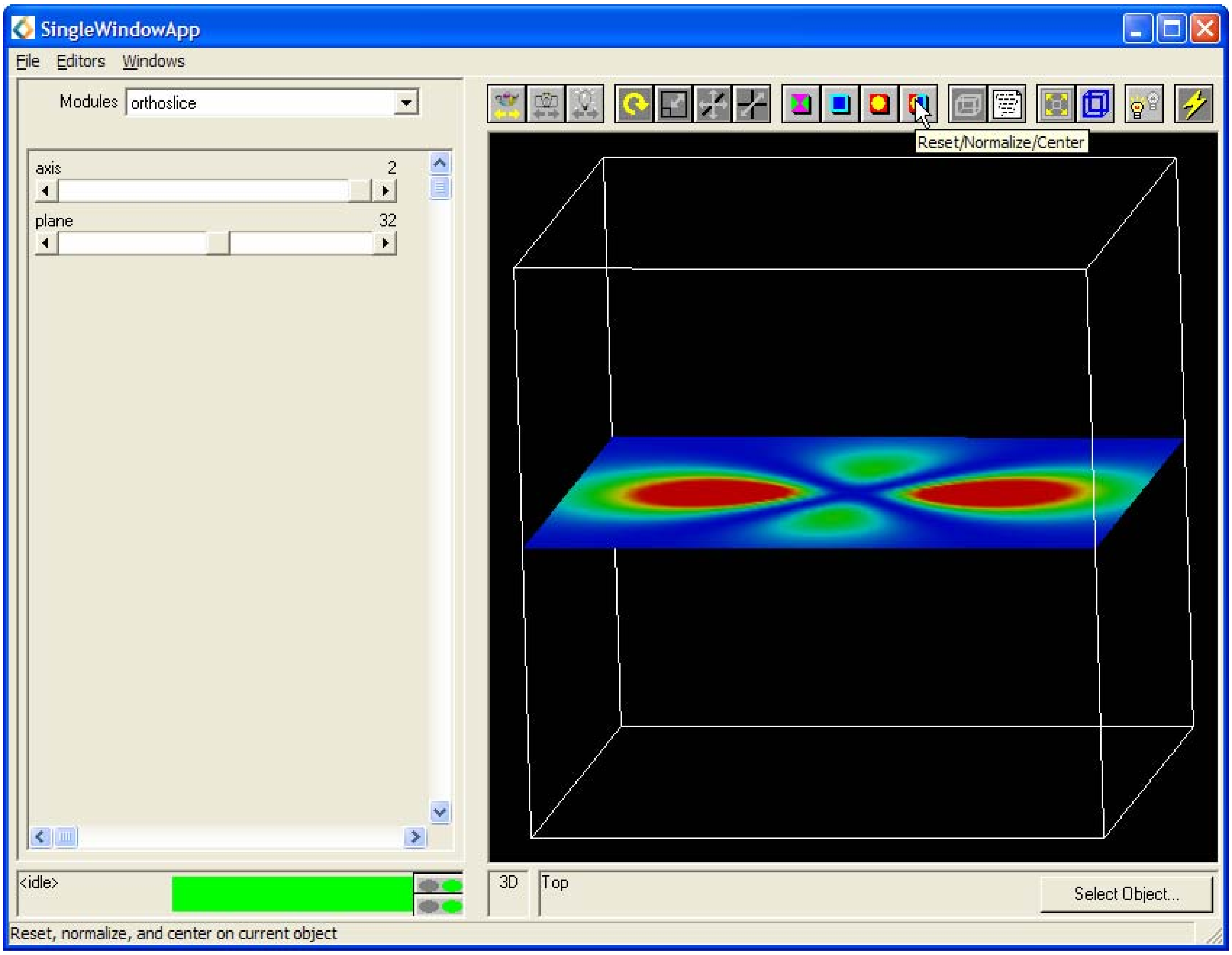

Figure C.19: `orthoslice` Editor Panel and Transformed Image

- The toolbar located above the DataViewer window, provides a quick way to manipulate images in the DataViewer.
- Move the mouse over the icons to reveal a small label indicating the activity of the icon.
- Click on the **Reset/Normalize/Center** icon, in order to suitably position the object in the window.
- To transform the image click on an icon in the Toolbar, place the cursor in the window and while holding down the **left** mouse button transform the object as desired.
- Now using the **scale**, **rotate** and **translate** icons manipulate the object until the scene appears similar to that shown in Figure C.19
- Select `orthoslice` form the **Modules** dropdown list to display its parameters in the Editor Panel.
- Altering the value of the **plane** slider will allow the differing densities throughout the hydrogen atom to be observed.
- Altering the value of the **axis** slider will alter the axis that the slice is perpendicular to.
- Alter the values of these two parameters and observe the effects.

C.3: Exercise 2

In this exercise the variation in bone density through the body of a lobster will be visualized.

- Using the `Data Import Wizard`, select the file browser and load the data file `lobster.dat`.
- Build the same network as Exercise 1 using the `Data Visualization Wizard`.
- Using similar manipulation techniques as those described in Exercise 1, investigate the bone density of the lobster within its exoskeleton.
- Alter the parameters of the `bounds` and `orthoslice` modules using the Editor Panel widgets.
- Rerun the `Data Visualization Wizard` from the DataViewer Editor Panel by selecting **Continue** >> and in the **Data Vis Wizard** - Visualization window select a different visualization method such as **isosurface** (Figure C.20).
- Alter the parameters of the new module using the Editor Panel widgets.

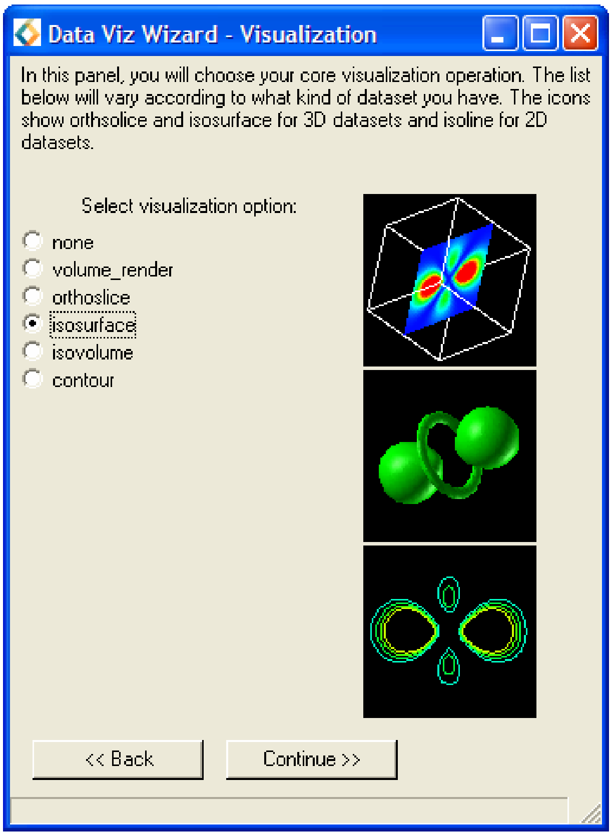

Figure C.20: **Data Viz Wizard** - Visualization

C.4: Exercise 3

In this example the data is from Whittakers field, CEDAR Farm, Reading, Berkshire and relates to soil properties within the field. The properties contained in the data file are `Potassium`, `Magnesium`, `Phosphorus`, `Ph` and `Organic Matter`. The data is contained in `SoilData.txt`. The data has been exported from Microsoft Excel into a text file, which contains the properties in addition to two columns of `X` and `Y` coordinates. A hardcopy of `SoilData.txt` can be found in Appendix D and should be referred as necessary.

This Geostatistics data is the subject of the visualization. The aim is to use the visualization as an aid to soil management for Precision Agriculture. AVS/Express does not have a read module designed to import data with a file type like `SoilData.txt`. The data will be imported using the `data_import_wizard` and visualized using the `data_visualization_wizard`.

C.4.1: Instancing the Wizards

- From the Toolbar at the top of the Network Editor select **File** followed by **New Application**.
- The **New Application** pop up window will be shown. Select **Single-window DataViewer**, **3D** and finally **OK**.
- Ensure the **Start** library is selected and locate the **ImportWizard** in the **Data Import** sub library.
- Select the **ImportWizard** using the **left** mouse button and drag onto the application workspace. Release the **left** mouse button to instance the Wizard to the workspace.
- Repeat this process for the **VizWizard** located in the **Visualization** library. The network should appear similar to that shown in Figure C.21.

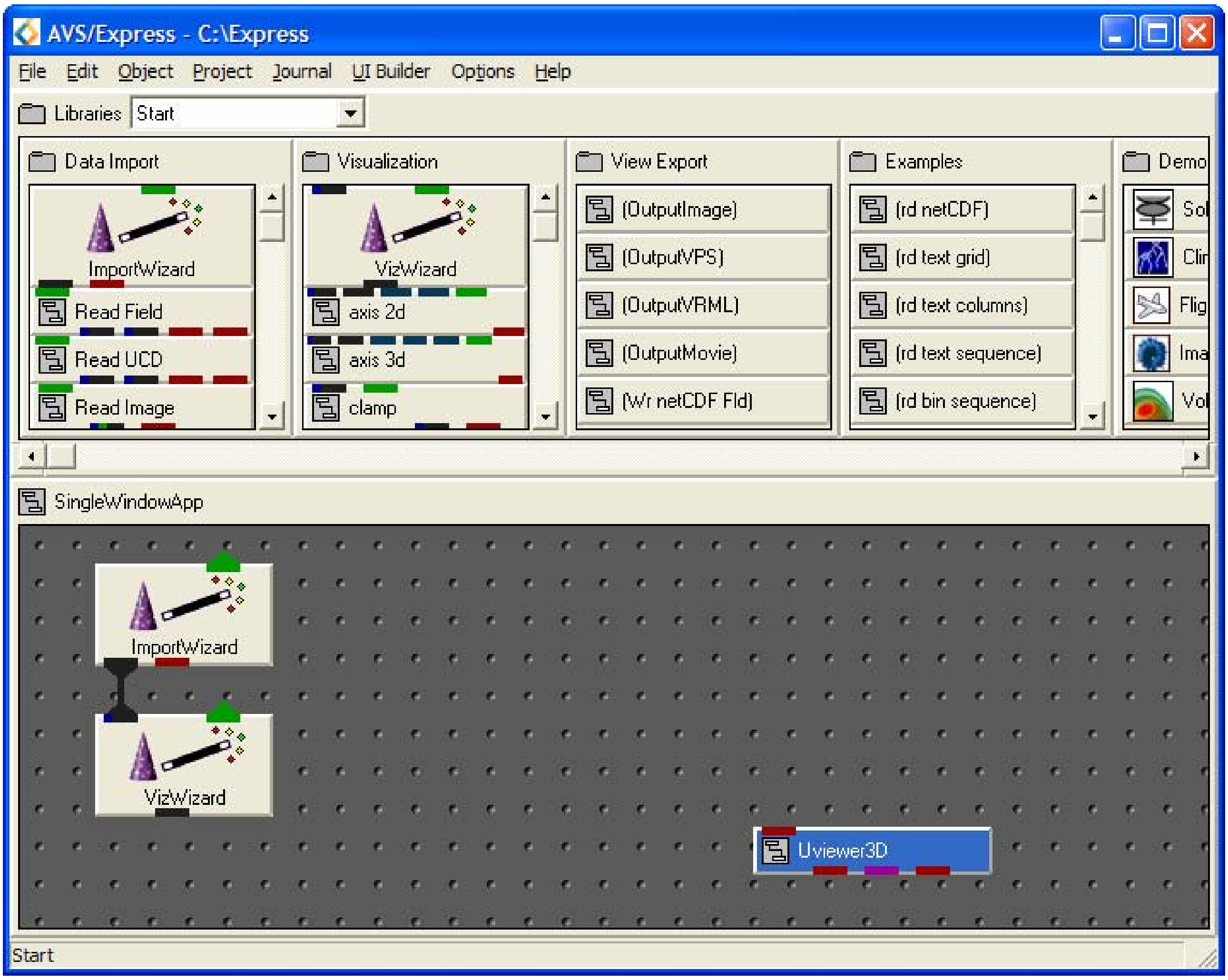

Figure C.21: Network Editor in **Start** Library with Wizards Instanced

C.4.2: reading the data file

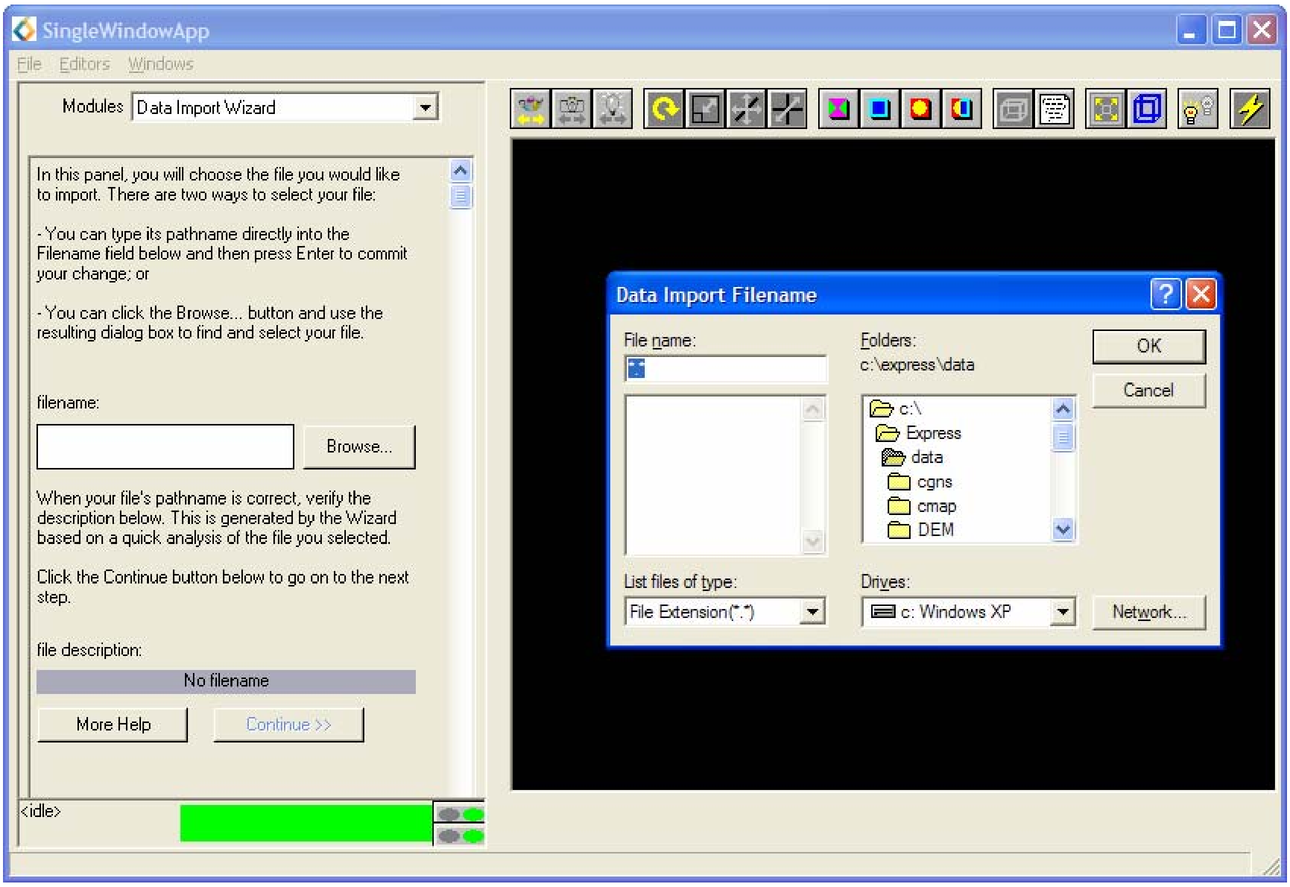

Figure C.22: **Data Import Filename** browser

- The `Data Import Wizard` **Modules** Editor panel should be open as shown in Figure C.22. Select the **Browse** button.
- Use the file browser to locate and load `SoilData.txt`.
- Select **Continue** >> and the **Unknown file import Wizard**[4] window will appear as shown in Figure C.23.

[4] The file extension "`.txt`" is unknown to AVS/Express. The data must be defined manually using the **Unknown file import wizard**. The wizard (Figure C.23) contains options and short descriptions. The data must be known and understood in order to select the correct option.

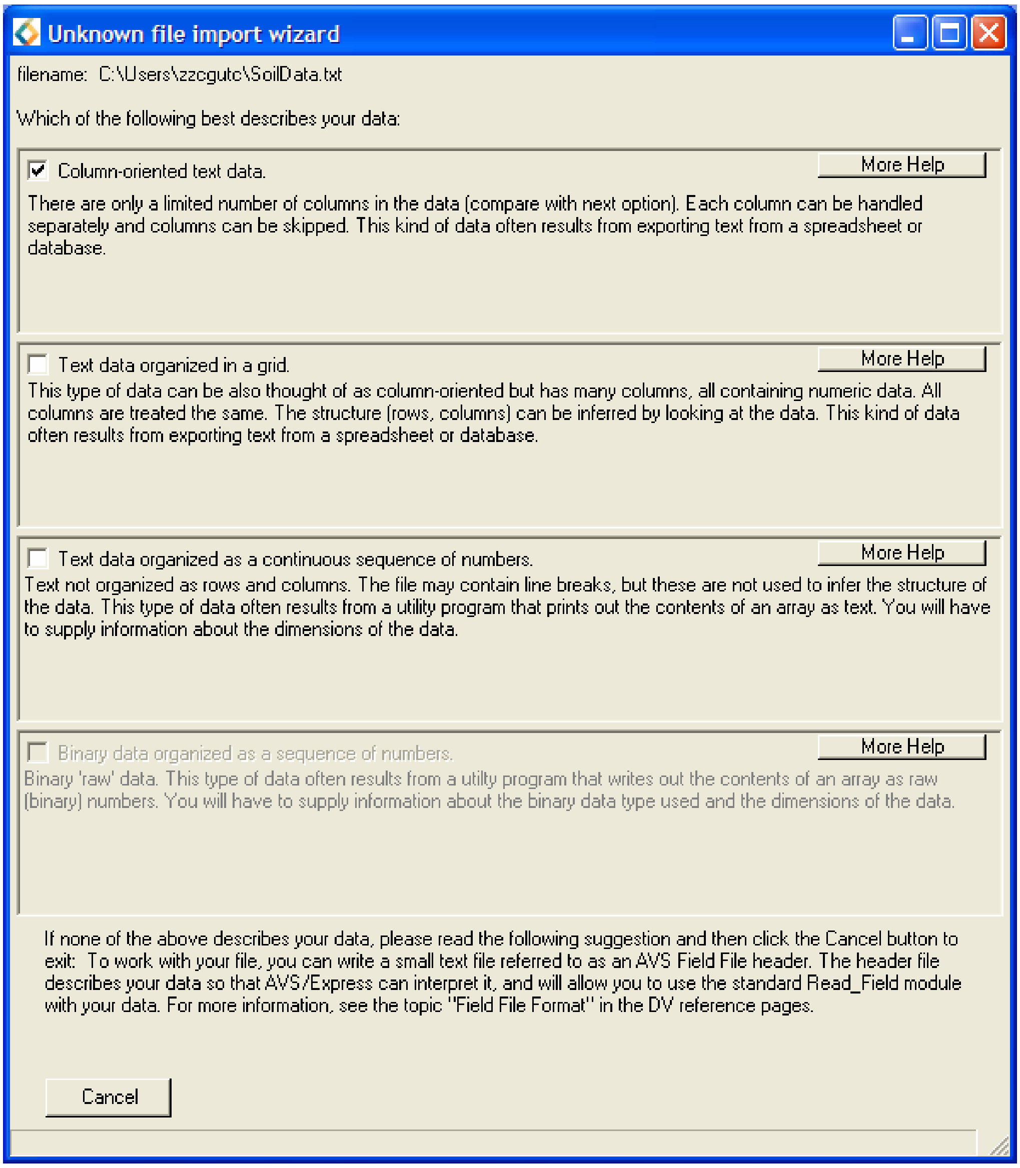

Figure C.23: **Unknown File Import Wizard**

- Select the **Column-orientated text data**[5] option and then select **Cancel**.
- Ignore the **Error Dialog** message shown in Figure C.24 – the error is due to the wizard automatically inputting incorrect values into the `Read Text Columns` module[6] (Figure C.25)

[5] Column-orientated text data allows each column to be individually defined. Columns can be skipped and given a different data type (float, int), there is more flexibility. This option is selected as the file contains coordinate and component data which need to be treated individually.

[6] The number of columns in the data file, the type of column separator in the data file and the presence of a header line.

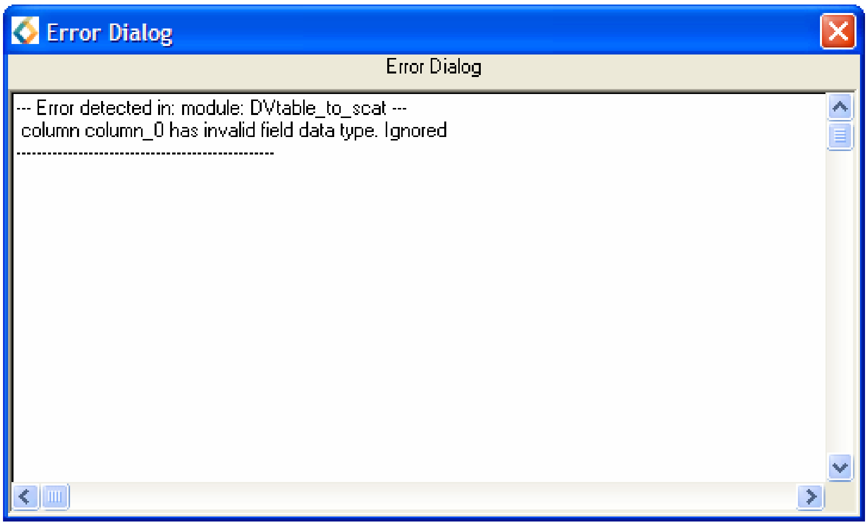

Figure C.24: **Error Dialog** Message

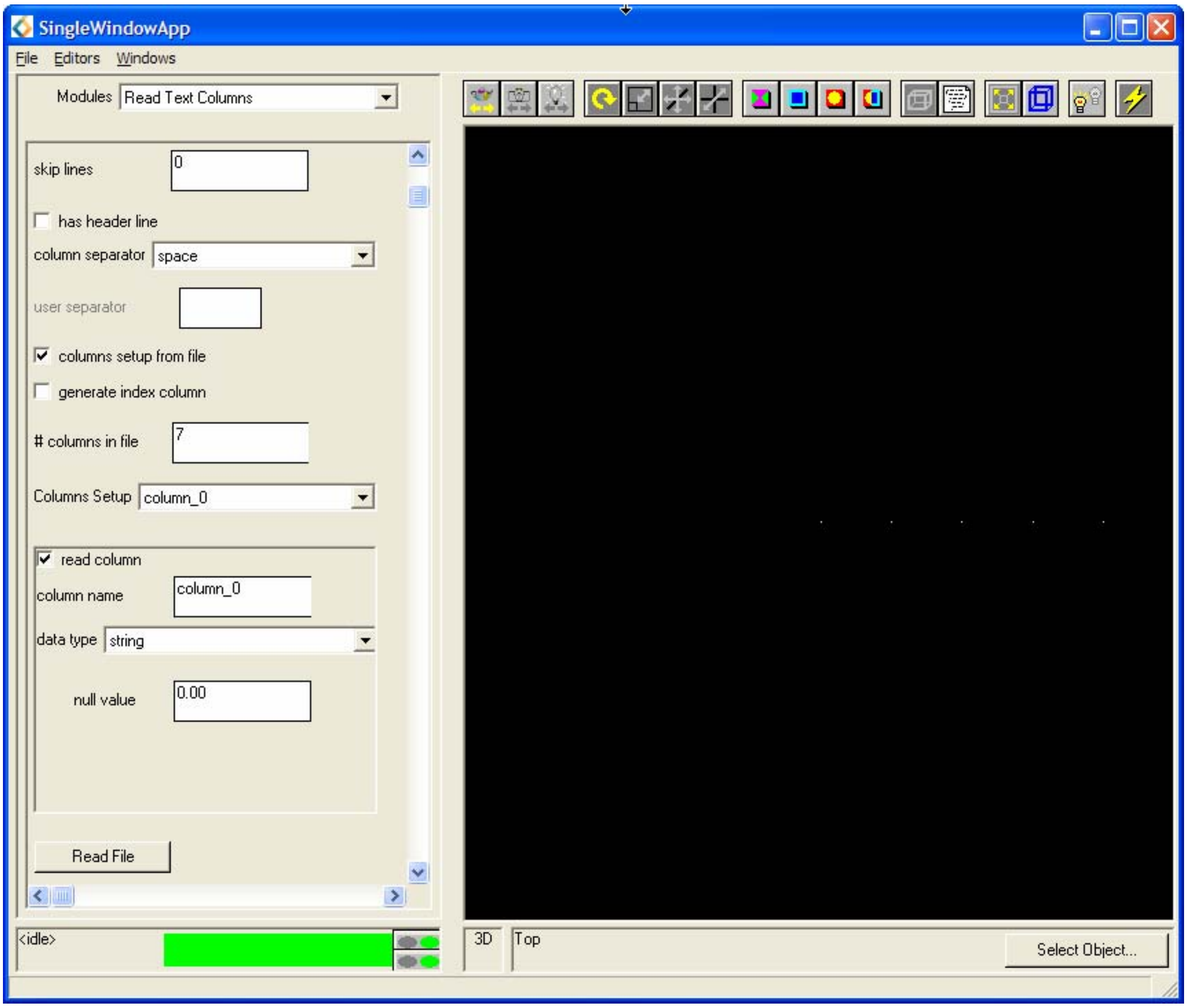

Figure C.25: `Read Text Columns` Editor Panel

C.4.3: Setting `Read Text Columns` Module Parameters

- The `Read Text Columns` **Modules** Editor Panel should be open as shown in Figure C.26. From here the parameters of the module can be set.

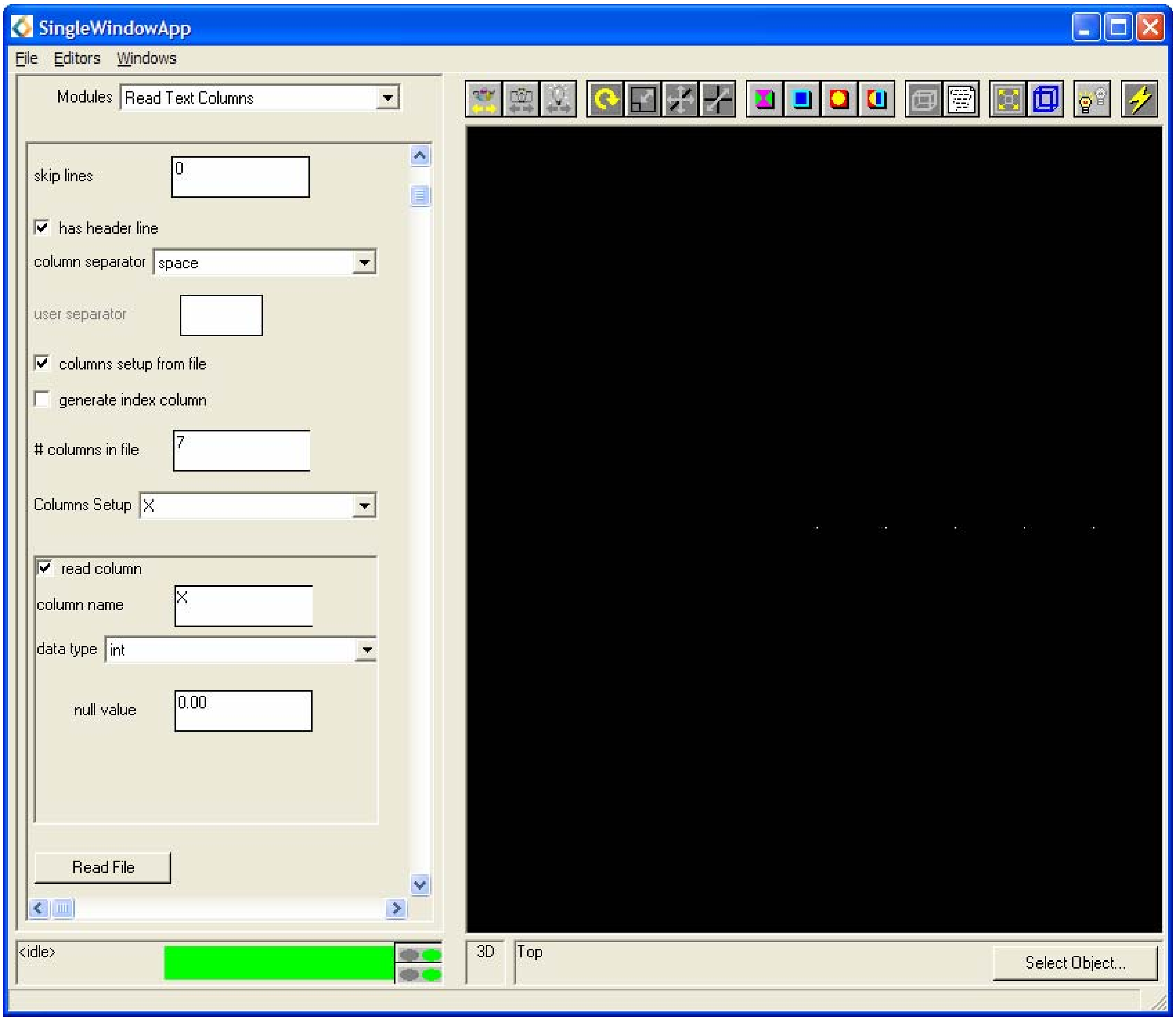

Figure C.26: `Read Text Columns` Editor Panel for **X** Coordinate

The following bullet points should provide enough information to set the correct parameters. It is still necessary to understand the structure of `SoilData.txt` and relate it to the parameters set (Appendix D).

- There is a **header line** describing the data in each of the **7 columns** in the file.
- There is a **space** which separates each of the columns.
- The **data type** of the x and y coordinates is **int**. Using the **Columns Setup** dropdown list select the required coordinate and set its properties using the widgets in the indented panel above the **Read File** button (Figure C.26).
- The **data type** of the 5 soil property components is **float**. Using the **Columns Setup** dropdown list select the required component and set its properties using the widgets in the indented panel above the **Read File** button (Figure C.27). It is important to set each component to **float**. This is tedious but necessary.

- Select **Read File** in order to read the data.

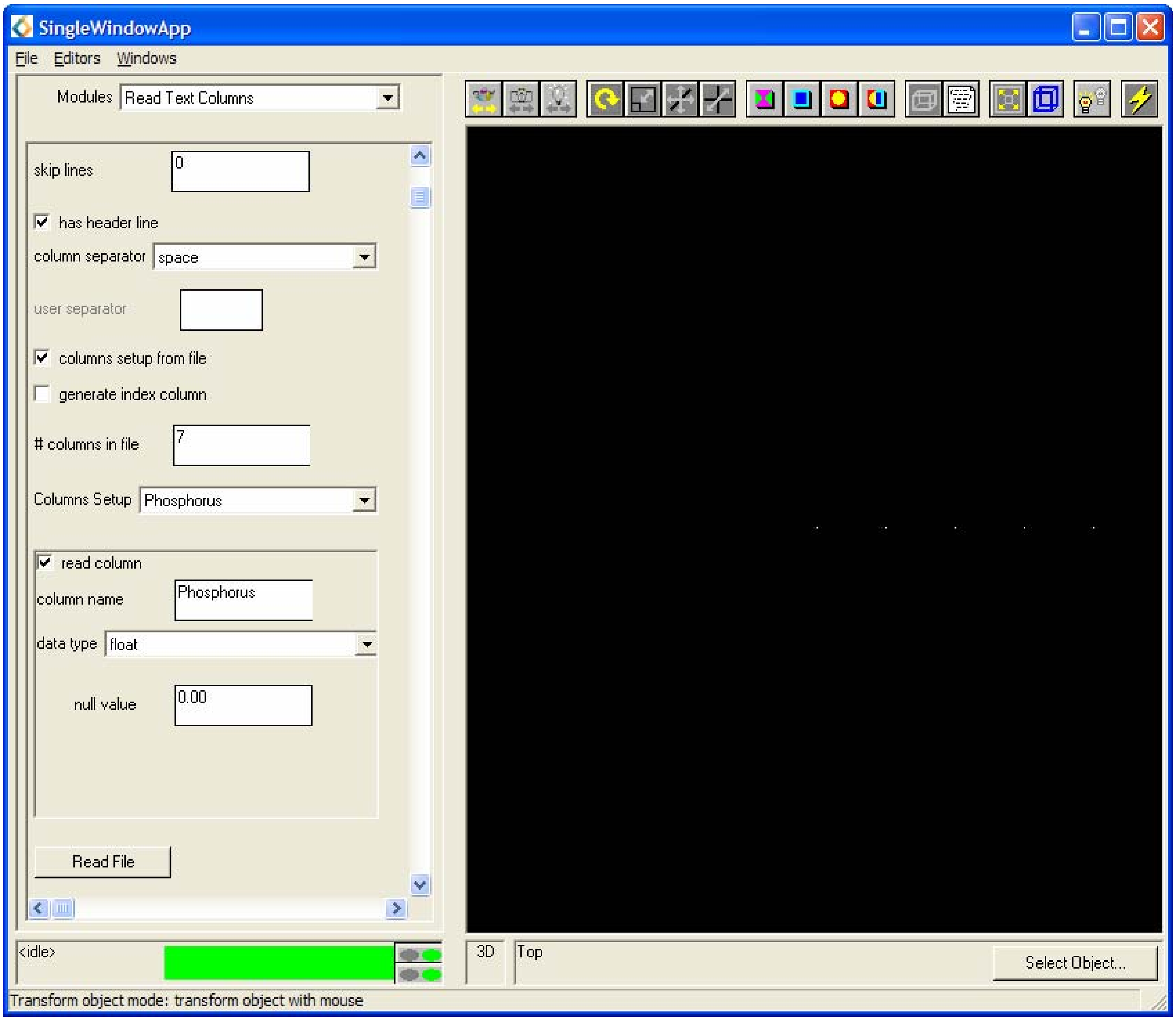

Figure C.27: `Read Text Columns` Editor Panel for **Phosphorus** Component

The data is being read into AVS/Express, however, it will not be visible as expected because the parameters of the `table to scatter field` module have not been set.

C.4.4: Improving the Macro Created by the ImportWizard

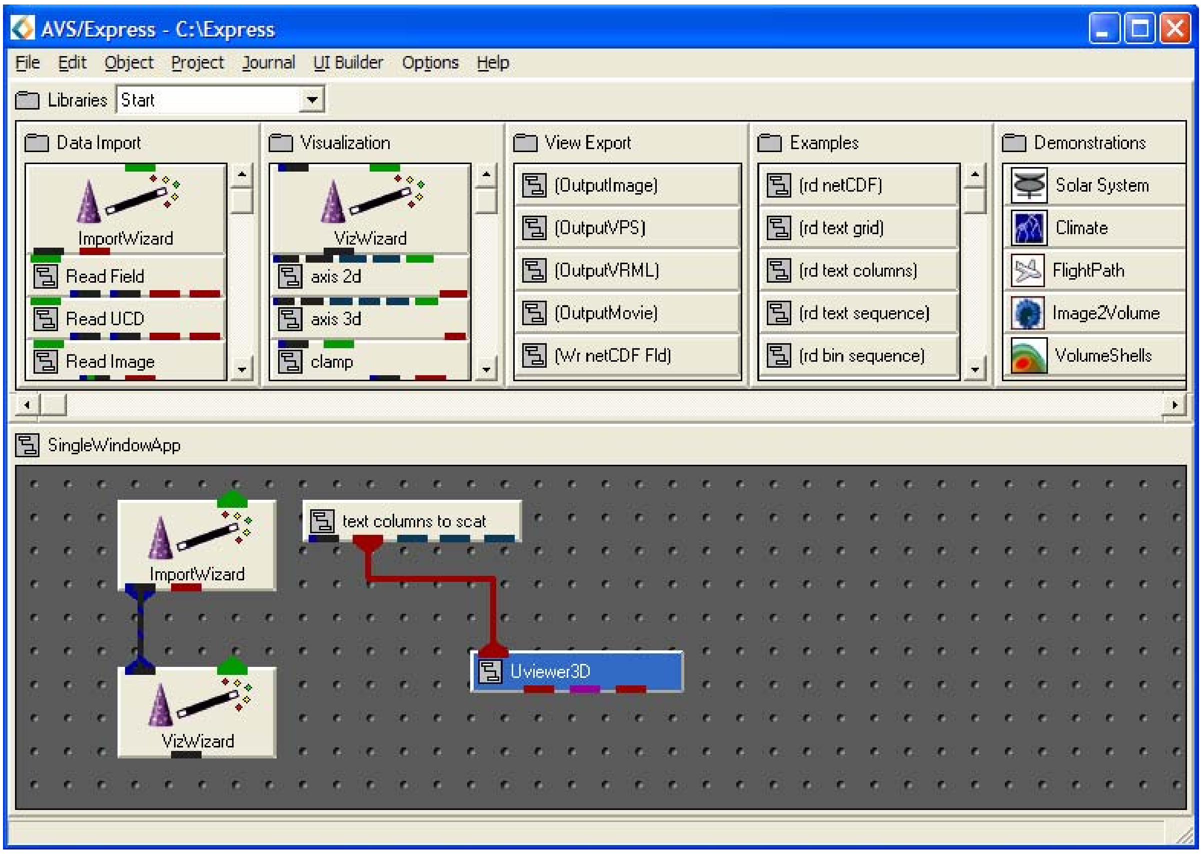

Figure C.28: Network Created by `Data_import_wizard`

- A macro `text columns to scat` has been loaded into the Network Editor by the `ImportWizard` (Figure C.28).
- Double click the macro to show its network in the Network Editor as seen in Figure C.29

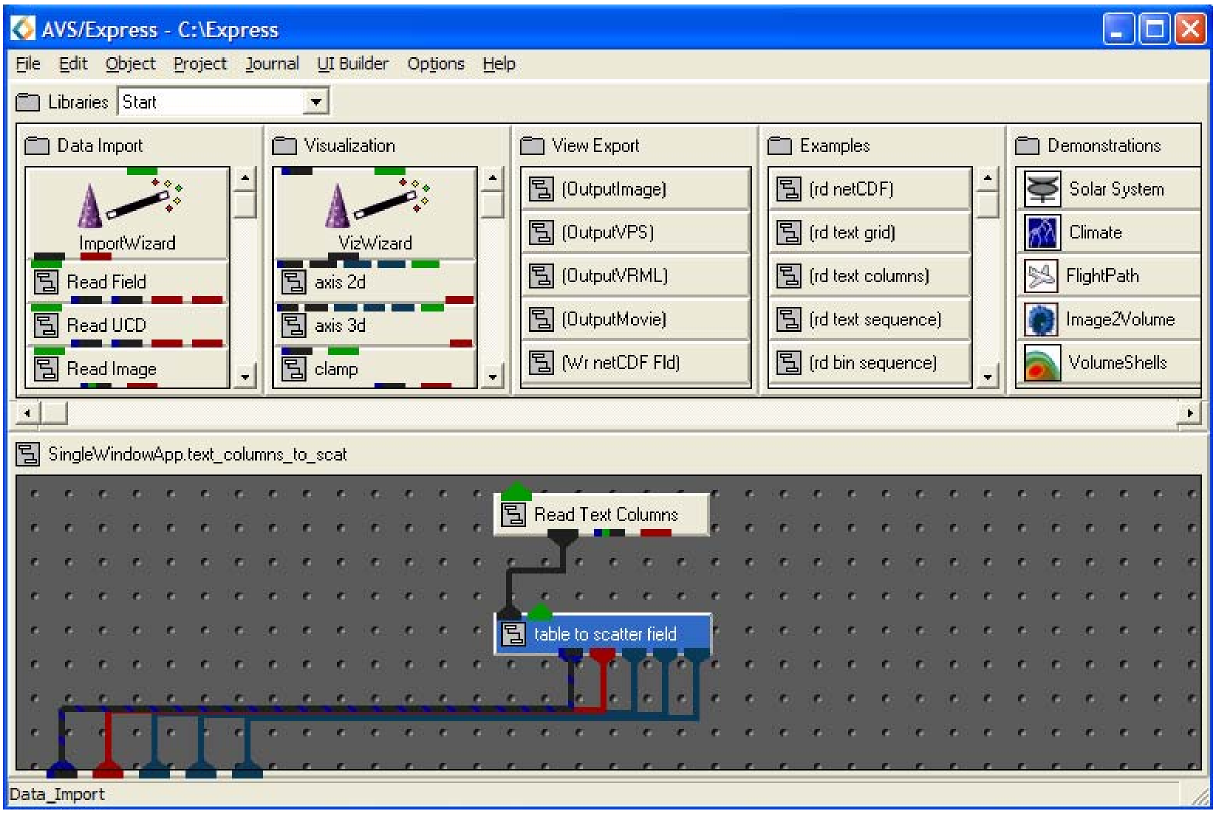

Figure C.29: `text_columns_to_scat` Macro

- The Macro, `text_columns_to_scat` consists of two modules:
  - o `Read Text Columns`: Reads an ASCII file containing any number of separated columns, often the result of exporting a spreadsheet
  - o `table to scatter field` (Mappers): maps the data in an array of columns to a scattered field.

The module `table to scatter field` is not the best module for the type of visualization required. The following module will be more useful:

  - o `table to uniform field` (Mappers): maps the data in an array of columns to a uniform field. Use uniform fields to represent regular data that forms continuous 1D, 2D, or 3D uniformly spaced arrays. Uniform field can be visualized in many ways including city plot, ribbon plot, surface plot, contour, isosurface, and isoline.

- Right click on the `table to scatter field` module and select **Delete** from the pop up menu. This will delete the `table to scatter field` module from the network of the `text columns to scat` macro.
- In the **Libraries** drop down list select **Main**.
- Locate the `table to uniform field` module in the **Mappers** sub library.
- Drag and drop the `table to uniform` module in to the `text columns to scat` macro network and connect to `Read Text Columns` as shown in Figure C.30.
- Connect all the output ports of the `table to uniform` module as shown in Figure C.30.

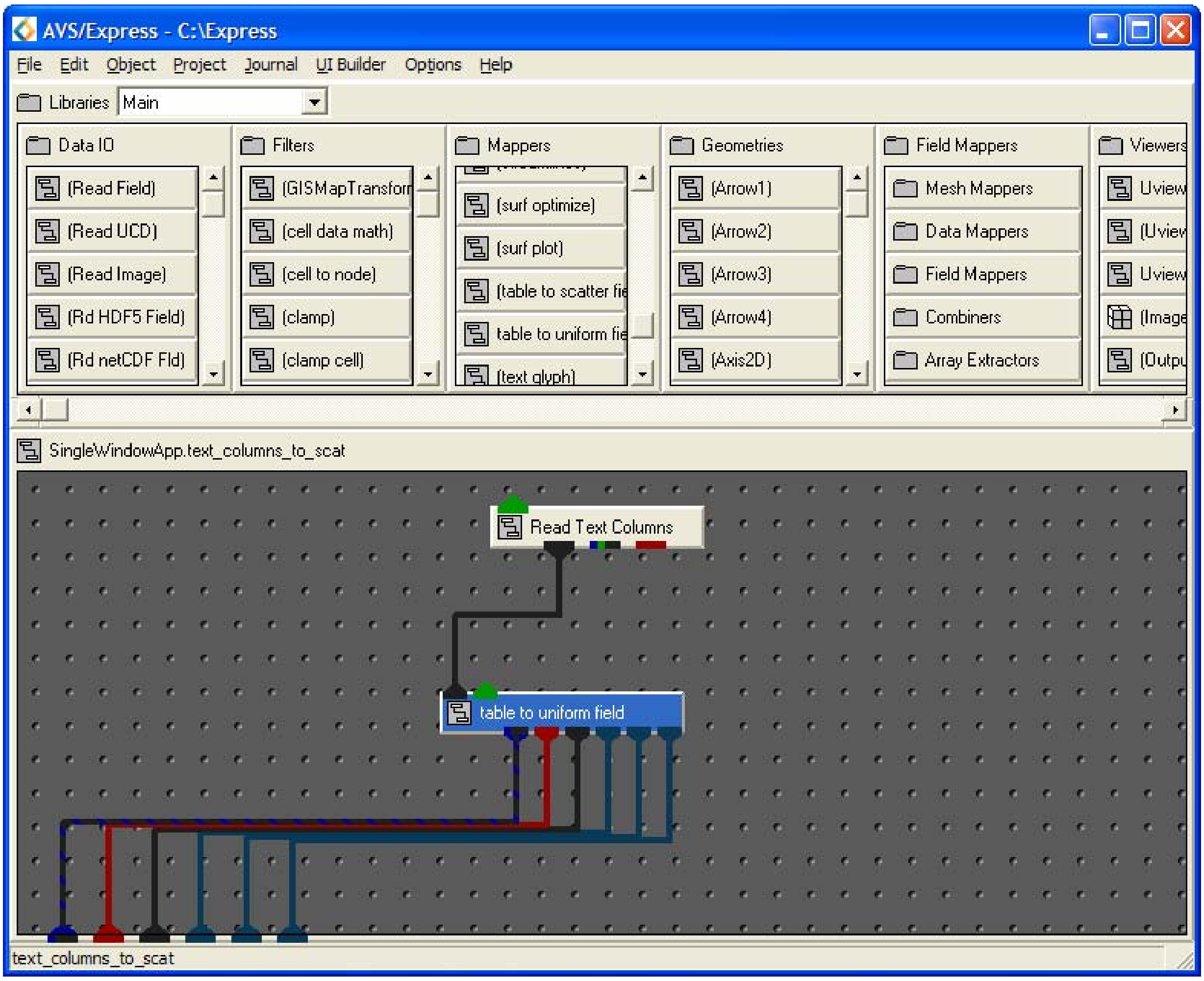

Figure C.30: New `text columns to uniform` Macro

- Left click on the bar at the top of the network area, from the pop up menu select **Close** to return to the initial view of the Network Editor.
- Reconnect the `text_columns_to_scat` macro to the `Uviewer3D` module as shown in Figure C.28.
- Right click on the `text columns to scat` macro and select **Rename** from the pop up menu. The **Rename Object** pop up window will appear as shown in Figure C.31.
- Rename the macro *text_columns_to_uniform* and select **Ok**. The improved macro will now have a suitable name in the network.

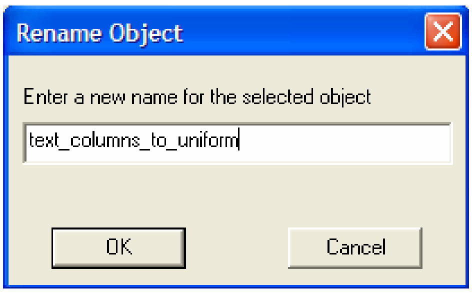

Figure C.31: **Rename Object** Pop up Window

C.4.5: Setting `table_to_uniform_field` Module Parameters

- Select the **Modules** command from the **Editors** drop down menu and select the `table to uniform field` module to open its Editor Panel as shown in Figure C.32. From here the axis of the module can be set.

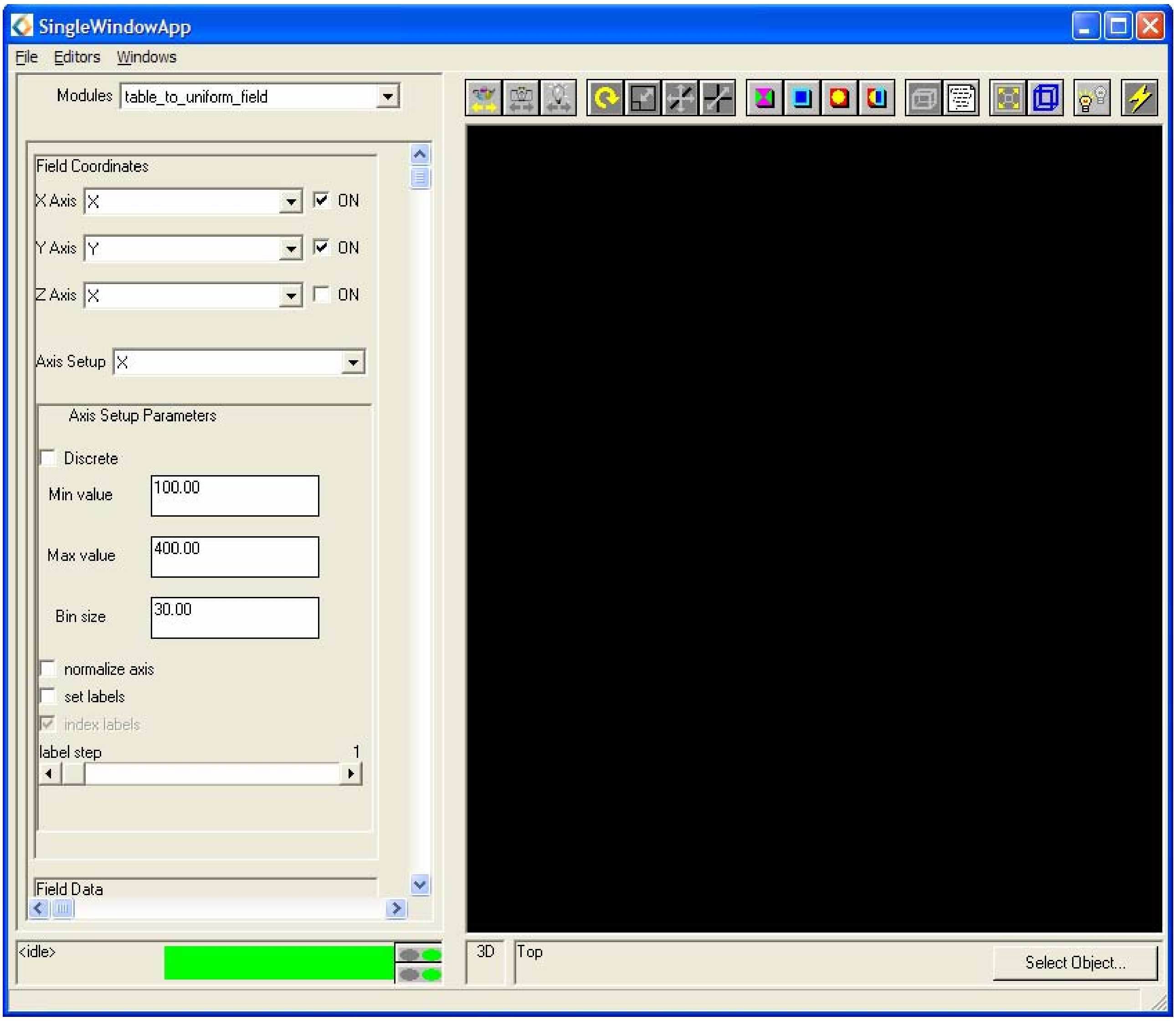

Figure C.27: `table_to_uniform_field` Editor Panel for **X** Coordinate

The following bullet points should provide enough information to set the correct parameters. It is still necessary to understand the structure of `SoilData.txt` and relate it to the parameters set (Appendix D).

- The **Field Coordinates** for this 2D data are the **X Axis** and the **Y Axis**.
- Ensure the **Axis Setup Parameters** for each coordinate are not set as **Discrete**, **normalize axis** or **set labels**. Using the **Axis Setup** dropdown list select the required coordinate and set its properties using the widgets in the indented panel below the **Axis Setup** drop down list (Figure C.32).
- Ensure the **Axis Setup Parameters** for each data component are not set as **Discrete**, **normalize axis** or **set labels**. Using the **Axis Setup** dropdown list select the required component and set its properties using the widgets in the indented panel below the **Axis Setup** drop down list (Figure C.33).

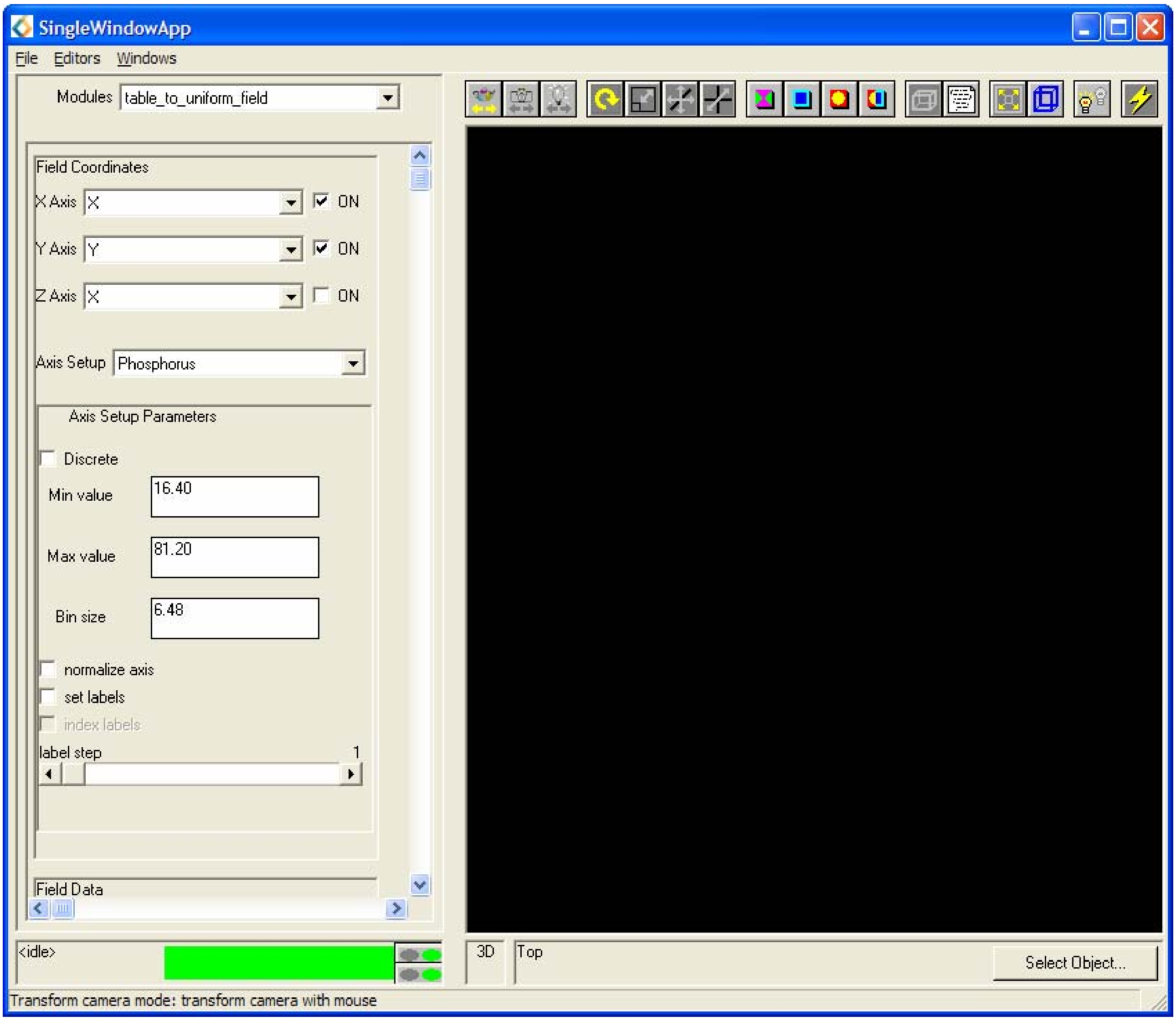

Figure C.33: `table_to_uniform_field` Editor Panel for **Phosphorus** Component

- The **Field Data** parameters should be set to those known to be data components; **Potassium**, **Magnesium**, **Phosphorus**, **Ph** and **OrganicMatter**[7]. Ensure that **Count Records** is not selected as **Field Data**
- The **Data Setup** type of each component should be set to **Average**

The Network Editor should resemble that shown in Figure C.34.

[7] If this was **Organic Matter** the Wizard would assume there where two different columns, must be written as **OrganicMatter** in the header line.

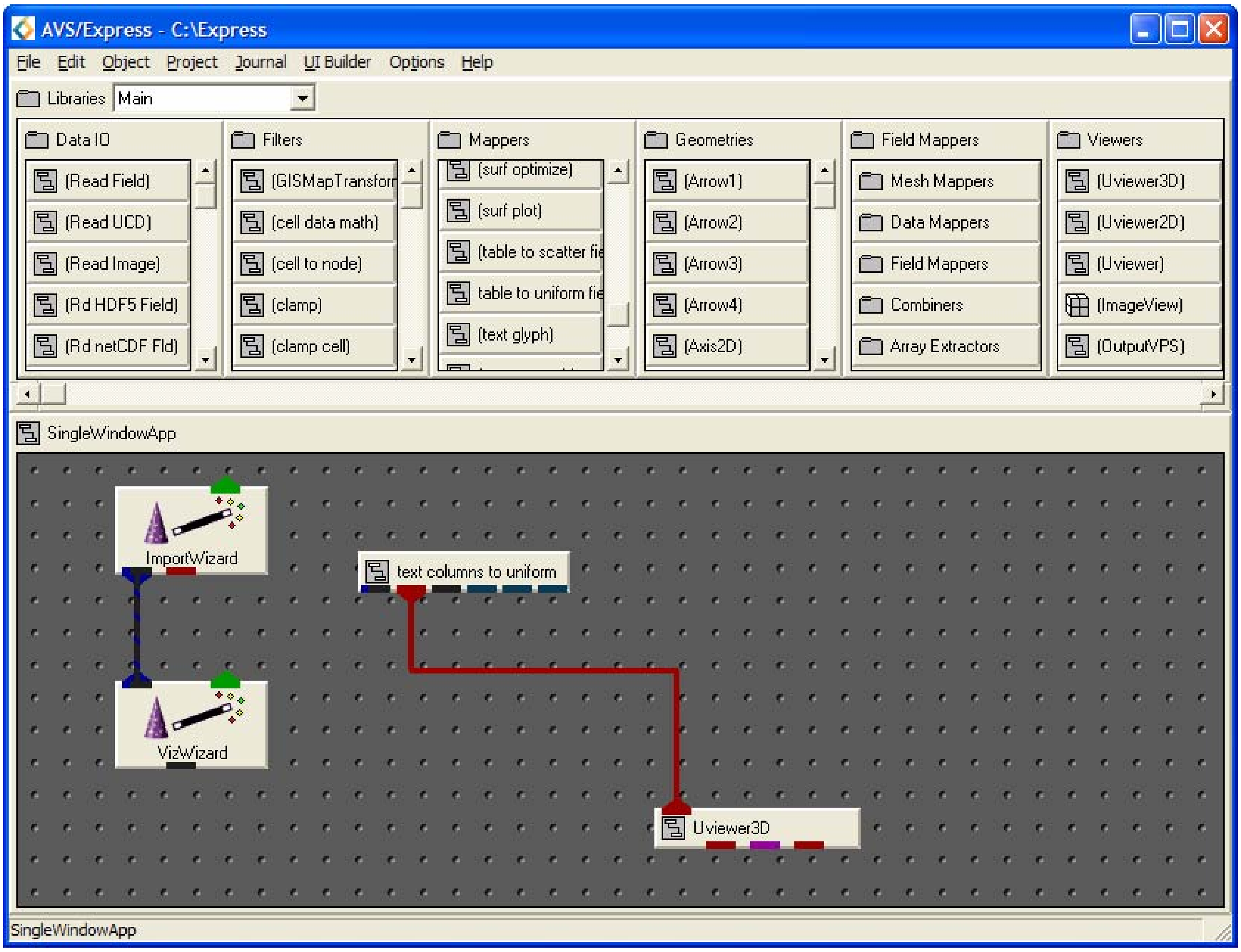

Figure C.34: Network produced by the `ImportWizard`

The image in the viewer should be similar to that shown in Figure C.35

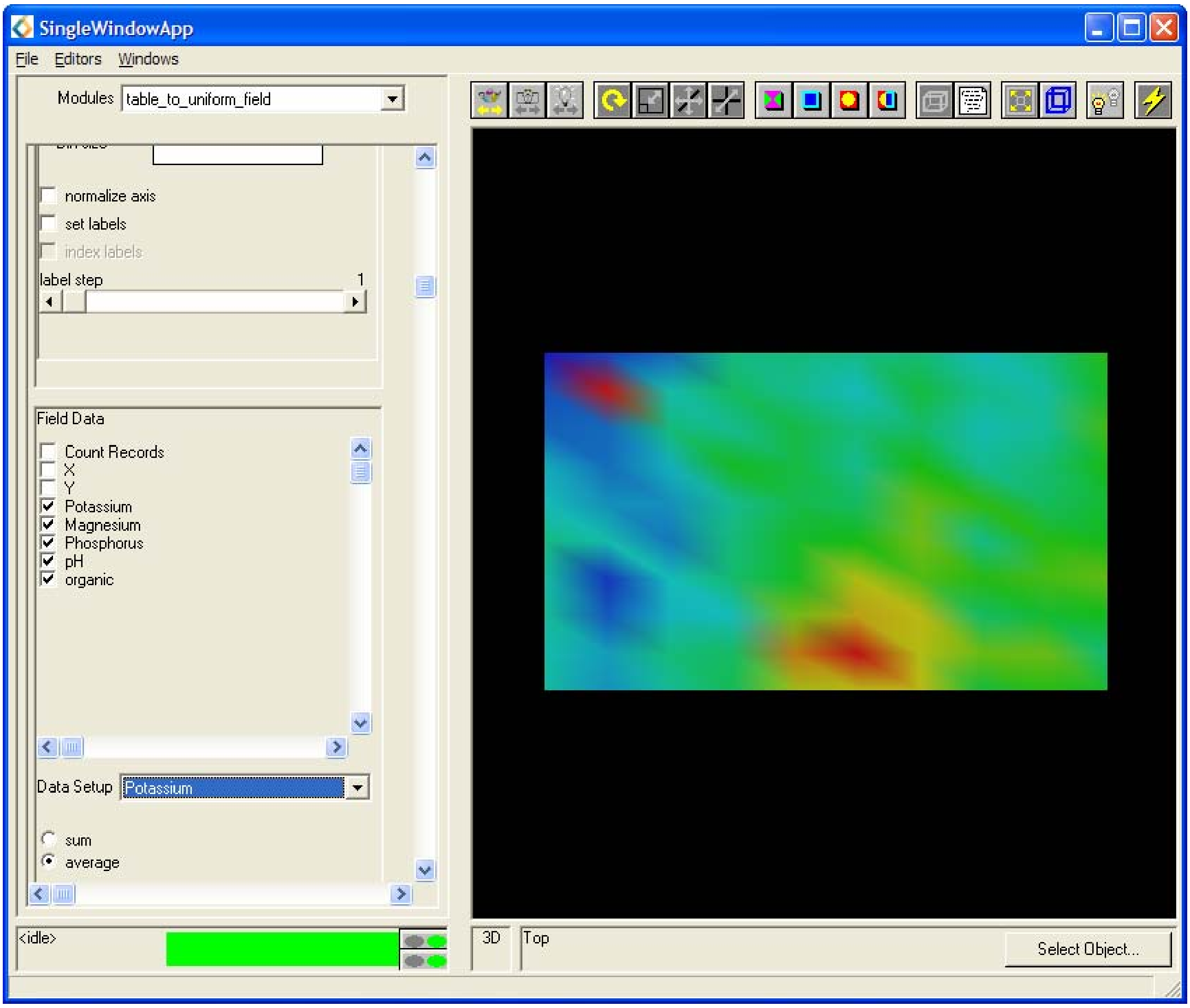

Figure C.35: Image Created by the `Data Import Wizard`

Now that an image has been obtained it is necessary to view it in a way that is useful and provides relevant information.

C.4.6: Visualizing the Data

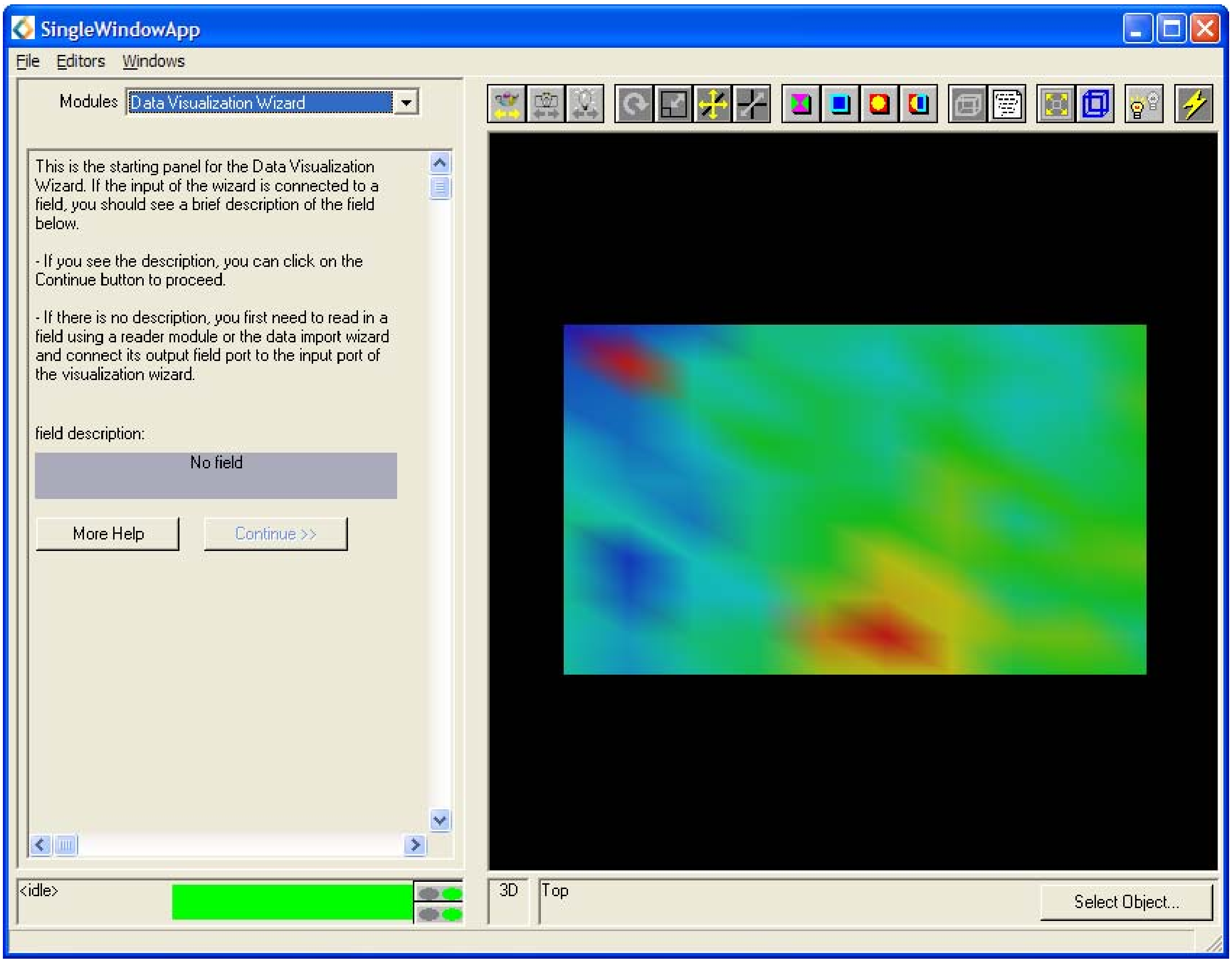

Figure C.36: `Data Visualization Wizard` Editor Panel

- Select the `Data Visualization Wizard` from the **Modules** drop down list, its Editor Panel will be displayed.

Usually **Continue** >> would be selected from the bottom of the Editor Panel, however in this case this is not possible[8]. There is a bug in the Wizard which has been reported to AVS/Express and can be fixed in a few simple steps:

- In the Network Editor double click `ImportWizard` to open up its network in the working area as shown in Figure C.37.

[8] In the `Data Visualization wizard` Editor Panel (Figure C.35) the **field description:** contains the statement **No field**. The Wizard is not receiving information regarding the `text columns to uniform` macro.

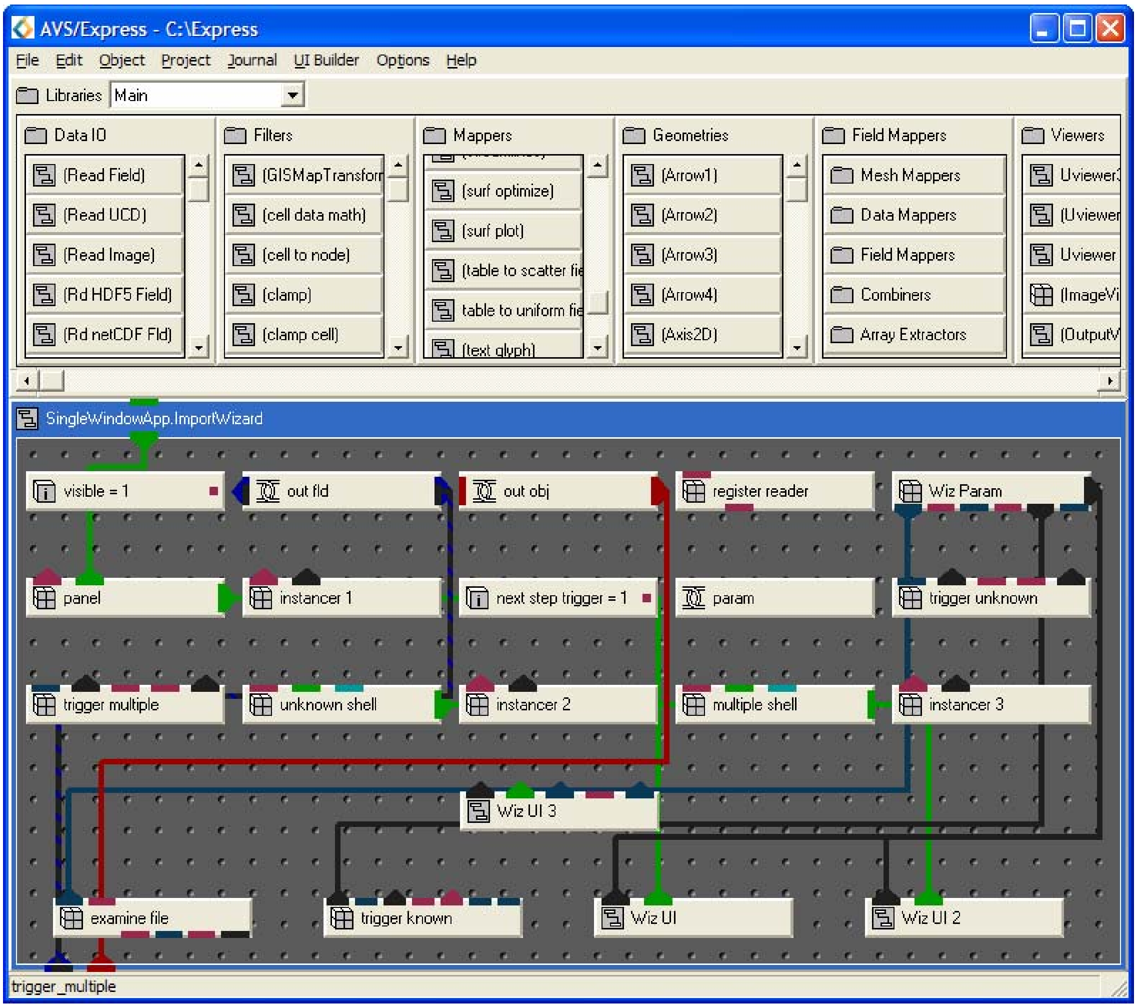

Figure C.37: `ImportWizard` Network

- Using the **left** mouse button double click on the module `out fld` in order to observe its contents[9]. A close up of the contents can be seen in Figure C.38.

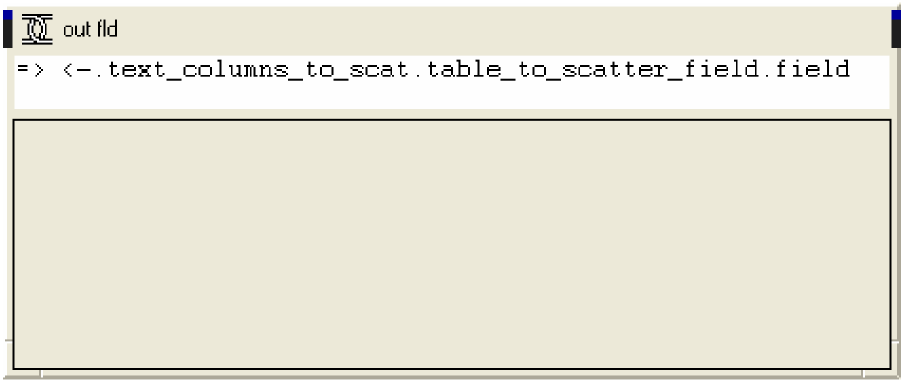

Figure C.38: `out fld` Module

[9] This is the link which connects `ImportWizard` and `VizWizard`. If this link is wrong, then information will not be passed between the two.

- There are three alterations, replace *scat* with *uniform*[10], *scatter* with *uniform*[11] and finally the last *field* with *out_fld*[12].
- Press the **Return** key on the keyboard in order to load up the correct icons as shown in Figure C.39.

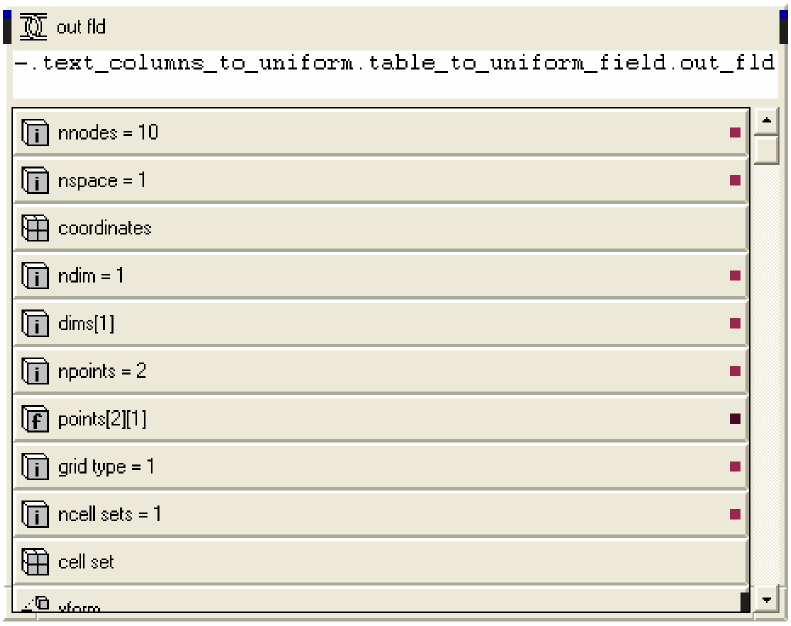

Figure C.39: Corrected `out_fld` Link

- **Right** click the top left corner of the `out_fld` module. From the pop up menu select **Close**
- **Right** click the top left hand corner of the `SingleWindowApp.ImportWizard` network. From the pop up menu select **Close**.

If the instructions have been followed correctly the bug should be fixed. It should now be possible to visualize the data using the Data Visualization Wizard.

- The parameters of the `Read_Text_Columns` and `table_to_scatter_field` modules may be reset to default when fixing the bug. Before you continue it is necessary to reset these parameters (Section C.4.4) and to obtain an image similar to that shown in Figure C.40.

---

[10] This refers to the macro which was originally `text_columns_to_scat`, and now is know as `text_columns_to_uniform`.

[11] This refers to the module `table_to_scatter_field` in the macro which was replaced by the `table_to_uniform_field` module.

[12] This is the bug, the `ImportWizard` assumes that there is an object called field inside the macro that gets instanced to read the file. In most cases this is likely to be alright, but in the case of the `text_columns_to_scat` and `text_columns_to_uniform` it is wrong

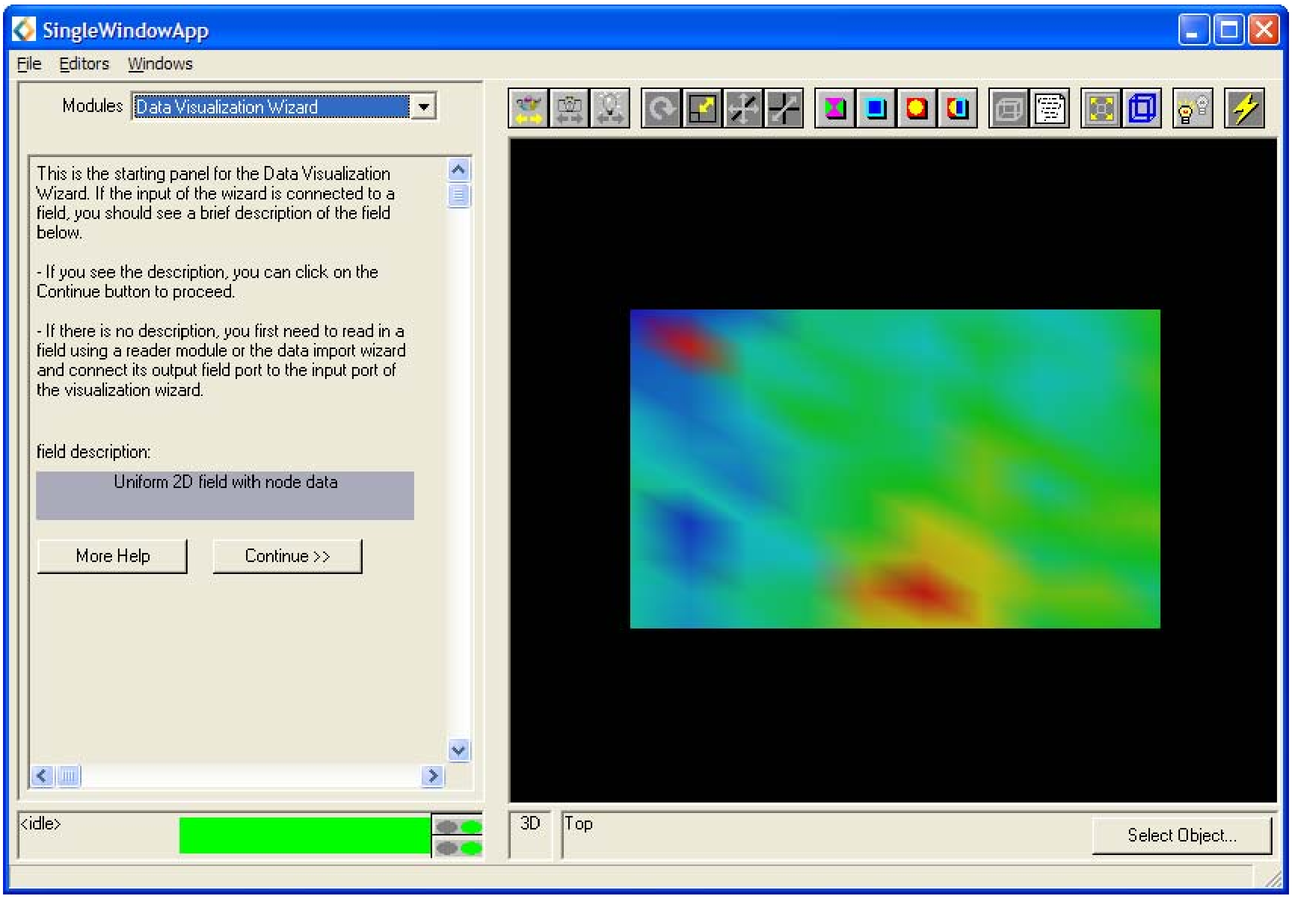

Figure C.40: `Data Visualization Wizard` Editor Panel

- In the **Modules** drop down list in the Data Viewer select `Data Visualization Wizard` to display its properties in the Editor Panel as shown in Figure C.40.
- Select **Continue** >> in order to load the **Data Viz Wizard** (Figure C.41), this is where the visualization method can be defined.

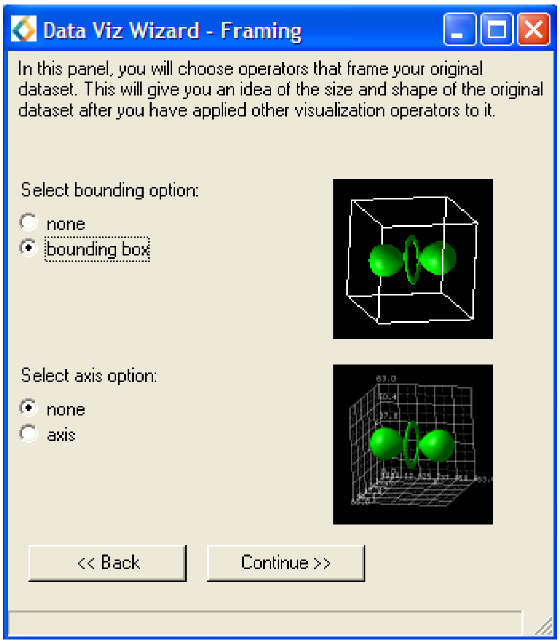

Figure C.41: **Data Viz Wizard** – Framing

- In the **Framing** window select the **bounding box** option, leave the other as default (**none**) and **Continue** >>.
- In the **Preprocessing** window leave all options as default (**none**) and select **Continue** >>.
- In the **Visualization** window select **solid_contour** and select **Continue** >>.
- Finally in the **Legend** window select **horizontal**, leave the other option as default (**none**) and select **Finish**.
- The data will not be visualized as a contour map[13] in the Scene window; however it will now have both a bounding box and a horizontal colour legend as shown in Figure C.42.

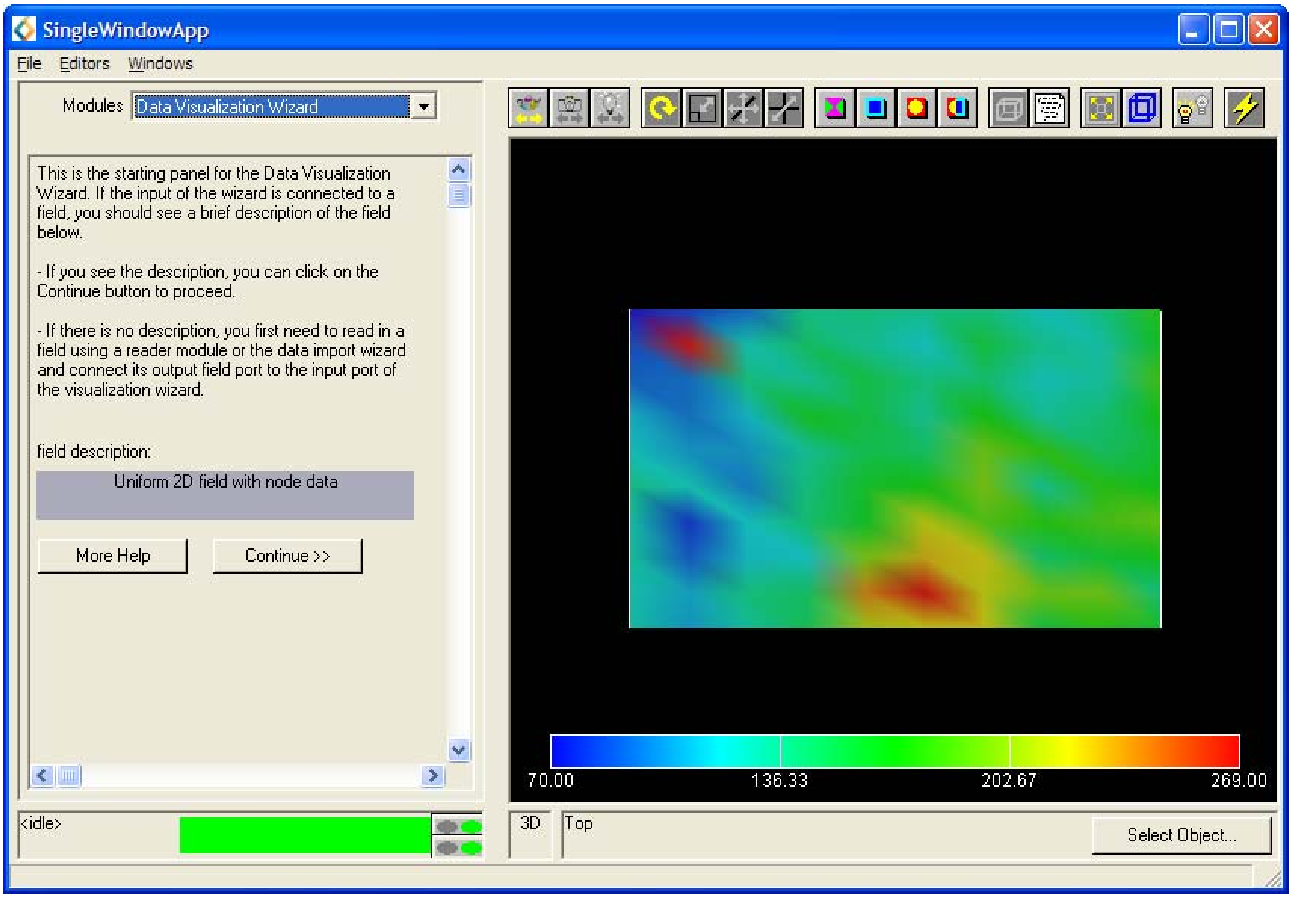

Figure C.42: Visualization produced using the `Data Visualization Wizard`

- In the Network Editor delete the connection that is directly between `text_columns_to_uniform` and `Uviewer3D`, so that it appears similar to that shown in Figure C.43.

[13] The reason for this is that the original connection between the macro `text_columns_to_uniform` and `Uviewer3D` still exists and is overriding the new network.

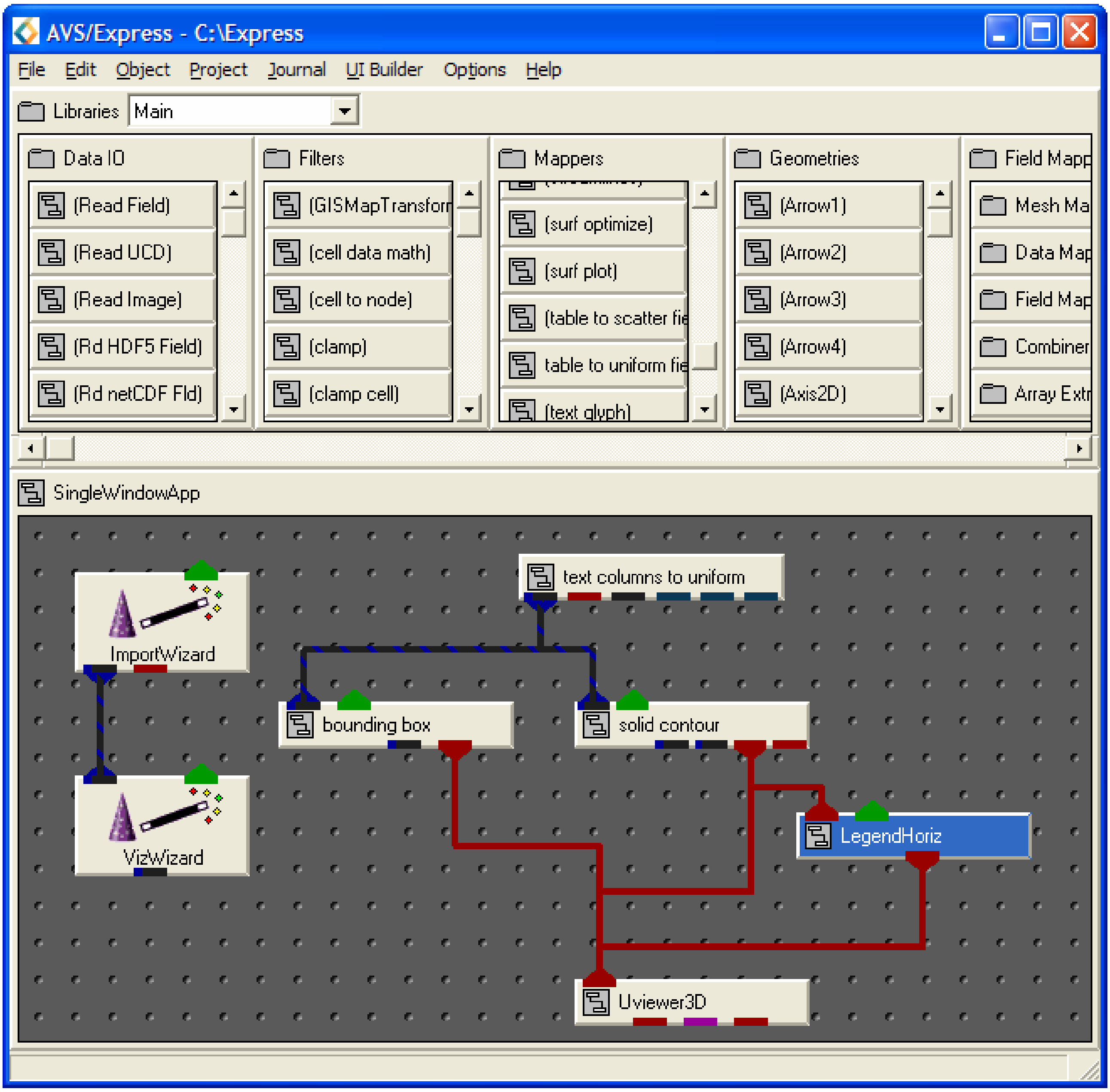

Figure C.43: Network Created by `VizWizard`

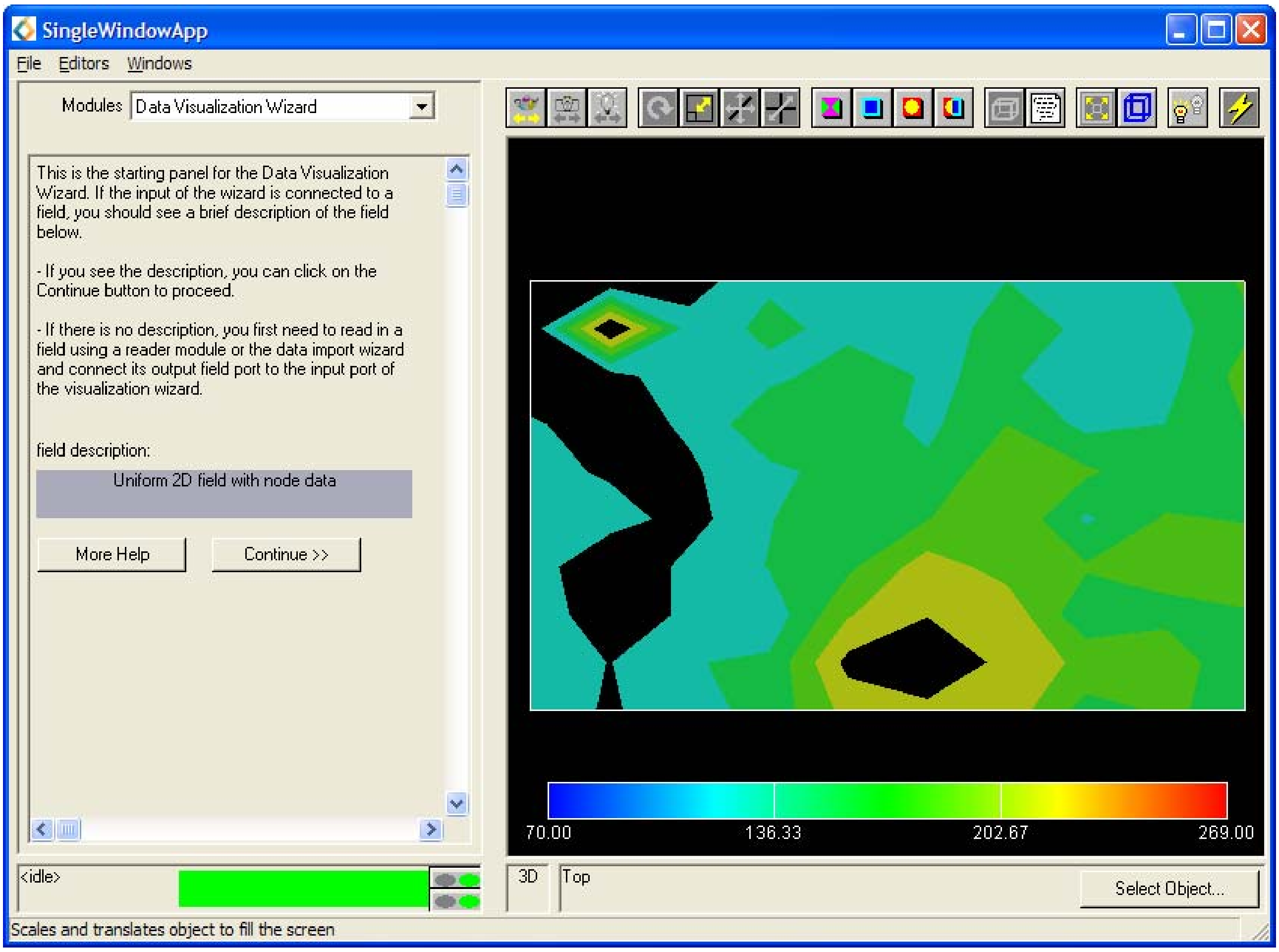

Figure C.44: Contour Map Visualization produced by the `Data Visualization Wizard`

The data should now be visualized as a contour map, with a bounding box and a horizontal legend as would be expected (Figure C.44). Now that a suitable visualization of the image has been obtained, it is necessary to manipulate the image into the desired format.

C.4.7: Interacting with the Objects

- The toolbar located above the DataViewer window, provides a quick way to manipulate images in the DataViewer.
- Click on the **Reset/Normalize/Center** icon, in order to suitably position the object in the window.
- Using the **scale**, **rotate** and **translate** icons manipulate the object until the scene appears similar to that shown in Figure C.45.
- Select `solid_contour` form the **Modules** dropdown list to display its parameters in the Editor Panel.
- Altering the **contour component** that is selected will alter the component which is shown in the Data Viewer Pad.
- Altering the value of the **number of contours** slider will alter the amount of contour regions produced.
- Altering the value of the **min level** slider and **max level** slider will alter the minimum and maximum values between which to extract data; the module extracts from the selected contour component all data values in this range.
- Selecting **Contour Lines** will create isolines.
- Alter these parameters to observe the effects and eventually gain a view similar to that shown in Figure C.45.

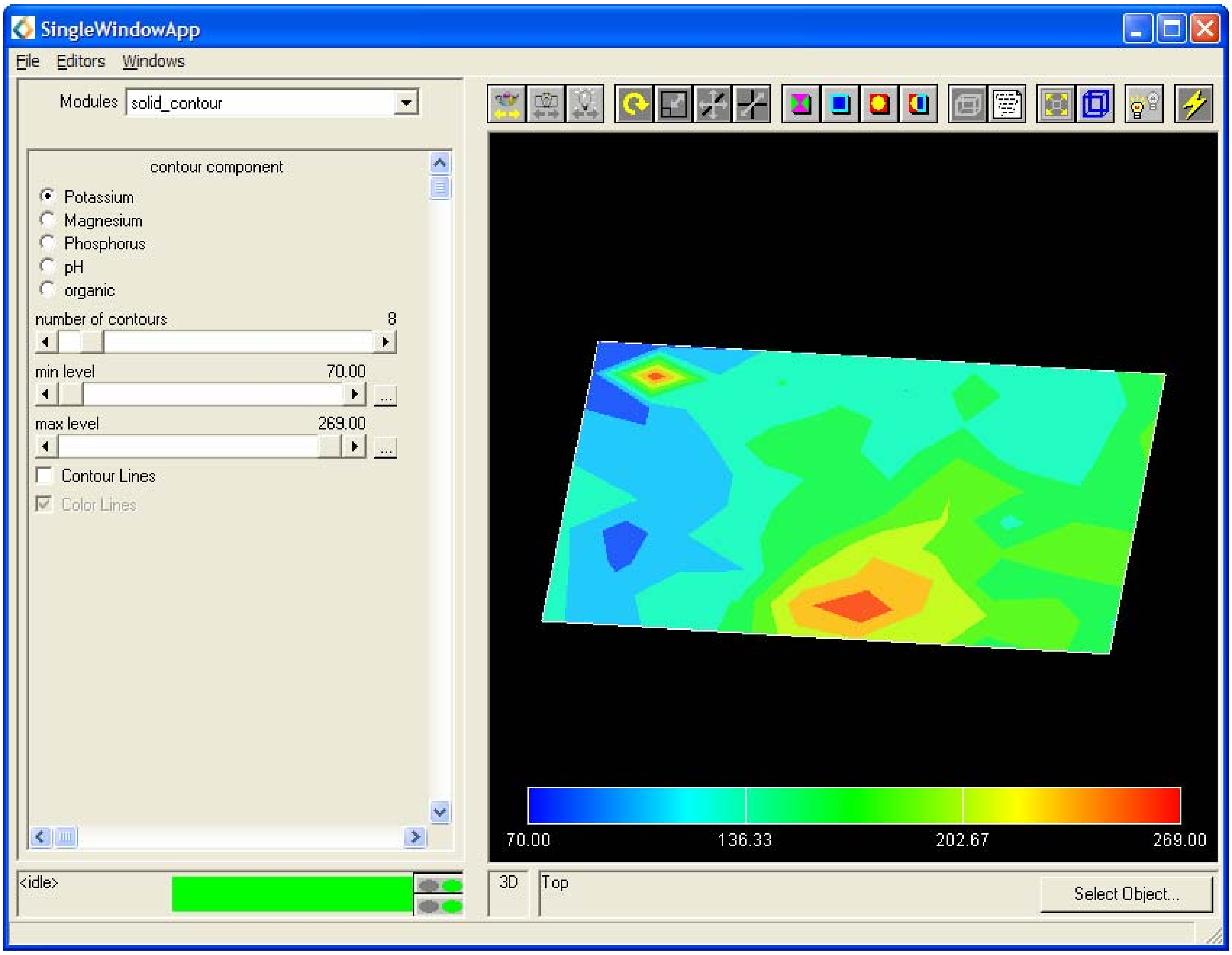

Figure C.45: `solid_contour` Editor Panel and Transformed Image

- Using the `Data Visualization Wizard` Editor Panel produce an image similar to that shown in Figure C.46. If necessary use trial and error in order to locate the correct modules.

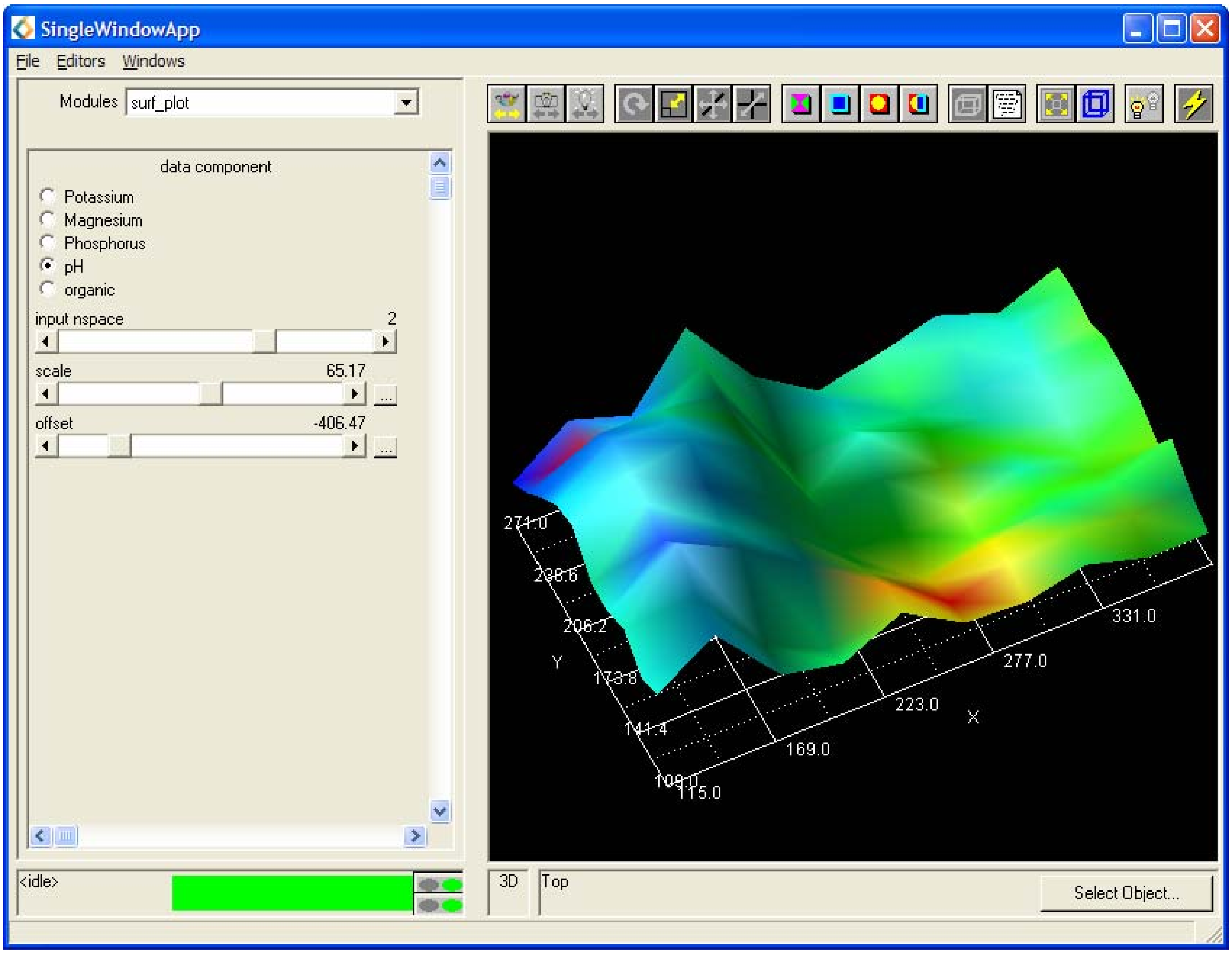

Figure C.46: `surf_plot` Editor Panel and Transformed Image

Appendix D: `SoilData.txt`

```
X Y Potassium Magnesium Phosphorus Ph OrganicMatter
100 100 128.00 61.00 29.00 7.80 4.40
100 120 113.00 77.00 37.00 7.60 4.43
100 140 163.00 104.00 35.60 7.80 4.99
100 160 147.00 128.00 33.00 7.70 5.05
100 180 152.00 146.00 41.40 7.40 4.99
100 200 134.00 145.00 36.20 7.60 4.84
100 220 114.00 137.00 41.00 7.50 4.65
100 240 79.00 126.00 34.80 7.20 4.17
100 260 67.00 123.00 35.60 7.30 4.27
100 280 59.00 114.00 31.40 6.80 4.09
120 100 94.00 60.00 38.80 7.10 4.49
120 120 129.00 64.00 39.00 6.90 4.29
120 140 96.00 77.00 31.20 7.00 4.42
120 160 106.00 106.00 36.80 6.80 4.69
120 180 119.00 122.00 36.40 7.10 4.46
120 200 107.00 140.00 41.00 7.60 4.72
120 220 113.00 134.00 37.20 7.50 4.80
120 240 103.00 186.00 43.40 7.30 4.86
120 260 87.00 163.00 36.40 7.00 4.61
120 280 81.00 154.00 38.00 6.70 4.35
140 100 106.00 48.00 48.20 7.70 4.62
140 120 109.00 49.00 34.60 7.80 4.64
140 140 89.00 45.00 35.60 8.10 4.54
140 160 68.00 46.00 23.60 7.90 4.06
140 180 123.00 78.00 33.00 7.90 4.70
140 200 89.00 97.00 31.80 8.00 5.08
140 220 69.00 83.00 29.40 8.10 5.18
140 240 94.00 134.00 47.80 7.70 5.09
140 260 429.00 268.00 48.20 7.20 4.50
140 280 82.00 135.00 32.60 7.20 4.17
160 100 101.00 58.00 35.20 7.30 3.69
160 120 125.00 67.00 32.80 7.20 4.43
160 140 93.00 53.00 19.80 7.80 4.33
160 160 96.00 58.00 21.00 7.80 4.62
160 180 77.00 62.00 16.40 7.60 4.01
160 200 88.00 86.00 18.80 7.90 3.88
160 220 104.00 110.00 20.00 7.90 3.96
160 240 142.00 150.00 19.80 7.70 4.43
160 260 97.00 100.00 18.20 8.00 4.24
160 280 96.00 100.00 19.60 7.80 4.17
180 100 158.00 99.00 32.80 6.60 3.83
180 120 151.00 71.00 31.00 6.70 4.27
180 140 139.00 71.00 28.00 7.70 4.48
180 160 117.00 56.00 25.20 8.00 4.50
```

```
180 180 119.00 65.00 27.20 8.00 4.49
180 200 123.00 54.00 19.00 8.10 4.10
180 220 125.00 59.00 27.20 7.90 4.45
180 240 133.00 82.00 26.20 7.70 4.32
180 260 126.00 101.00 40.00 6.70 4.23
180 280 98.00 127.00 22.20 6.80 4.09
200 100 149.00 135.00 37.20 6.80 3.75
200 120 126.00 68.00 41.80 6.80 5.78
200 140 119.00 37.00 37.40 7.30 4.13
200 160 141.00 50.00 39.00 7.30 4.92
200 180 138.00 50.00 30.00 7.40 4.57
200 200 132.00 53.00 38.80 7.60 4.76
200 220 164.00 58.00 42.80 7.60 4.77
200 240 132.00 73.00 38.80 7.60 4.25
200 260 146.00 92.00 42.60 7.60 3.92
200 280 136.00 54.00 38.80 7.70 4.77
220 100 173.00 96.00 52.00 7.30 4.41
220 120 238.00 57.00 45.40 7.40 3.87
220 140 198.00 71.00 48.80 7.20 4.85
220 160 143.00 64.00 54.00 6.90 3.91
220 180 188.00 60.00 65.40 6.60 4.71
220 200 170.00 83.00 55.60 6.50 4.92
220 220 166.00 75.00 51.60 6.50 5.01
220 240 182.00 166.00 55.40 6.40 4.88
220 260 138.00 260.00 40.20 7.10 4.73
220 280 154.00 295.00 34.20 7.20 5.11
240 100 247.00 140.00 56.80 6.90 4.65
240 120 239.00 130.00 66.00 6.80 4.12
240 140 196.00 83.00 56.00 6.60 4.74
240 160 184.00 168.00 54.80 6.50 4.42
240 180 172.00 147.00 43.60 6.60 4.94
240 200 132.00 150.00 33.40 6.70 3.88
240 220 123.00 155.00 41.20 6.70 4.90
240 240 125.00 172.00 42.60 6.60 4.62
240 260 127.00 175.00 50.00 6.50 4.65
240 280 122.00 100.00 44.40 6.50 4.49
260 100 217.00 113.00 59.40 6.70 4.97
260 120 269.00 116.00 70.00 6.60 4.94
260 140 227.00 100.00 71.00 6.50 4.05
260 160 215.00 170.00 66.80 6.20 4.32
260 180 194.00 170.00 63.00 6.40 5.15
260 200 146.00 160.00 47.40 6.70 5.17
260 220 133.00 180.00 45.60 6.90 5.54
260 240 137.00 166.00 52.40 6.60 5.10
260 260 119.00 155.00 47.80 6.60 4.46
260 280 134.00 100.00 54.40 6.40 4.94
280 100 229.00 116.00 79.00 6.80 4.59
280 120 216.00 175.00 81.00 6.70 4.70
280 140 244.00 113.00 70.60 6.70 4.65
```

```
280 160 189.00 167.00 68.80 6.40 4.68
280 180 202.00 210.00 55.40 6.40 4.81
280 200 198.00 233.00 56.60 6.60 5.22
280 220 180.00 182.00 65.80 6.70 5.32
280 240 144.00 138.00 64.20 6.50 4.78
280 260 150.00 158.00 64.20 6.40 5.00
280 280 132.00 160.00 54.80 6.40 5.31
300 100 208.00 100.00 75.60 6.50 4.61
300 120 186.00 96.00 66.40 6.70 4.39
300 140 196.00 110.00 65.80 6.70 4.52
300 160 185.00 129.00 60.80 6.40 4.69
300 180 191.00 139.00 59.60 6.40 4.40
300 200 170.00 126.00 59.40 6.60 4.61
300 220 141.00 122.00 52.60 6.50 4.48
300 240 143.00 123.00 54.60 6.50 4.36
300 260 141.00 137.00 52.80 6.50 4.79
300 280 141.00 162.00 59.20 6.60 5.23
320 100 157.00 102.00 66.40 6.80 4.63
320 120 193.00 115.00 72.40 6.70 4.18
320 140 145.00 94.00 62.00 6.80 4.44
320 160 167.00 123.00 53.60 6.80 4.54
320 180 135.00 102.00 53.20 6.60 4.33
320 200 175.00 131.00 60.20 6.50 3.96
320 220 131.00 109.00 52.80 6.50 3.48
320 240 122.00 119.00 49.20 6.50 4.25
320 260 133.00 142.00 44.60 6.70 4.49
320 280 120.00 167.00 56.60 6.70 4.58
340 100 159.00 108.00 72.60 6.70 3.69
340 120 200.00 88.00 81.20 6.70 4.21
340 140 167.00 114.00 68.60 6.70 4.24
340 160 168.00 129.00 60.80 6.70 4.39
340 180 157.00 105.00 67.20 6.50 4.11
340 200 125.00 99.00 64.40 6.50 4.36
340 220 143.00 111.00 53.00 6.40 3.97
340 240 141.00 135.00 53.20 6.30 4.54
340 260 108.00 127.00 50.00 6.70 4.75
340 280 142.00 168.00 55.40 6.50 3.88
360 100 189.00 86.00 67.00 6.50 4.00
360 120 188.00 86.00 64.80 6.80 4.45
360 140 151.00 77.00 51.20 6.90 4.34
360 160 198.00 120.00 58.80 6.50 4.63
360 180 190.00 123.00 64.20 6.40 4.87
360 200 168.00 113.00 65.40 6.50 4.22
360 220 143.00 100.00 53.60 6.70 4.38
360 240 134.00 122.00 54.40 6.70 4.48
360 260 144.00 157.00 60.80 6.60 4.35
360 280 131.00 163.00 63.20 6.40 4.32
380 100 138.00 76.00 54.20 6.60 3.34
380 120 127.00 53.00 47.60 6.90 3.76
```

```
380 140 175.00 79.00 61.40 7.10 4.15
380 160 190.00 98.00 57.40 6.70 4.47
380 180 153.00 84.00 48.80 6.90 4.67
380 200 150.00 96.00 51.40 6.80 4.32
380 220 139.00 85.00 46.00 6.90 4.22
380 240 166.00 114.00 47.60 6.80 4.42
380 260 133.00 120.00 40.80 7.00 4.48
380 280 164.00 156.00 46.00 6.80 4.73
400 100 181.00 81.00 62.20 6.60 3.99
400 120 152.00 74.00 59.00 6.60 4.11
400 140 155.00 76.00 52.40 6.80 4.06
400 160 188.00 91.00 55.60 6.40 4.20
400 180 182.00 112.00 50.20 6.70 4.57
400 200 170.00 87.00 43.40 6.90 4.65
400 220 200.00 106.00 48.40 6.80 4.93
400 240 213.00 125.00 52.80 6.60 4.67
400 260 194.00 133.00 46.20 6.50 4.40
400 280 184.00 162.00 50.60 6.60 4.39
120 110 161.00 57.00 37.60 7.00 4.62
120 130 131.00 64.00 31.00 6.90 4.17
190 120 175.00 69.00 47.00 6.60 4.14
210 120 135.00 59.00 42.20 6.50 4.52
290 120 245.00 124.00 77.40 6.90 4.83
310 120 194.00 101.00 55.00 7.00 4.48
150 160 90.00 46.00 18.00 7.70 4.81
170 160 145.00 388.00 31.20 7.20 4.58
310 160 214.00 127.00 51.80 6.50 4.12
330 160 139.00 102.00 43.40 6.60 3.90
380 150 168.00 75.00 49.40 7.20 4.30
380 170 196.00 98.00 47.20 6.80 4.23
230 180 127.00 77.00 41.00 6.30 4.63
250 180 143.00 123.00 42.80 6.40 4.42
140 190 122.00 76.00 32.00 7.80 4.76
140 210 122.00 84.00 42.00 7.80 4.99
280 190 212.00 145.00 60.00 6.40 4.32
280 210 186.00 171.00 55.40 6.70 4.44
220 210 174.00 119.00 57.00 6.20 4.86
220 230 155.00 114.00 61.40 6.50 5.07
340 210 134.00 93.00 53.60 6.50 4.26
340 230 110.00 109.00 40.60 6.60 4.13
370 240 169.00 105.00 55.60 7.10 3.98
390 240 177.00 100.00 51.40 6.30 3.65
110 260 103.00 144.00 42.80 6.60 4.70
130 260 106.00 134.00 44.60 7.00 4.95
180 250 150.00 158.00 31.00 6.70 4.21
180 270 87.00 116.00 30.00 6.30 4.30
300 250 177.00 117.00 64.20 6.40 4.92
300 270 146.00 132.00 58.60 6.60 4.78
```