Working with FieldView

SOFTWARE RELEASE
VERSION 2023

11/7/2023

Tecplot, Inc.

Table of Contents

Chapter 3
Basic Operations 101

Chapter 4
Computational Surface 161

List of Figures

Chapter 1

XDB Work-flows for CFD

1

FieldView is able to import as well as write XDB files. An XDB file can be managed just like any other dataset in **FieldView**, and can be manipulated in the following ways:

1. Datasets can be scaled, rotated, translated, mirrored or rotationally duplicated

2. Multiple copies of a surface can be created; each surface can be independently transformed, scaled and/or rotated

3. 2D plots can be created on surfaces; surfaces can be probed

4. XDB datasets can be appended and Dataset Comparison can be used (please refer to the **Working with XDBs Tutorial** in the **User's Guide**)

5. Linked Surface sweeps and Merged Transient sweeps can be performed

6. Geometric functions "X", "Y" and "Z", etc., are always present in the data so need not (cannot) be selected for export. Quantities "I", "J", and "K" (for Structured cases) are not available for export when creating XDB files because they are reserved names in FieldView. They *can* however be exported by creating formulas, for example `myI="I"`, `myJ="J"`, `myK="K"`.

7. New XDB extracts can be created from XDB datasets (useful for limiting time ranges for example)

Note that Vector components do not need to be individually saved when creating XDB files. If a vector quantity is saved, all components of the vector are available for scalar coloring and thresholding.

To see how this is relevant to your workflows, consider a normal postprocessing session, illustrated in **Figure 1**. Typically, you read in a dataset and create several postprocessing objects (surfaces and rakes) to compose a visual representation of your CFD dataset. Next, you may create various images and animations for later review. You may also interrogate some of these surfaces to calculate things such as forces, loads, moments and heat fluxes.
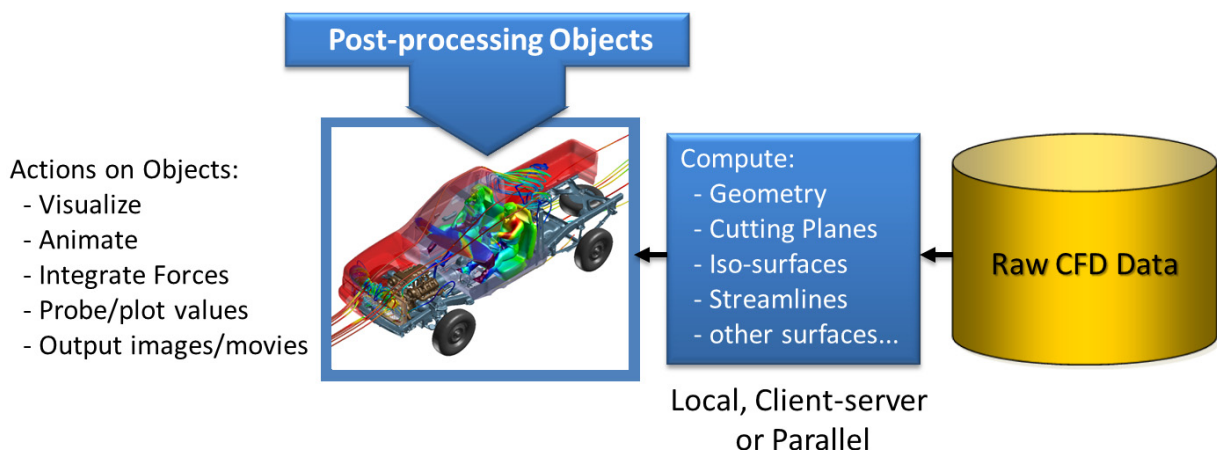
Figure 1 Typical Postprocessing Work Session

After creating the postprocessing objects from your initial postprocessing session, you write and later import the XDB dataset containing your postprocessing objects.
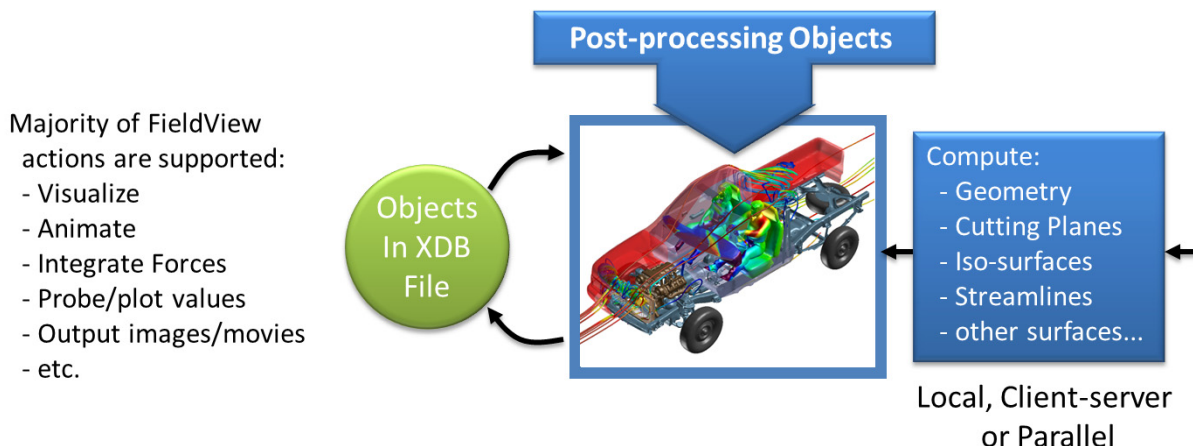
Figure 2 Postprocessing with XDB files

As noted above, an XDB dataset can be manipulated in the same was as any dataset, allowing you to create new representations of your postprocessing objects, and conduct quantitative postprocessing as well. This approach is particularly well suited to working with transient datasets, outlined in **Figure 3** below. We recommend that you run **Field-View** in batch mode to create the XDB file for the entire transient run. This can be done by first reading a time step, and running **FieldView** interactively to create your postprocessing objects of interest. Then, in batch, you can use your RESTARTS with either a SCRIPT or an **FVX** script to create the XDB file.
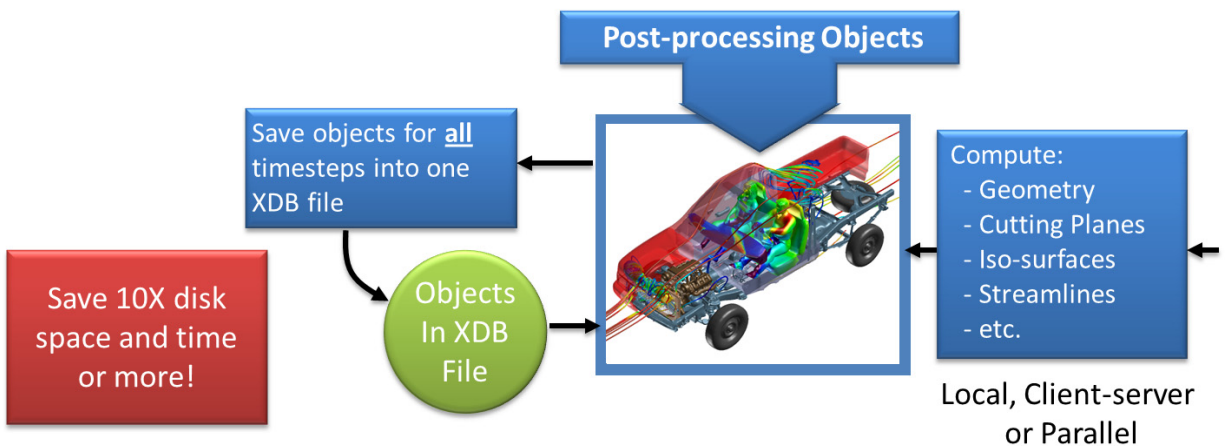
Figure 3 Transient XDB postprocessing

Once your XDB is written, you can import your XDB file interactively, to postprocess your results further.
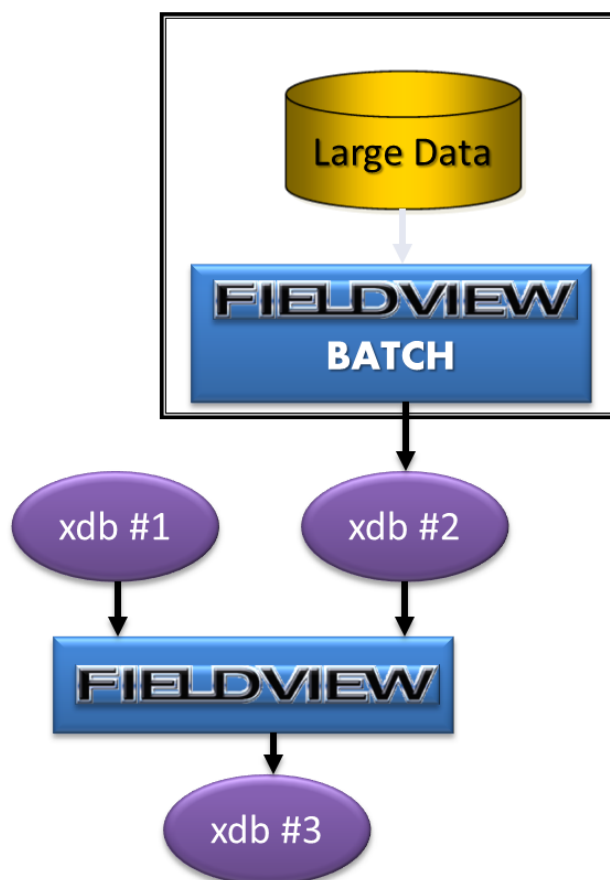
Figure 4 Overall XDB Workflow

In **Figure 4**, we illustrate the flexibility of this XDB workflow approach. It is possible to run **FieldView** in batch creating an XDB file (in this case, xdb #2). This file, from a previous batch session, can be read interactively, and compared with a different XDB file (xdb #1). From these comparisons, it is possible to create a third XDB dataset (xdb #3), based on a numerical difference of the first two, which can then be further postprocessed.

An XDB file is imported from the Data Input menu, as shown in **Figure 5**. An XDB file may be imported using either the direct or client-server modes. XDB files can not be imported using parallel servers unless read via PFPR **Partitioned File Parallel Reader (PFPR)** layout files. However XDB files tend to be small and can be efficiently read without the need for parallel. This menu is also where you can select *Local Parallel* or your own custom server configuration as explained on page 19 in the Installation Guide.
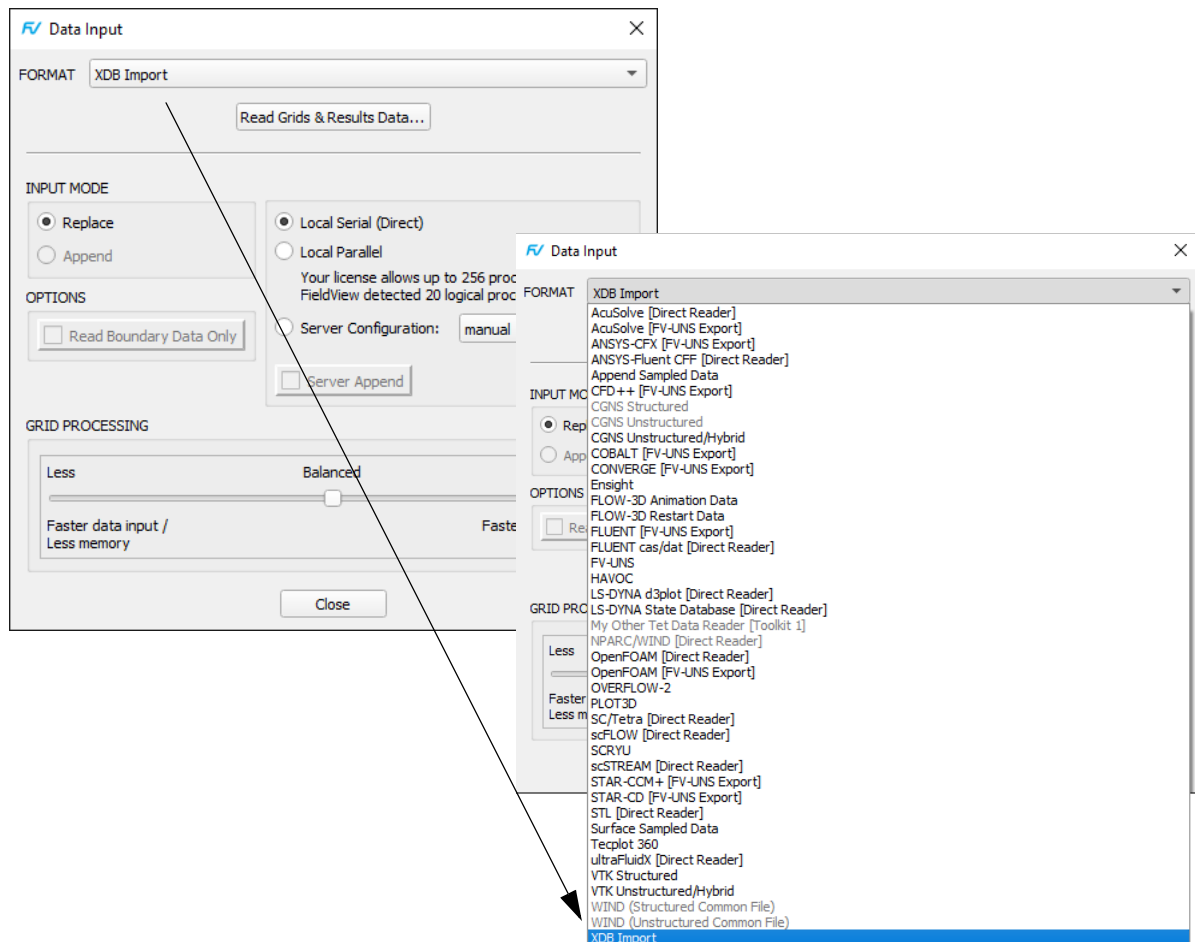
Figure 5 XDB Import

The XDB reader has been extended in **FieldView 19** to support the new XDB format introduced with XDBLib 2.0 (only supported on Linux64). XDBLib 2.0 is **FieldView's** solution for in situ post-processing. It's a library that allows solvers with native in situ post-processing and the ones instrumented for in situ thanks to libsim which writes directly to XDB extracts, without having to write full 3D datasets. This is particularly interesting for large and transient simulations, for which full 3D mesh and results are too large and take too long to process. If you're interested into getting access to the XDBLib 2.0 library or in getting help in instrumenting your solver for in situ post-processing, please contact **Tecplot, Inc**.

There are essentially four types of XDB files:

1. Steady state, which is a snapshot of all surfaces and rakes

2. Surface sweep, containing a sweep of a coordinate, computational or iso-surface (and possibly other surfaces)

3. Transient sweep, which is a sweep saving all time steps to a single file

4. Transient sweep, saving each time step to a single file to create a series of files

Although information is available within the XDB files to describe what they were originally, all surfaces are stored as boundary types. A boundary type naming convention has been established to create some correspondence between the original surface type and its name within the XDB file. An illustration of the boundary type naming convention is shown in **Figure 6**. In general, the first part of the boundary type name corresponds to the type of surface it was originally derived from. The remaining part of the name attempts to provide additional context for that surface up to an 80 character limit.
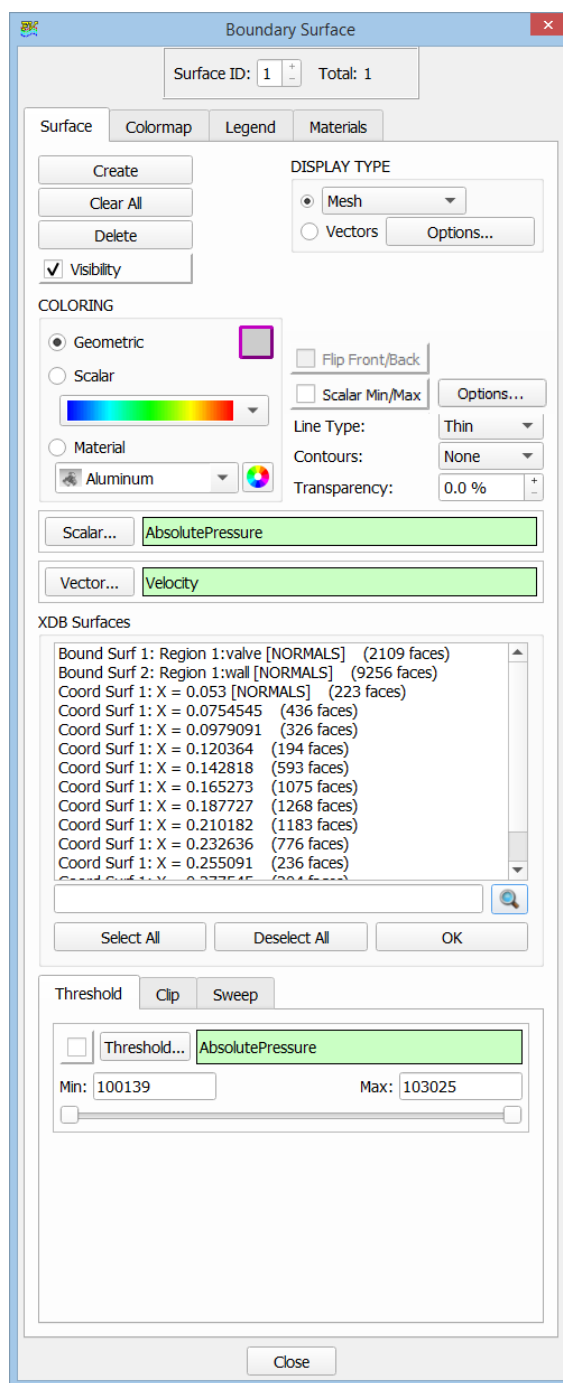
Figure 6 XDB Boundary Surfaces

Streamline, streakline or particle path data may also be present in XDB files. This type of data is read into **FieldView** as particle paths, and a separate step is required. The Parti-cle Path input panel, when opened, automatically points to the XDB file which has

already been read in. The Particle Path Data Input panel (see **Figure 7**) can be used to read data directly or with a server.
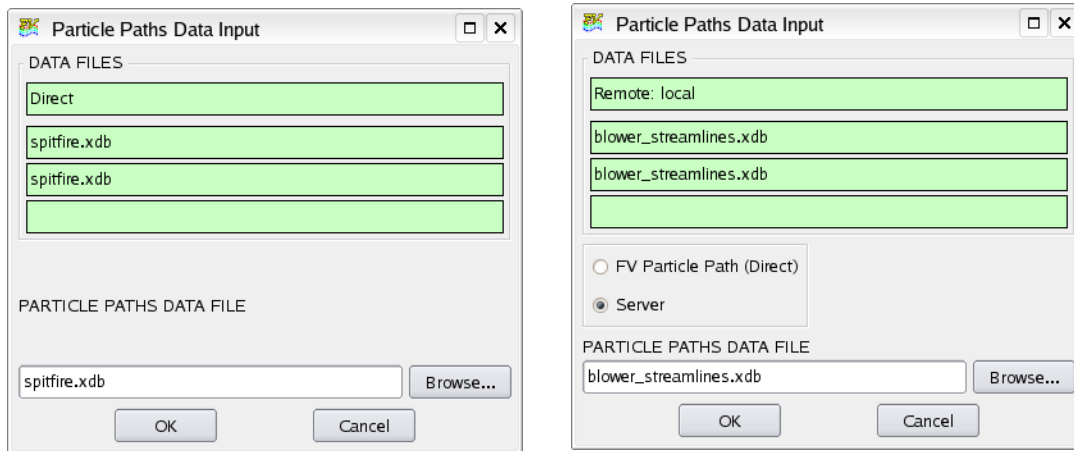
Figure 7 XDB Particle Path Data Input

If you attempt to read particle path data from an XDB file which does not have any, you will be prompted with the following error message:
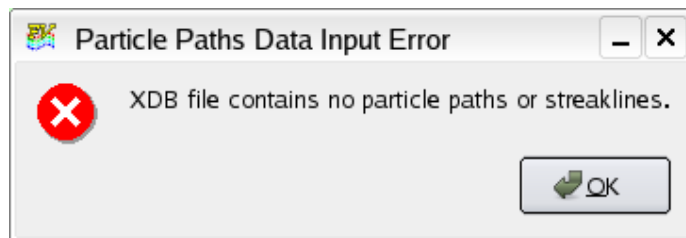
Figure 8 XDB Particle Path Data Input Error

In the case where an XDB file contains both streamlines or streaklines and particle path data, only the particle path is read since **FieldView** currently only supports one particle path per dataset.

Full **FVX** support has been provided to import XDB files. A sample **FVX** listing below illustrates the correct syntax.

```
local dataset_1_info_table = read_dataset( {
    data_format = "xdb_import",
    input_parameters = {
        name = "/usr3/xdb/spitfire.xdb",
        options = {
            input_mode = "replace",
```

```
                boundary_only = "off"
            } -- options
        } -- input_parameters
    } ) -- read_dataset

    local particle_paths_1 = read_particle_paths(
        {
            scalar_colormap = {
                name = "spectrum",
                invert = "off",
                filled_contour = "off",
            }, -- scalar_colormap
            ppath_func = "Duration",
            format = "xdb import",
            number_of_contours = 16,
            geometric_color = 4,
            line_type = "thin",
            scalar_range = {
                min = 0,
                local_min = 0,
                max = 0.2000000029802322,
                local_max = 0.2000000029802322,
                use_local = "off",
            }, -- scalar_range
            select_by = {
                initial_value_variable = "none",
                value_on_path_variable = "none",
                tags = "none",
            }, -- select_by
            visibility = "on",
            scalar_func = "none",
            dataset = 1,
            filename = "/usr3/xdb/spitfire.xdb",
        }
    ) -- particle_paths_1
```

Function Selection for XDB Export

Two methods have been available to specify the functions to be saved when exporting an XDB file via either the Write or Build mode. The first method lists the functions to be exported in a file called `xdb_vars`. A basic version of this file is included in the standard **FieldView** installation in the `/fv/data` subdirectory. Any modified version of this file, placed in a user's `HOME` directory, would be used instead of the basic version of this file supplied with **FieldView**. As an alternative to using the `xdb_vars` file, **FieldView** checks for a file called `root.xfn`, which has a list of variable names, one per line, when

saving an XDB file with the base name `root`. The `root.xfn` search path starts in the directory where the XDB file is being saved, then goes to the users `HOME` directory and finally looks in the **FieldView** installation area (`/fv/data`).

To simplify the task of selecting the functions to be saved when exporting an XDB, a GUI is provided, as shown in **Figure 9**.
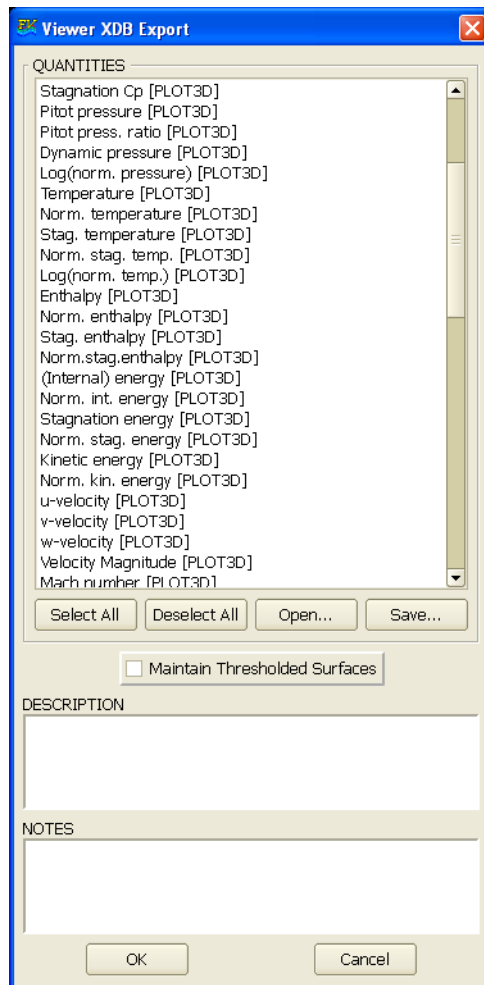
Figure 9 Viewer XDB Export Panel

The QUANTITIES section of the Viewer XDB Export panel in Viewer XDB Write and Viewer XDB Build modes allows interactive function selection for XDB export. It lists all scalar and vector functions and formulas valid for the current dataset, excluding geometric functions ("X", "Y" and "Z", "Rcyl: (X^2+Y^2)^.5", ...through "atan(Y/Z)", as well as "I", "J", and "K" (for structured data.) Scalar functions are listed first, followed by vector functions. Scalar and vector functions are further broken up by categories. NOTE: The geometric functions are automatically available in the export, and quantities I, J, or K (for

Structured data) can be exported if by creating formulas, for example, *myI = "I"*, *myJ = "J"*, *myK = "K"*.

To simplify function selection when exporting an XDB file, functions in the QUANTITIES section of this GUI are pre-selected according to the following rules:

• They have been used for a visualization object whose visibility is ON. In other words, if a function is in active use on any surface or rake for the current dataset, it will be pre-selected for XDB export.
• The function is included in the `xdb_vars` file.

For certain visualization objects, functions in use are not pre-selected:

• Scalar functions in use by 2D Plots are not pre-selected.
• Vector functions that are only used for streamline or vortex core calculations are not pre-selected.

There are four buttons located immediately beneath the QUANTITIES section of this GUI. The Select All/Deselect All buttons allow you to change your current selected functions, selecting either all or none, respectively.   If you choose the Select All button, you will see the following XDB Write Warning message. In general, the intent for XDB exports is that fewer functions are saved, containing only the functions of immediate interest.

Figure 10 XDB Write Warning

Pressing the Open... button lets you browse to and open a previously created `.xfn` file. When this is done, any functions in the QUANTITIES list matching those in the `.xfn` file are pre-selected. This action will de-select anything that was already selected.

Pressing the Save... button opens a browser and lets you save a `.xfn` file, independently from the XDB export. This saved `.xfn` file can be re-used in a later session.

Functions can be selected interactively, overriding the pre-selection rules. Selecting a function by left-clicking on it will select only that function and clear any pre-existing selec-

tions. To select more than one function, hold the [Ctrl] key down and left-click on the desired functions. To select a range, left-click on the function at the top of the range, hold the [Shift] key and left-click on the function at the bottom of the range. You can also select a range of functions by left-clicking and dragging.

The Viewer XDB Export panel (**Figure 9**) has a toggle button to "Maintain Thresholded Surfaces" in Viewer XDB Write and Viewer XDB Build modes. When enabled, only the surface geometry visible after threshold clipping is output, potentially reducing file size substantially. This option can make extracts even smaller, allowing faster transfer and read operations for these files. It also reduces the memory required for loading them into **FieldView** or **XDBview**, pushing further the benefits of XDBs.

Thresholding includes:
• clipping by the selected threshold function, and
• min/max coordinate axis limiting (coordinate surfaces only).

Thresholded surfaces are exported with the tag `[THR]` embedded in the boundary surface name and include the following types:
• Coordinate surfaces
• Iso-surfaces
• Boundary surfaces (unstructured datasets only)

Thresholding by the selected threshold function will be disregarded for computational surfaces or boundary surfaces of structured datasets when exported. However, they will continue to always clip by IJK coordinate axis limiting. If either surface types are present, visible and thresholded, complete unthresholded surfaces are exported and the following warning is issued:

XDB Write Warning: Computational or Boundary Surfaces
of structured datasets have been detected and will not
maintain thresholding on export.

Dynamic clipping has no effect on the exported XDB files.

To export the XDB file, press the OK button at the bottom left corner of the panel. When you export an XDB file, a `.xfn` file with the same name as the XDB file is written. If a `.xfn` file with the same name already exists, you will see the following overwrite confirmation:

Figure 11 XDB Function List Overwrite Confirmation

In this way, you can choose to preserve or overwrite your existing `.xfn` file. After a selection has been made here, the XDB is exported.

If you choose not to export the XDB, you can push the Cancel button at the lower right corner of the Viewer XDB Export panel.

Important notes and limitations
Surface normals are always exported for all surfaces and can be used in subsequent calculations. The surface normal function, named "N", and its components (N.vector.x, N.vector.y, N.vector.z) are reserved names.

The limit on the functions exported to an XDB is 100. If you attempt to export more than 100 functions, you will be presented with a console warning advising you that you have exceeded this limit. At this point, an XDB file will be exported containing only the first 100 of all the functions selected.

XDB exports saved via SCRIPT commands will use either the `xdb_vars` or the `.xfn` method to specify the functions to export.

XDB Format

It is possible to write face-based results to XDB files for unstructured datasets only. This further enables quantitative postprocessing via XDB workflows - integrations performed on surfaces with face-based results in XDB files will match those reported by the finite-volume based CFD solver codes.

For transient cases, XDBs are saved using one file per time step, offering a 5X to 7X improvement to read a transient XDB export. Exported filenames contain the time step number embedded in the base name, as specified by the user, according to the following rules:

- The `.xdb` extension (case insensitive) will be automatically appended to all XDB exports. If you have already specified the `.xdb` extension in the base name, it will not be redundantly added.

- The time step number matches the integer for the TIME STEP as shown in the Transient Data Controls panel for the current dataset.
- The time step number will be inserted in front of the first `"."` in the non-directory part of the specified filename. If the base name is specified as `/opt/data/my.transient.export.xdb`, then the name for the XDB for time step "10" will be `/opt/data/my10.transient.export.xdb`.
- If the last character before the `"."` is a digit, we will insert a `"_"` in between this digit and the time step number. For example, if your specified XDB filename is `F18.xdb`, the name of the XDB for time step "12" will be `F18_12.xdb`.

When you specify an XDB filename for a transient case, either interactively or from a SCRIPT containing the `XDB_ENABLE xdb_file_name` and `SWEEP TIME` commands, you will see the following console printout:

```
XDB Extract data will be exported to xdb_file_name

XDB Extract filename is modified to include the time step number.
```

A complete restart is always written when an XDB is exported to allow the easy re-creation of the original set of visualization objects stored in the XDB. For transient XDBs, this complete restart will only be written once using the base name specified for the XDB and will not have the time step number included.

Transient XDBs include the solver solution time, enabling time synchronization between XDB files and particle path data. The PARTICLE SET format is used for transient streakline data to take advantage of the substantial performance benefits for this feature. Due to these format implementations, **FieldView** releases prior to 14 are unable to read the newer XDB files.

If particle path data is imported into **FieldView** along with steady or transient volume data, it will not be redundantly stored within XDB exports.

It is still possible to create XDB files using the older XDB file format, which is compatible with the **FieldView** Advanced Technology Viewer. To do this, you must set this environment variable: `FV_USE_FV13_XDB_FORMAT`

Encryption

For cases where XDB are used in a high performance extract based workflow and not shared with the outside, an encryption step can now be skipped at export by running FieldView with the environment variable `FV_PROTO_QUICK_XDB` set to any value. This will bring a speed up of about 20% for write and read operations. Having the environment variable set at read time is not needed to read unencrypted XDB files.

Known limitation: unencrypted XDB files are only compatible with FieldView 20 or higher. Attempting to read such an XDB file with an earlier version of FieldView or with *XDB-view2* will print the console message:

```
XDB file version is higher than expected (not supported)
```

Chapter 2

FieldView Interface

2

A unified GUI is used for all platforms.

Transform Controls Toolbar

Main Toolbar

Viewer Toolbar

Right mouse click on the Main Toolbar or any toolbar to control toolbar visibility

Side Toolbar

Visualization Panels

This space is the Graphics Window in which all visualization (other than 2D plotting) takes place. It can be resized as small as the limits imposed by the toolbars. Larger windows will yield larger saved images.

Figure 12 FieldView Main Window

The **FieldView** interface (**Figure 12**) consists of a main Graphical User Interface (GUI). This interface allows you to quickly and easily access visualization tools and properties. The GUI consists of typical Windows-like pull-down menus as well as several toolbars. The toolbars contain shortcut buttons which either perform operations themselves or open interface panels. Highlights of this interface include:

Dockable Toolbars for customized GUI layout
Redundant access to common functionality and panels
Automatically saved preferences for the main window size & location of all panels
Interface starts up initially at full screen resolution

The **FieldView** interface is designed for a minimum screen resolution of 1280 by 1024 pixels. However, some laptop and projector displays have a shorter vertical resolution. Also, it is possible on some Windows systems to decrease the available vertical resolution through the use of a custom DPI setting to make system fonts easier to read. For these reasons, **FieldView** makes the tallest panels "scrollable" if it determines, at start-up, that they are too tall for the available screen space. The minimum vertical resolution for scrollable panels is 600 pixels.

Saved Preferences for the FieldView Interface

Saving the size and location of the main window, along with the location of the toolbars (and whether they are docked or not) is part of the broader functionality of saving preferences. This information is stored in `FieldView.ini`. Its location depends on the system where **FieldView** is being run.

On Linux/macOS,
`/home/user/.config/Tecplot/FieldView.ini`

On WIndows,
`C:\Users\user\AppData\Roaming\Tecplot\FieldView.ini`

Note that the position of the file browsers, warning or error messages and confirmation pop-ups are not saved. Layouts are not saved as part of your **FieldView** preferences on exit. As a result, **FieldView** will start with a single window unless a Complete Restart with a multi-window layout is read at start-up.

FieldView Preference Override

In some cases, it may be desirable to turn off the default behavior which saves and restores user preferences, such as the GUI layout and last file locations browsed. This is controlled by setting the following environment variable: `FV_NO_PREFERENCES`

When this environment variable is set, the **FieldView** preference file (`FieldView.ini`) will neither be read on start up of nor written on exit from **FieldView**.

Multi-Window Operation

The **FieldView** graphics window can be split into more than one window. The basic GUI features for multi-windows are shown in **Figure 13**. Each window has the full functionality of interaction and transform control available within the full graphics window. Interactive mouse control, locked transforms, the click viewer, World, Dataset, Region, Surface, Plot and Light object transforms as well as Detach/Reset all work for the current window whether you have a single window or multi-window arrangement. The following attributes can be uniquely set for each window:

> Background Color
> Background Image and position
> Outline set either ON or OFF
> Axis Markers, set either ON or OFF
> Rotation Increment
> Scale Factor
> Move Factor
> Perspective, set either ON or OFF
> Perspective Angle, when Perspective is ON
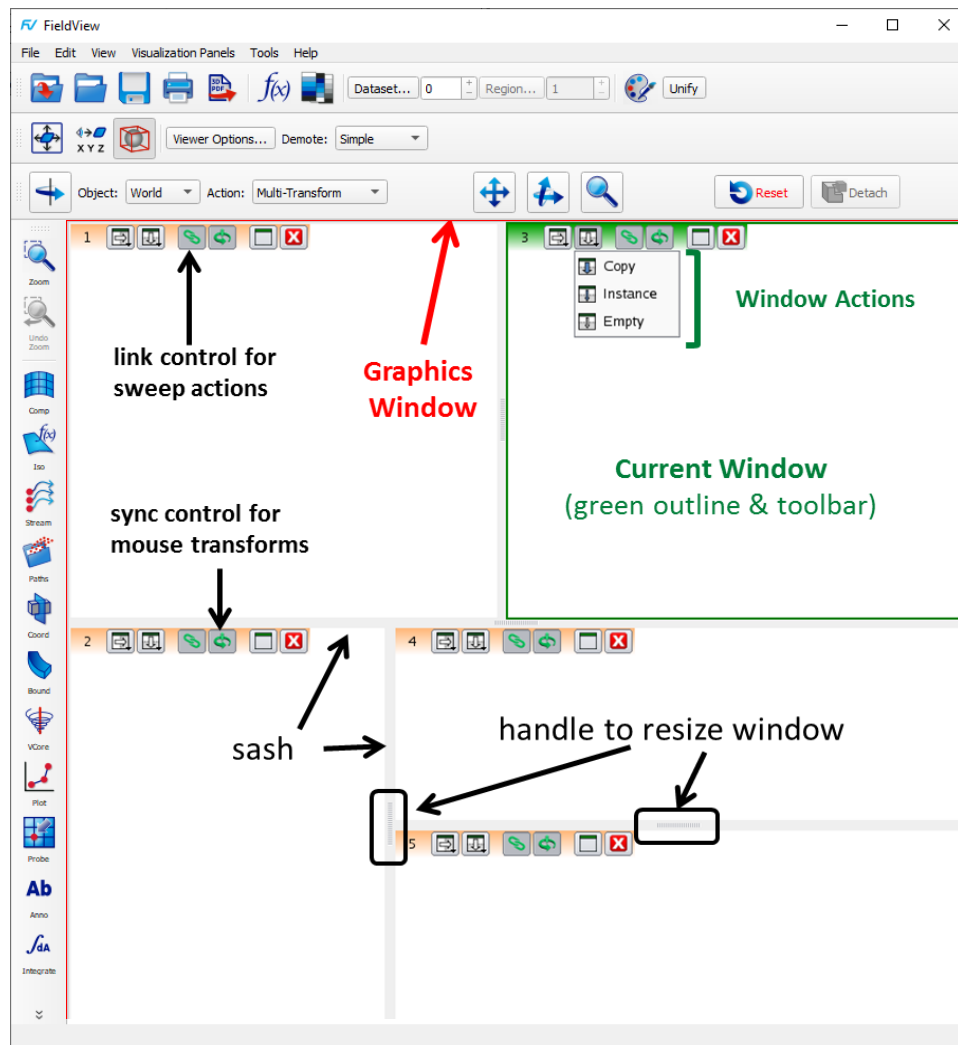
Figure 13 Getting to know multi-window operation

There is a toolbar within each window, fixed in the upper left corner. A Window Actions pulldown menu, accessible from either the vertical or horizontal split icons on the window toolbar, lets you create a new window by splitting the current one. Windows can be split repeatedly, either horizontally or vertically. When a window is split, the new window resulting from that action becomes the current window.

The toolbar for the current window is colored green and is highlighted with a green border. To make a different window current, left-click anywhere in the window of interest.

Sashes are used to identify the borders separating each window. A handle located at the center of each sash can be used to interactively resize windows. Windows within the graphics window are interconnected. If a window is resized, any windows bordering it will also be resized, growing or shrinking to fully occupy the space defined by the graphics window.

Window Split Actions

When you start **FieldView**, the green toolbar in the upper left corner of your graphics window appears as illustrated in **Figure 14**. Clicking on either the vertical or horizontal split buttons gives you options for splitting the current window. Three distinct actions can be performed when a window is split: **Copy**, **Instance** or **Empty**. **Copy** and **Instance** perform specific operations with the visualization objects (surfaces and rakes) present in the window prior to splitting it.
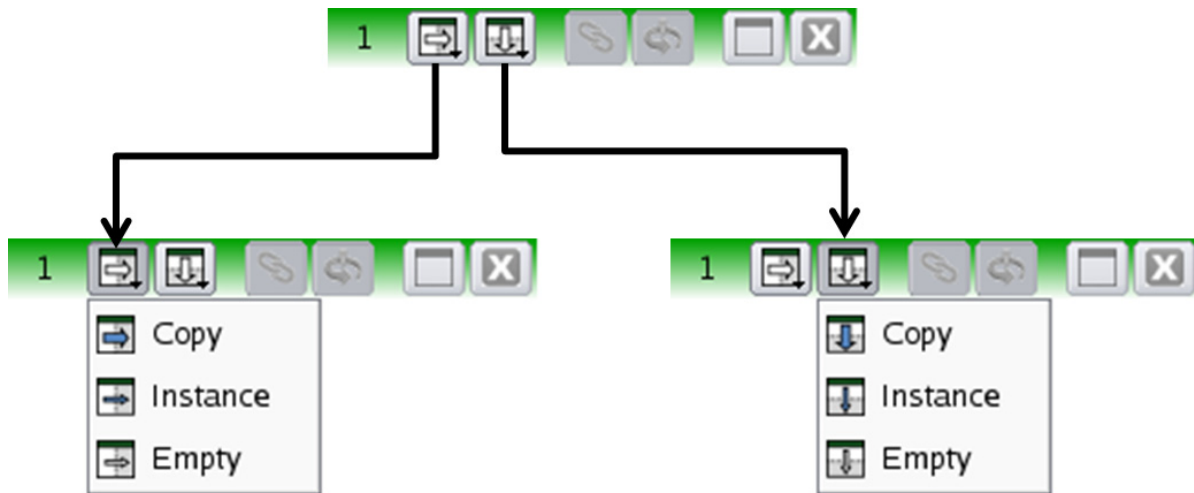
Figure 14 The Window Toolbar and Action menus

Window Action: Copy

When **Copy** is selected from the Window Actions pulldown menu, a virtual representation for all of the datasets in current window is created. As part of the **Copy** action, the visualization objects for each dataset in the original window are re-created on virtual datasets in the new window produced by the split. These visualization objects are independent representations and can be displayed with different attributes than their original content. In **Figure 15**, we start (at left) with a coordinate plane colored with the scalar function Temperature, and a boundary surface with the display type Mesh. After a Vertical Split, **Copy** action, a new coordinate surface and a new boundary surface are created in window 2 at right. At this point, the scalar function for the coordinate surface in window 2 is changed from Temperature to Concentration.
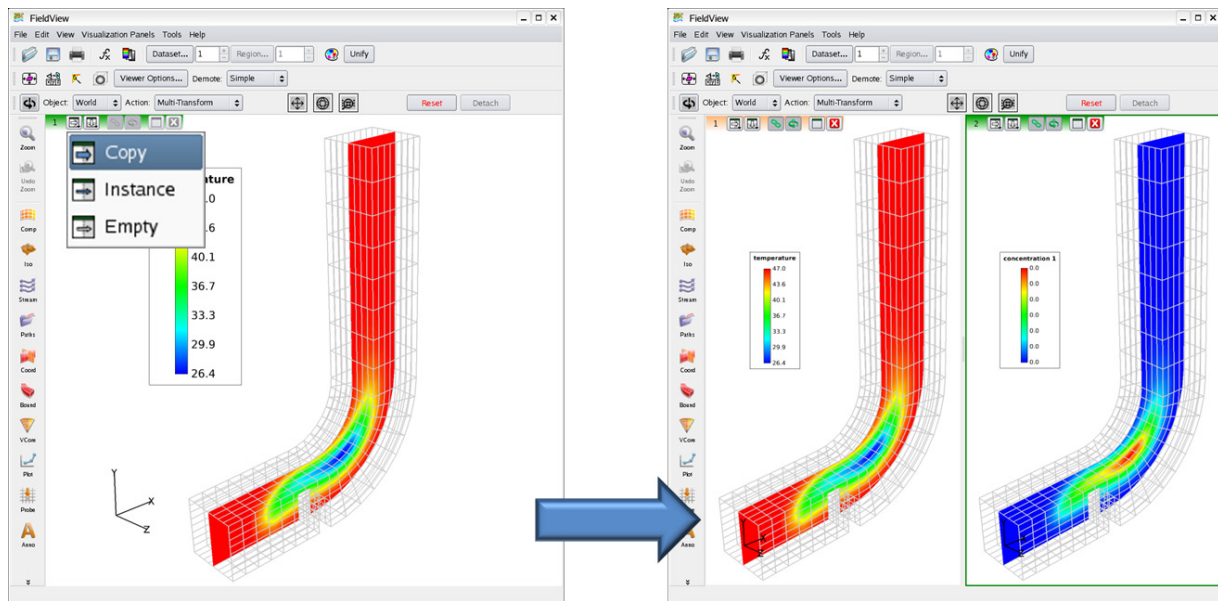
Figure 15 Vertical Split, Copy action

Following a **Copy** action, it is possible to create new visualization objects in one window and not the other. It is also possible to save a Current Dataset Restart for the data in one window, make another window Current, and Open the Current Dataset Restart, creating the surfaces and rakes defined from the restart on the current dataset in the current window.

Window Action: Instance

When the **Instance** action is selected, a view-only representation of the visualization object(s) for all datasets in the original window is created in the new window Selecting any instanced visualization object by quick-picking in any window lets you modify it. Modifications, such as changing the scalar function or scalar range for example, are automatically propagated to all instances of that visualization object.

In **Figure 16**, we start (at left) with the same surfaces that were used in the **Copy** example above. After a Vertical Split, **Instance** action, the visualization objects are displayed in both windows. The view direction can be modified independently for each window, letting you view these visualization objects from different angles.

If you create a new surface or rake on any dataset in either window, an instance of that object is displayed in all windows which were created using the **Instance** action.
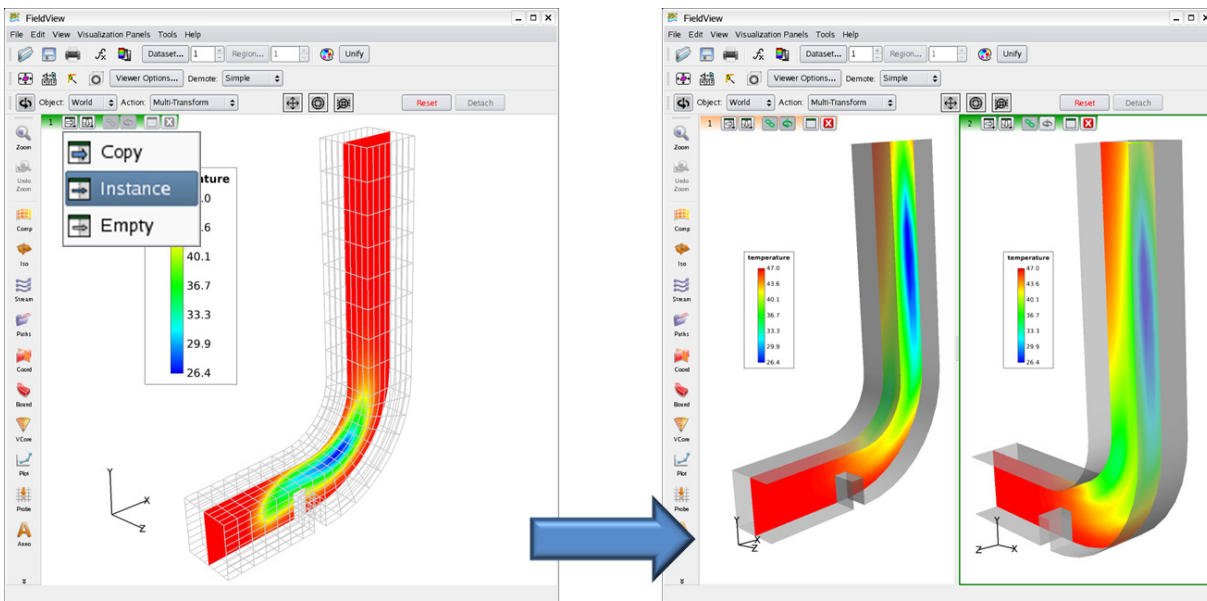
Figure 16 Vertical Split, Instance action

Window Action: Empty

When the **Empty** action is selected, a new window is created without any virtual copies
of datasets or visualization objects, as shown in **Figure 17**. If the empty window remains
current, any dataset when read in will be displayed in it. The contents of all other win-
dows will remain undisturbed.
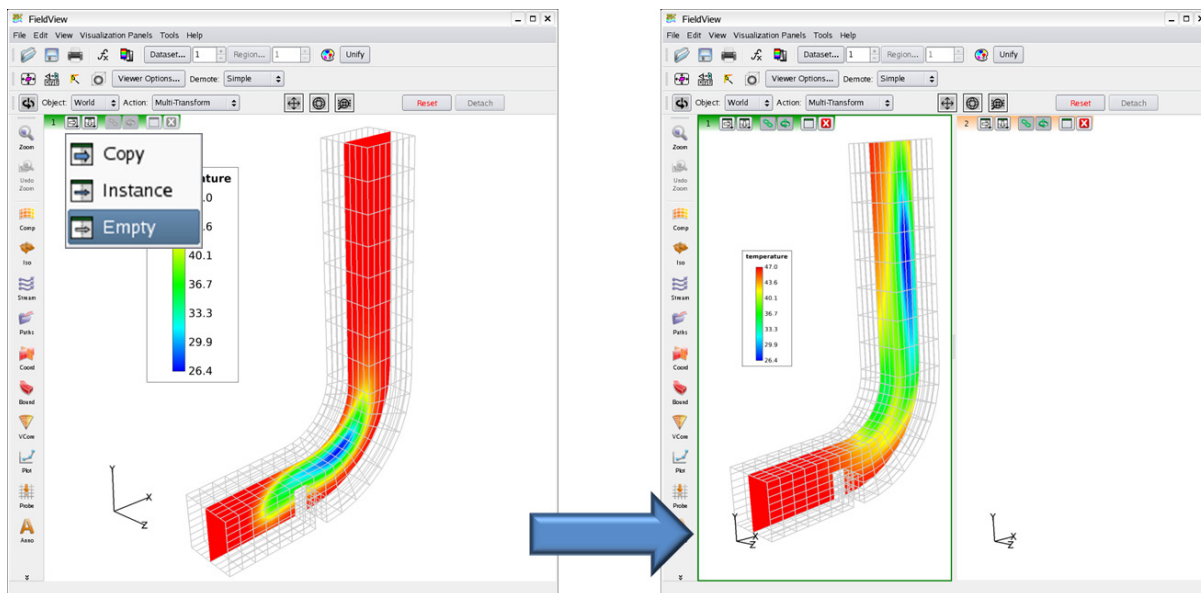
Figure 17 Vertical Split, Empty action

Maximizing, Restoring and Closing a Window

When multiple windows are present, it is sometimes useful to maximize one window temporarily. This can be done by pressing the ☐ on the current window toolbar. When the current window is maximized, the button label changes to ⬒. Pressing ⬒ restores the window to its original size and location.

To close the current window, press **X** on the window toolbar. Note that this icon does not control the visibility of the toolbar in the window. When a window is closed, **all** of the visualization objects (surfaces, rakes and annotations) in the closed window will be deleted. There is no undo operation to recover visualization objects from a closed window.

Showing/Hiding the Window Toolbar

The window toolbar is colored in green in the current window (see **Figure 14**) and beige in the non-current window(s). Its visibility can be controlled uniformly for all windows or individually for the current window. By default, the window toolbar is visible in all windows.
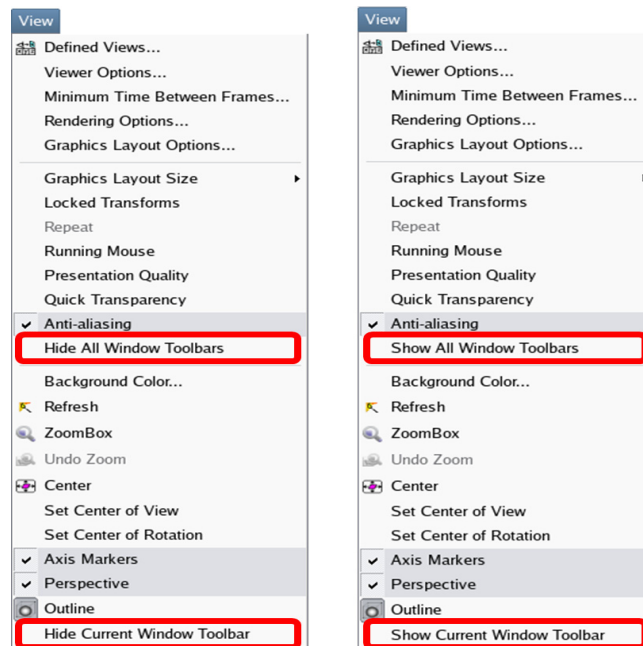
Figure 18 View pulldown menu

View pulldown menu commands to turn off the visibility of all window toolbars or just the current window toolbar, respectively (see **Figure 18,** left), are:

View ➔ Hide All Window Toolbars

View ➔ Hide Current Window Toolbar

View pulldown menu commands to turn on the visibility of all window toolbars or just the current window toolbar, respectively (see **Figure 18,** right), are:

View → Show All Window Toolbars

View → Show Current Window Toolbar

Multi-Window Examples
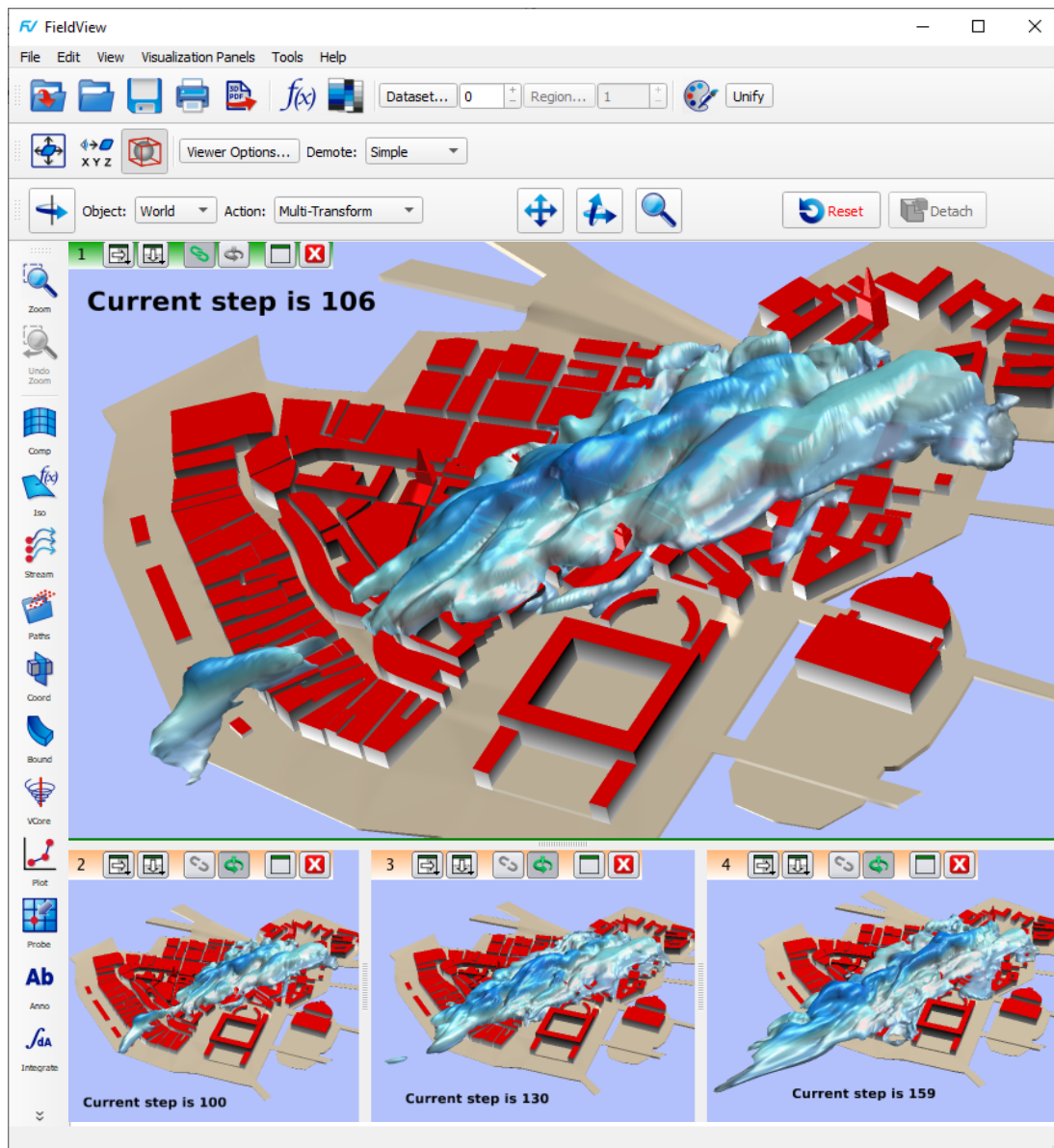
Figure 19 Multiple views of a transient urban dispersion simulation

In **Figure 19** above, visualization objects in the top window are created interactively, showing a dispersion cloud above a city. Creation of a new window as well as copying these visualization objects happens with a single mouse click. Transient sweeping can be applied to just one or all of the windows simultaneously to study how the dispersed cloud changes in time.
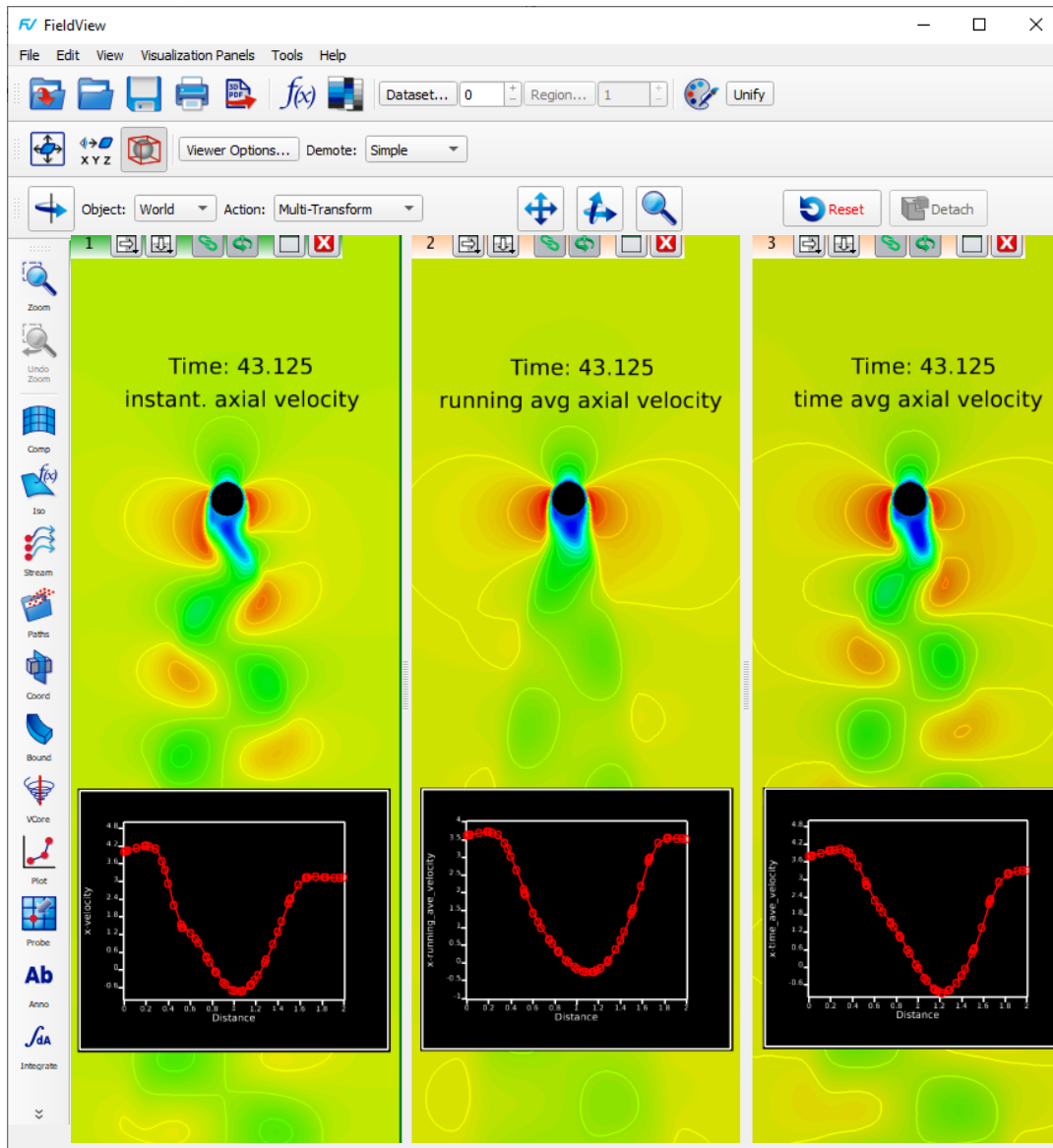
Figure 20 Vortex Shedding in multiple windows

There is no practical restriction on the number of windows which can be created. Also, the content of a window is not restricted in any way. In **Figure 20**, three windows display a coordinate surface and an embedded plot. A different scalar is illustrated in each of the

three windows. You can sweep through time for all three windows simultaneously, updating all coordinate planes and embedded plots at the same time.

Working with Toolbars

Toolbars are used to group icons for quick access to the main features of **FieldView**. The Main Toolbar, illustrated in **Figure 21** below, provides direct access for saving and reading Complete Restarts, direct printing, and access to the Function Specification Panel, the Scalar Colormap Specification Panel, the Dataset and Region Controls Panels, the Color Mixer and the Unify button.

.
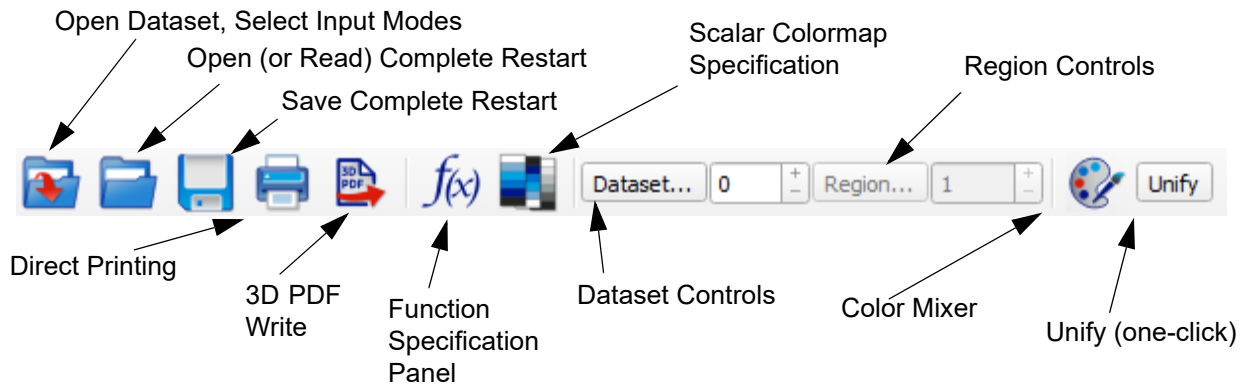
Figure 21 Main Toolbar

Opening Datasets - Users can open Datasets using the file icon on the Main Toolbar, or by using File > Data Input. This also allows for selection of local or Parallel input options.

Saving or Opening Restarts - Users can save or open Restarts using the file icons on the Main Toolbar, or by using File > Save/Open. Note that this will default to Complete Restarts, but also allows selection of component restarts.

Direct Printing to a configured printer - Users can use send graphics to a Postscript printer using the icon on the Main Toolbar, or by using File > Print.

3D PDF write - Users can use the icon on the Main Toolbar, or by using Tools > 3D PDF write.

Function Specification - Users can choose, edit or create functions using the icon on the Main Toolbar, or by using Edit > Function Specification.

Colormap Specification - Users can choose or modify Scalar Colormaps, object colors or Environments using the icon on the Main Toolbar, or by using Edit > Scalar Colormap Specification.

Dataset and Region Controls - Users can specify duplication or transforms by using the icons on the Main Toolbar, or by using Tools > Dataset or Region.

Color Mixer - Users can create customer colors by using icon on the Main Toolbar, or by using Tools > Color Mixer.

Unify - Users can Unify objects using the button on the Main Toolbar, or by using Tools > Unify.

The Transform Controls Toolbar, illustrated in **Figure 22** below, provides icons and buttons to Center the Object, Refresh the graphics window, Toggle Outline on or off, bring up the Defined Views and Viewer Options panels, and control Demotion. Except for Demotion, the Transform Controls toolbar functionality is redundantly accessible through View menu on the Main Toolbar.

Center (one-click)   Toggle Outline ON/OFF   Demotion Control

Viewer Options...   Demote: Unchanged

Defined Views
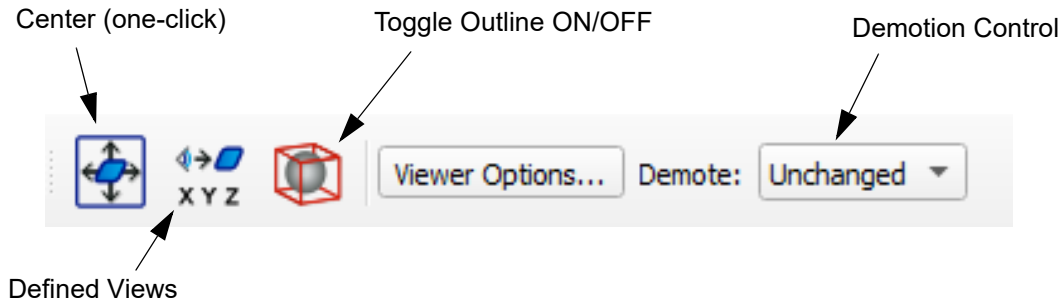
Figure 22 Transform Controls Toolbar

The Viewer Toolbar, illustrated in **Figure 26** allows you to use the mouse to translate and transform various items (World, Dataset, Region, Surface, etc.) that make up your visualization. There are several different modes and many of the modes have different icons and actions. The various modes are activated based on the *current* Object and settings on View menu (see **Figure 60**) and Viewer Options panel.

Interacting with the Mouse

M2 (Middle Mouse)

Can also press Alt-M1 (Linux/Windows) or option-M1 (macOS).
On a 2-button mouse, can press both buttons at the same time.

M3 (Right Mouse)

Can also press Shift-Alt-M1 (Linux/Windows)
or control-M1 (macOS).

M1 (Left Mouse)
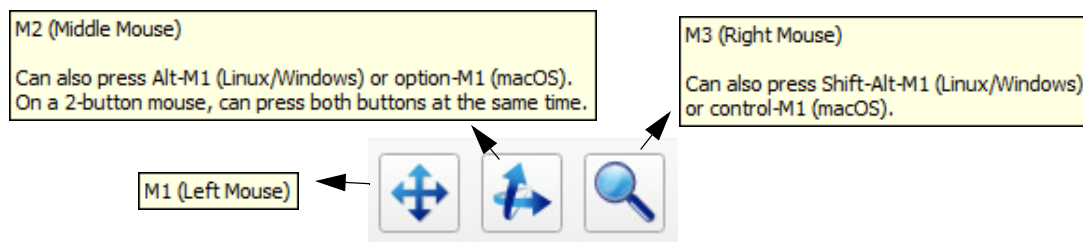
The Mouse Icon area shows depictions of the actions of the mouse buttons appropriate to the Action selection and the current Object. Pressing a mouse button will highlight an icon. Releasing the mouse button will un-highlight the icon.   **M1** is the left mouse button, **M2** is the middle mouse button and **M3** is the right mouse button. Note: You do not have to click on these icons. They are presented on the screen with mouse-over "hints" (shown above) for information only.

Figure 23 Mouse Icons

Note: If the current Object is Surface, and Surface transforms are invalid, then the mouse icons will be blank and no transformation can take place. Surface transforms are invalid, for example, if the surface belongs to a transformed region. If the current Object is Region, and Region transforms are invalid, then the same will occur.

Using FieldView Without a 3-Button Mouse

Interacting with FieldView differs depending on whether you have three, two or one-button mouse. With a three-button mouse, the mapping of the icons to the mouse is simply left to right. FieldView allows Users to interact *without* a 3-button mouse or with a laptop Touchpad, by providing the following mouse-button emulations, via modifier keys:

[M2] - Use the Left+Right Mouse (together), or Alt-M1 to perform the transformation shown on the middle icon.

[M1] - Use the Left Mouse to perform the transformation shown on the left icon.

[M3] - Use the Right Mouse, or Shift-Alt-M1 to perform the right icon transformation.
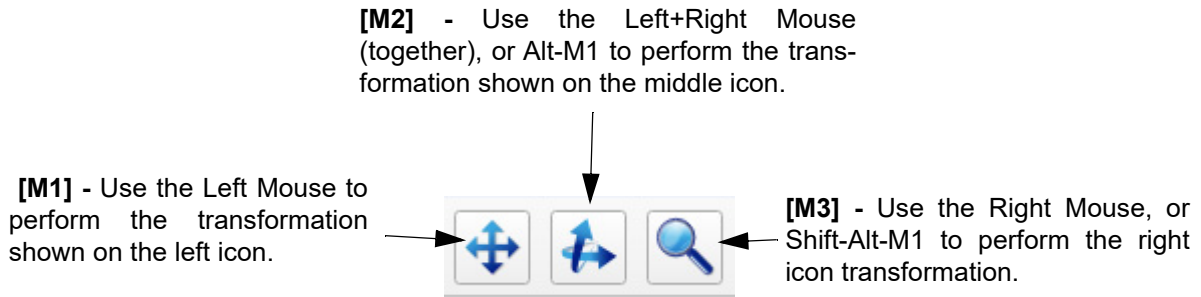
Figure 24 Alternate Mouse Button Mappings

On laptop Touchpads (also known as Trackpads) all three mouse interactions can be made by using modifier keys as shown below. FieldView provides hints throughout it's interface for mouse actions.

Linux and Windows

Press **Alt** together with pressing touchpad for **M2**.

Press **Shift** and **Alt** together with pressing touchpad for **M3**.

Apple

Press **option** together with pressing touchpad for **M2**.

Press **control** together with pressing touchpad for **M3.**

Figure 25 Mouse Emulation for Touchpads

Mouse Controls

The mouse controls available on the Viewer Toolbar (see **Figure 26**) allow you to easily manipulate the view of your data. In addition, pressing the spacebar while the **Graphics Window** has focus will 'toggle' the icons to those shown in **Figure 28** (and back again) allowing you even more functionality. These three icons show the functionality of the mouse buttons.

Interactive Mouse Control

Once an Object and Action are selected, you must move the cursor into the graphics window and hold down the appropriate mouse key to initiate the Transform action. Only one mouse button can be active at a time. The Mouse Icon area shows the appropriate pictograms depending upon the Action selection and the current Object. Pressing a mouse button will highlight the active selection. Releasing the mouse button will unhighlight the current action.

Note: If the current Object is Surface, and Surface transforms are invalid, then the mouse icons will be blank and no transformation can take place. Surface transforms are invalid, for example, if the surface belongs to a transformed region. If the current Object is Region, and Region transforms are invalid, then the same will occur.

This menu controls the **Object** that is affected by the view transformations: World, Dataset, Region, Surface, Plot or Light.

Press **[M2]** for **XYZ Rotation**. For a two button mouse, press both the Left and Right Mouse buttons together.

The **Detach** button separates the current Surface (if Object: Surface) or Region (if Object: Region) from the transform hierarchy. See explanation below.

Press **[M1]** for combined screen **XY Translation** (X translation is horizontal movement, Y translation is vertical movement).

Pess **[M3]** for: **Magnification** when Object is World and **Scaling** when Object is Dataset.

Reset resets all interactive transforms which have been performed on the current Object. Transforms set on the Region or Dataset Panel are not reset.

Toggle between Locked Transform controls, and Multi-Transform controls.

Figure 26 Viewer Toolbar

Detach

The Detach button can be used to isolate the *current* Region or Surface from the transform hierarchy. This means that once a region/surface is detached, it is *no longer affected* by subsequent World, Region or Dataset transformations. Regions are not affected by World, Dataset or Surface transforms and Surfaces are not affected by World, Dataset or Region transforms. Neither Region nor Surface are affected by a Center command.

Important Note: You must detach the Region/Surface first, then transform it with a Region/Surface transform. The act of detaching the surface drops all interactive transforms. That is, if you have any Dataset transforms (done interactively or on the Dataset Controls panel) or Region transforms (done interactively or on the Region Controls panel), these will be undone. This may cause the Region/Surface to transform unexpectedly.

Model Relative Transforms

In order to access the model relative transforms, the *current* Object must be Dataset and the Viewer Options panel must have the Model Relative button *on*, not the Screen Relative button. When the icons shown in **Figure 27** are displayed, all rotations will take place around the *dataset*, or *model* axes.

Toggle between Locked Transform controls, and Multi-Transform controls.

Press [**M2**] to rotate the current Object around the model X axis. For a two button mouse, press both the **Left** and **Right Mouse** buttons together.

Press [**M1**] to rotate the current Object around the model Y axis.

Press [**M3**] to rotate the current Object around the model Z axis.
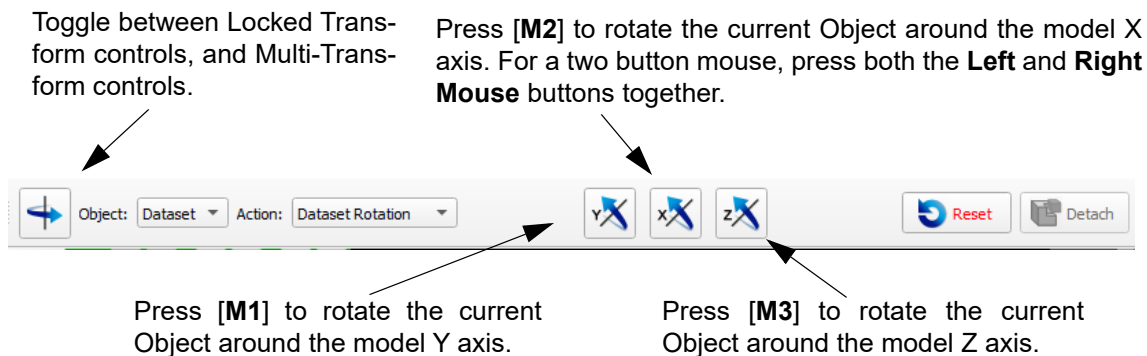
Figure 27 Viewer Toolbar - Model Relative

Spacebar Toggle

Under Interactive Mouse control, an additional set of transform controls can be accessed by pressing the spacebar on your keyboard after clicking in the graphics window. This will change the icons and their behavior to those shown in **Figure 28** (except if Object is Light, in which case **[M2]** changes from XYZ rotation to Screen-Z rotation). Pressing the spacebar again will return the icons to their previous state.

Note: The toggle mode only functions if the **Graphics Window** has the mouse focus.

Toggle between Locked Transform controls, and Multi-Transform controls.

Press [**M2**] for Screen-Z rotation.

Click [**M1**] and drag to create a Zoom Box (like the control on the Side Toolbar).

Press [**M3**] for Z translation. This only works when Perspective is turned on.
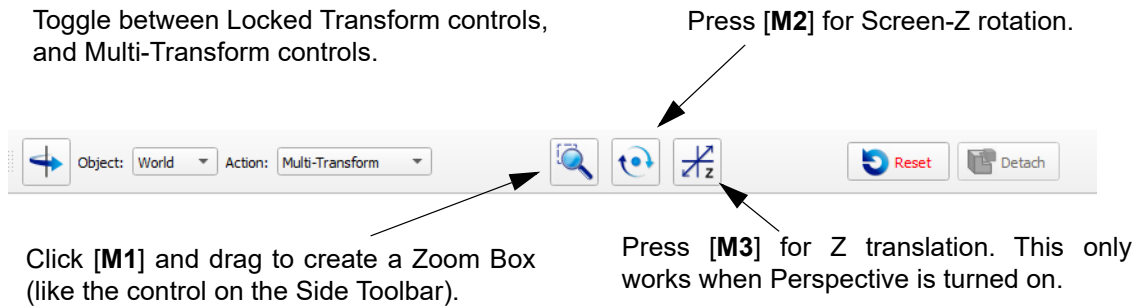
Figure 28 Viewer Toolbar - spacebar toggle

Running Mouse

If Running Mouse (from the View pull-down menu or redundantly, from the Viewer Options panel) is turned *on*, then the actions of the mouse change. The effects differ depending on the current state of the Viewer Toolbar. This mode will cause the current action to be performed *until released*. That is, if the current mouse action is Zoom **[M3]**, then Running Mouse causes the view to zoom as long as the right mouse button is held. Releasing the mouse stops the motion. Normally, *without* Running Mouse turned on, the object in the **Graphics Window** *only* moves if the mouse is moving. The Running Mouse mode causes the mouse to *set objects into motion*.

If the current transform mode is Locked Transforms (see section below), Running Mode causes right mouse movement to perform the selected locked transform *until released*. Normally, these actions only take place while the mouse is moving (i.e., stopping the mouse stops the action).

Locked Transforms

Locked transforms are enabled by clicking on the left-most button on the Viewer Toolbar. With locked transforms enabled, there are two different modes of interaction with the right mouse button. By default, this toolbar comes up with the right mouse icon active.

The Right Mouse button comes up active by default. The remaining icons, when selected, illustrated what action the right mouse button will have in the graphics window.

When you click the Right Mouse button, the remaining icons, when pressed, will rotate, transform or zoom the object in the graphics window by a specified amount.

Figure 29 Locked Transforms and the Right Mouse

In this mode the *right* mouse **[M3]** button can be used to move the *current* Object in an interactive mode. This is different than the default interactive mode. Mouse behavior is controlled by the state of Running Mouse as described previously. When the Viewer Toolbar is in the locked transform mode, you can only transform in one axis at a time. That is, you can translate up and down by holding down the right mouse button in the graphics window and moving it, but to translate left or right, a different pair of translation buttons needs to be activated. The axes around which the transforms take place are in Screen space (the default). If the current Object is Dataset and the Model Relative option is selected on the Viewer Options panel, then the axes are the Dataset (or Model) axes (**Figure 31**).

Activate this button to turn on the right mouse control of the current Object.

This button pair rotates the current Object around the screen Z axis (which is positive to the right).

This button pair trans-lates the current Object left or right.

This button pair rotates the current Object around the screen X axis (which is positive to the right).

This button pair zooms the current Object in and out.

This button pair translates the current Object up or down.

This button pair rotates the current Object around the screen Y axis (which is positive to the right).

Figure 30 Viewer Tool Bar - Locked Transforms

This button pair rotates the current Object around the model Y axis.

This button pair zooms the current Object in and out.

This button pair translates the current Object up and down.

This button pair rotates the current Object around the model X axis.

This button pair rotates the current Object around the model Z axis.

This button pair translates the current Object left and right.

Figure 31 Viewer Toolbar - Locked Dataset Model Relative

Click Viewer Control with the Right Mouse **[M3]** Button

When the right mouse icon on the locked transforms toolbar is pressed, the remaining icons on the toolbar, when clicked, translate, rotate or zoom your Object by a pre-specified amount according to the settings on the Viewer Options panel. Please refer to the section titled **View > Viewer Options** for additional details. This mode of operation is referred to as the Click Viewer mode. With this mode set, you cannot move your current Object using any of the mouse buttons.

View >Viewer Options

Settings on the Viewer options panel control how much your current Object will move when you are working in the 'Click Viewer' mode.
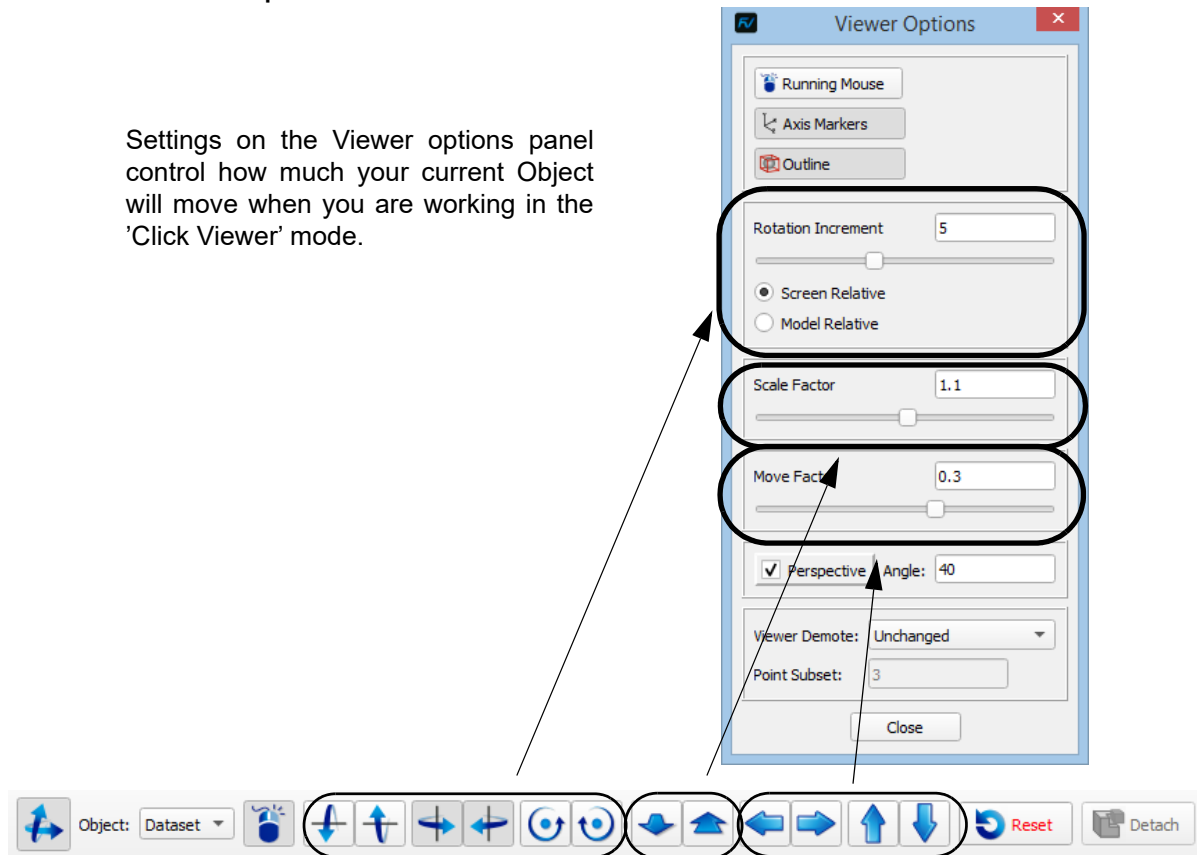
Figure 32 Locked Transforms and the Right Mouse

Viewer Options > Viewer Demote: Pull-Down
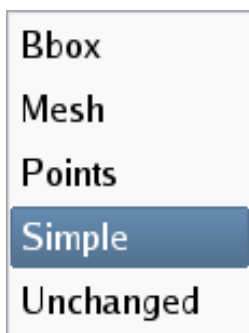
The Demote: setting determines whether objects in the graphics window are changed to simpler representations when interactive transforms are performed. The Refresh button will restore to an un-demoted view. The demotion options are: Bounding Box, Mesh, Points, Simple and Unchanged (no demotion). Note: For some display settings/hardware types, Bounding Box is the default setting.

Figure 33 Demote: Pull-Down

Due to the single-color-per-face display of face data, demotion of boundary surfaces with face data is handled differently. If Demote: is set to Points or Mesh, then the demotion color will be white (if the background is black) or black (if the background is white).

Object / Action Selection

Press and hold down to select the current Object for transformation.

Press and hold down to select an Action to be performed on the current Object.
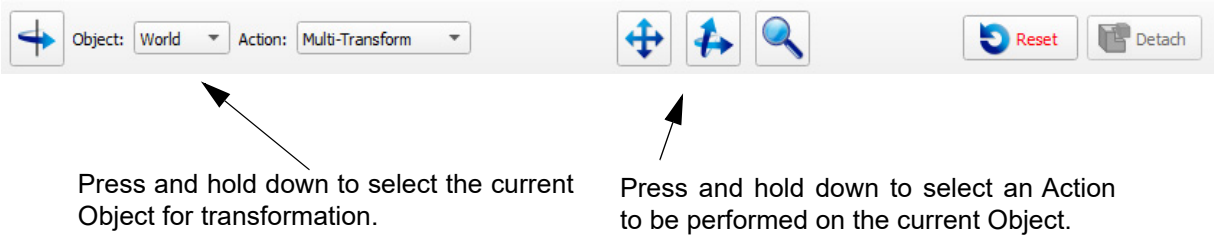
Figure 34 Object / Action Selection

Once an Object is selected, there are two mouse interpretation modes - Running and Tracking:

Running: When this button is turned on, then clicking and moving the mouse in the graphics window sets the object in motion (whether it be translation, rotation, zoom, etc.). The object will stay in motion until the mouse button is released. The speed at which the transform takes place is dependent on the distance the mouse is moved from the position of the initial click of the mouse button. If the mouse button is clicked at one location and the mouse is dragged only a short distance, then the transformation will be slow/small. If the mouse button is clicked at one location and the mouse is dragged a long distance, then the transformation will be fast/large. Note: The distance is the determining factor for the rate of transformation. The speed of the mouse drag does not affect the rate or transformation.

Tracking: This setting is the default and exists when the Track button is on. In this mode, the object moves when the mouse moves. When the mouse movement stops, the object movement stops. If the mouse is moved quickly, the object moves quickly. Note: Track mode can malfunction if you click/drag rapidly while a surface or rake is animating. Simply slowing down your interaction will yield accurate movement of graphics.

Object Selection

Select World as the Object when you wish to transform (change the view of) all surfaces and particle paths in the graphics window together in a group, relative to the screen orientation (see **Figure 35**).

Transforms the World, which contains all of the surfaces, streamlines and particle paths on the screen.

Transforms the current Dataset

Transforms the current Region

Transforms the current Surface

Transforms the current 2D Plot

Transforms the Light direction
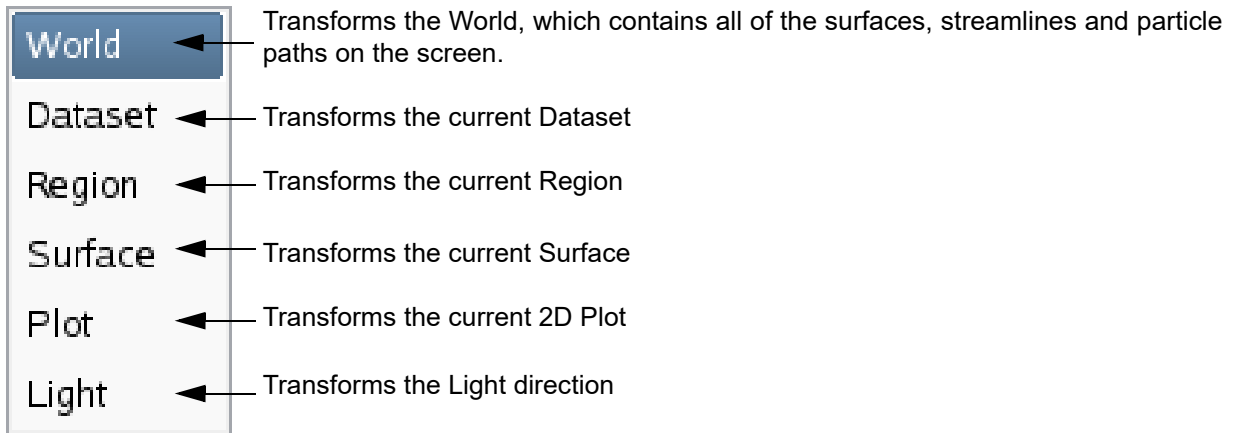
Figure 35   Object Option Menu

Synchronized World Transform Behavior

By default, world transforms are synchronized across all windows - if you perform a transform in one window, visualization objects in all the other windows will also move, as illustrated at left in **Figure 36**. The transform icon, fourth from left in the window toolbar, illustrates the state of synchronized world transform behavior for that window. It can be toggled on (icon colored in green) or off (icon colored in gray) by clicking directly on the icon.

When the synchronized world transform behavior is turned off for a window, all trans-forms at the world hierarchical level are applied only to the contents of that window when it is current. If the window is not current, world transforms performed in any other win-dow will not be applied to its contents.

Hierarchical transform actions at the Dataset, Region, Surface and Plot level are not affected by the state of synchronized world transform behavior.
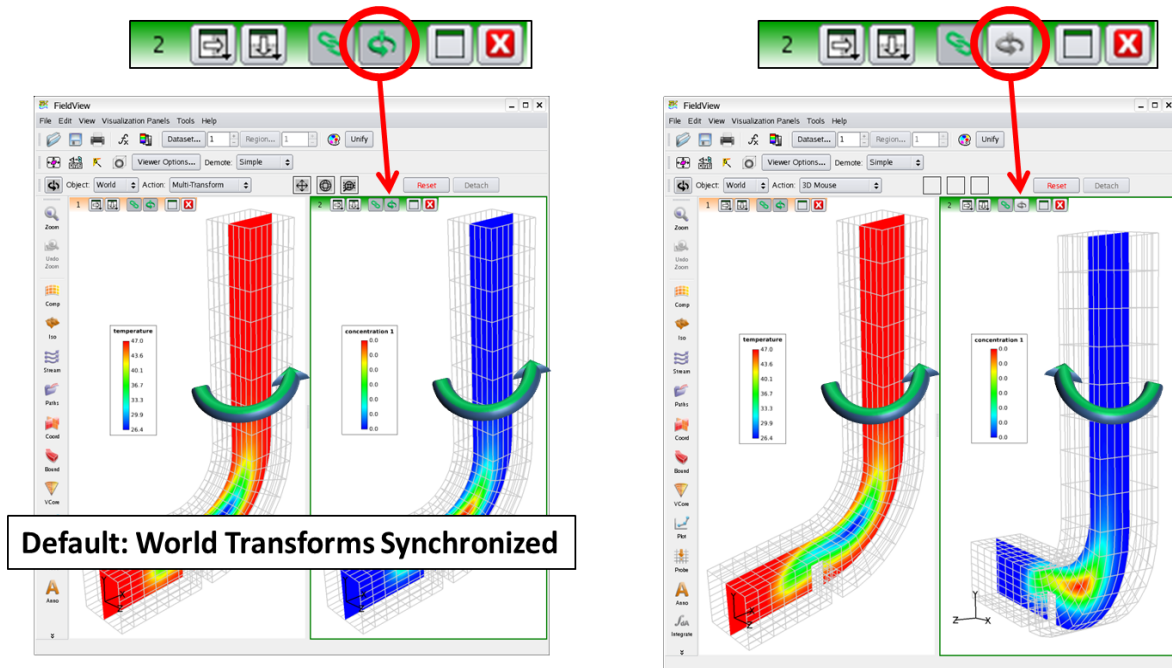
Figure 36 Synchronized World Transform Behavior

Hierarchical transform actions at the Dataset, Region, Surface and Plot levels apply to the current window under the condition that the window content is created on different datasets. If the content within a window has been created though a **Copy** action, Dataset, Region, Surface and Plot transforms are applied to the current window only. If the content within a window has been created through an **Instance** action, Dataset, Region, Surface and Plot transforms are applied to all windows with "instanced" content.

Select Dataset as the Object when you wish to transform (change the view or the orientation relative to the dataset local origin of) all surfaces and particle paths of the current Dataset.

Select Region as the Object when you wish to transform the current region. Note that all contents of the selected Region (grids, surfaces and streamlines, but not including Particle Paths) will be transformed as well. The streamline will follow the seed. If the seed is part of a transformed Region, it will transform, and take the entire streamline with it. These streamlines will still animate and be correctly computed as if the Region Transforms did not exist. Region transform information will be stored in the Dataset Restart file. Please refer to **Chapter 3** for a schematic of this hierarchy.

Select Surface as the Object when you wish to transform the current surface. This feature will allow you to move any surface away from its initial position. This includes Computational, Iso-, Boundary and Coordinate surface. It excludes Streamlines, Particle Paths, Auxiliary Seed planes, Vortex Cores and Surface Flows (Separation and Reattachment lines). To move a desired surface, make sure it is the current surface. Then

with the Object: as Surface, you can use the mouse transform controls to rotate, translate and zoom the surface. An Iso-surface, for instance, can be created and then duplicated and then its duplicate extracted away from the dataset and zoomed in for closer examination. The extracted surface can then be swept in place. Extracted surfaces will not affect the dataset outline, but the World bounding box dimensions will change if the extracted surface is moved beyond the original extents of the dataset. Rotations and Scaling (zooming) will occur about the center of the surface, not World or Dataset center. The Reset button will undo all surface transforms for the current surface. Surface transform information will be stored in the respective surface restart file.

The Detach button can be used to isolate the current Region or Surface from the transform hierarchy. This means that once a region/surface is detached, it is no longer affected by subsequent World, Region or Dataset transformations. Regions will not be affected by World, Dataset or Surface transforms and Surfaces will not be affected by World, Dataset or Region transforms. Neither Region nor Surface will be affected by a Center command. A reset of a detached Region or Surface will effectively reattach it to the transform hierarchy.

Detaching and Transforming the Region/Surface
You must detach the Region/Surface first, then transform it with a Region/Surface transform. The act of detaching the surface will drop all interactive transforms. That is, if you have any Dataset transforms (done interactively or on the Dataset Controls panel) or Region transforms (done interactively or on the Region Controls panel), these will be undone. This may cause the Region/Surface to transform unexpectedly. Non-uniform scaling and duplication are also dropped.
A detached Region or Surface will be affected by World zooms (using [M3] when the transform Object is World).

Mirroring and Transforming
In previous versions of **FieldView**, surface transform for mirrored datasets would result in surfaces that 'split' at the mirror axis and transform in opposite directions. **FieldView** now transforms the surface as one piece. In addition, surfaces with Dataset Scaling applied will now correctly translate/rotate.

Note: Translating surfaces can allow you to create unphysical and incorrect images of your geometry and solution. Adding annotations through the Titles panel is one way to avoid later confusion.

Select Plot to move the embedded 2D or Surface Plot within the graphics window. You will be able to control the size and location of the embedded plot within the graphics window.

Select Light will allow for the rotation of the light orientation using the Multi-Transform rotation or Locked Rotation Actions. **FieldView** uses a two-sided, infinite "light source" (therefore rotating the light source 180 degrees on any axis will result in no change to the lighting). While transforming the light source a visual feedback in the form of a circular marker and dashed line originating from the axes marker display will indicate the light source orientation. Note that the axis markers will have to be turned on to see the light source orientation feedback. In all transform modes, the axes displayed will indicate the dataset axes of the current dataset. The Reset button will restore the light source position to its default orientation of [1,1,1] whenever Light is the current transform object and the Reset button is pressed.

Note: Rotating the light is a screen relative action. This means that an X rotation will (in almost all cases) not appear to be relative to the X axis display marker.

Hierarchical transform actions at the Light level are always applied to the current window only. This lets you position the light source independently in one window while retaining the same view in other windows.

Selection of a non-existent Object will cause a Warning popup to be issued upon any mouse action in the graphics window. The Mouse Icon area will have no display when a Title or Legend Object is selected when no current Title or Legend exists.

Action Selection

Select this option transform using to allow for translation, rotation or scale of the selected Object.

Select this option to allow for translation, rotation or scale of the selected Object.

Select this option to allow locked (constrained) rotation of the selected object.

Select this option to allow locked (constrained) rotation of the current Dataset about the dataset's origin in its local coordinate system, about the dataset's axes.

Select this option to allow for locked (constrained) translation of any Object.

Action: Multi-Transform
- 3D Mouse
- Fly-Thru Transform
- Multi-Transform
- Locked Rotation
- Dataset Rotation
- Locked Translation

Figure 37 Action Option Menu

Action: Fly-Thru Transform

To make fly-thrus easier, the Action Fly-Thru Transform is available, as shown in **Figure 38** below.

Figure 38 Fly-Thru transform action

In this mode, the third mouse button on the standard 3-button mouse is re-mapped to permit Z translation. Note that Z translation is only possible when perspective is turned on, which can be done from View menu or the Viewer Options menu.
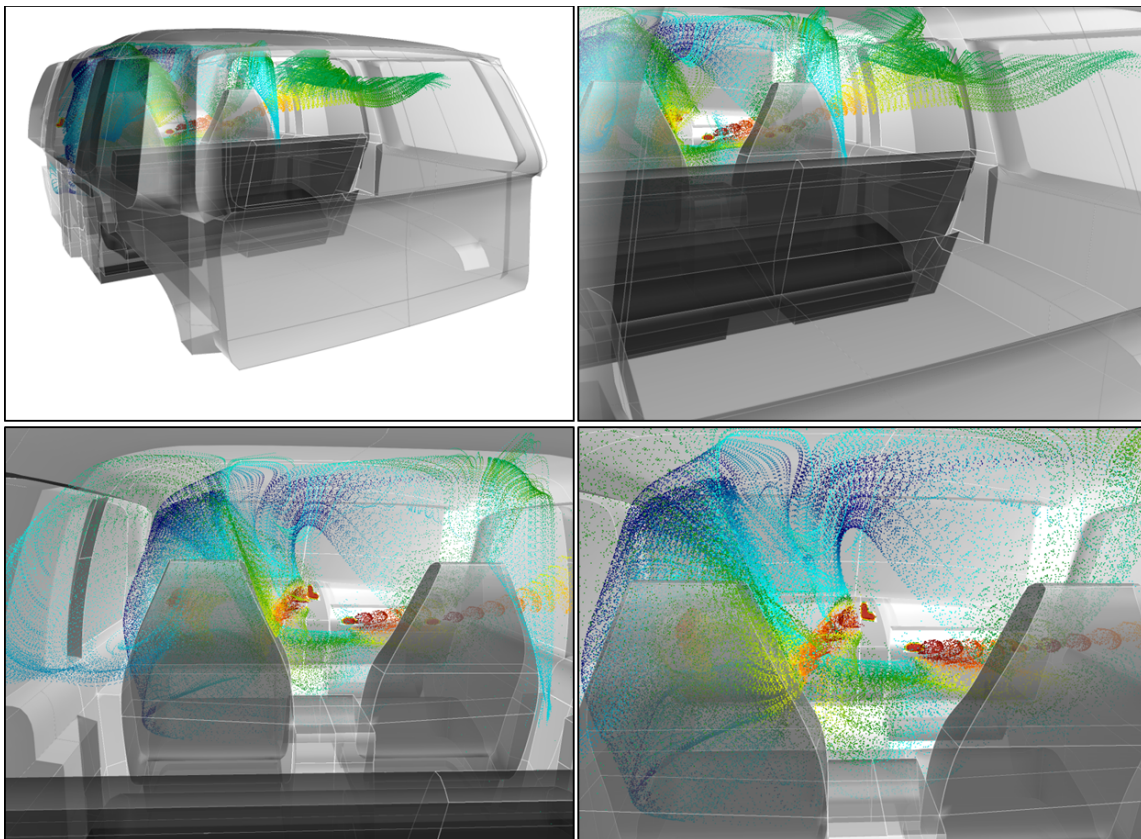
Figure 39 Mouse fly-thru capability

Object: World/Dataset

Note: When transforming the Dataset in relation to the World, the orientation is changed and the only way to restore the original orientation is to press the Reset button. This will set the orientation back to that orientation which existed after the data was read in. It will reset all World transforms done, it will *not* merely undo the most recent action.

Object: World/Dataset Action: Multi-Transform

The transform is specified by clicking on [M1] for XY translation, [M2] for XYZ rotation, and [M3] for magnification/scaling. Actions are performed relative to screen orientation, with the exception of Dataset scaling, which is uniform about the dataset's local origin. The hints on mouse button functionality appear as shown in **Figure 40** below.

Press **[M1]** for com-
bined XY translation
(X translation is hori-
zontal movement, Y
translation is vertical
movement).

Press **[M3]** for: Magnifica-
tion when Object is World,
Scaling when Object is
Dataset.

Press **[M2]** for XYZ rotation.
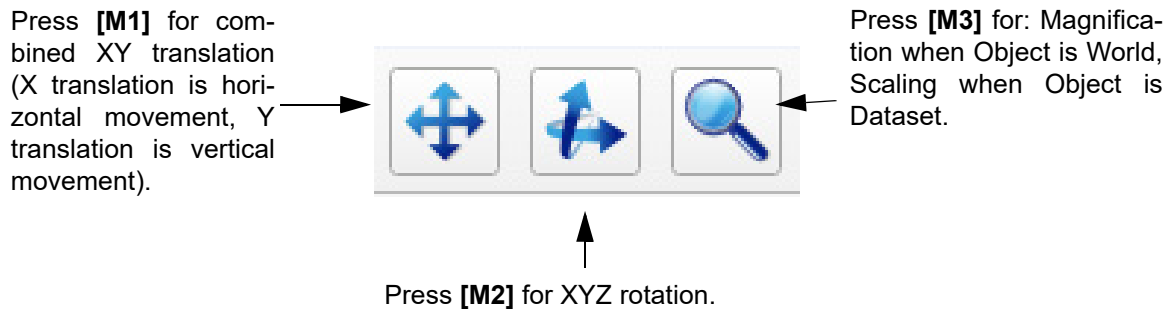
Figure 40 World/Dataset Multi-Transform

If the current Object is World, Dataset, Region, Surface or Light (but not Title or Legend or Plot), and the current Action is Multi-Transform, then an additional set of transform controls can be toggled merely by pressing the Space Bar on your keyboard. This will change the icons and their behavior to those shown in **Figure 41**. Pressing the Space Bar again will return the icons to their previous state.

Note: The 'toggle' mode only functions if the graphics window is the window which has focus. Depending on your computer settings, a window may need to be clicked to gain focus. "Quick-picking" on an object (Shift-M1) will normally cause the panel that comes up to have focus. You may need to click on the graphics window to gain focus there for this feature to work.

Click and drag to create Zoom Box. Action may be repeated for a total of 10 zooms. To Unzoom, you must change the Action to Zoom Box.

Press **[M3]** for: Z translation. This only works when Perspective is turned on.

Press [M2] for Screen Z rotation.

Figure 41 Toggle Transform Icons

When the `-fc` command line option is used when running **FieldView**, the transform specification of the mouse buttons is changed. The FAST-like transforms are accomplished by clicking on [M1] for XYZ rotation, [M2] for magnification/scaling, and [M3] for XY translation. Actions are performed relative to screen orientation, with the exception of Dataset scaling, which is uniform about the Dataset's local origin. In addition, the default demotion mode will be None and the default tracking will change from Track to Running.

Press **[M1]** for XYZ rotation.
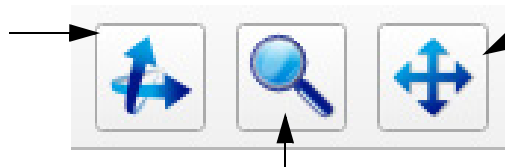
Press **[M3]** for combined XY translation (X translation is horizontal movement, Y translation is vertical movement).

Press [M2] for: Magnification when Object is World, Scaling when Object is Dataset.

Figure 42 FAST World/Dataset Multi-Transform

Again, recognize that you can use the spacebar, when the graphics window has the mouse focus to toggle between the mouse controls shown in **Figure 42** above and those shown in **Figure 43** below.

Press **[M1]** for Screen
Z rotation.

Press **[M3]** for: Z transla-
tion. This only works
when Perspective is
turned on.

Click and drag **[M2]** to create Zoom Box. Action may be repeated for a
total of 10 zooms. To Unzoom, you must change the Action to Zoom Box.
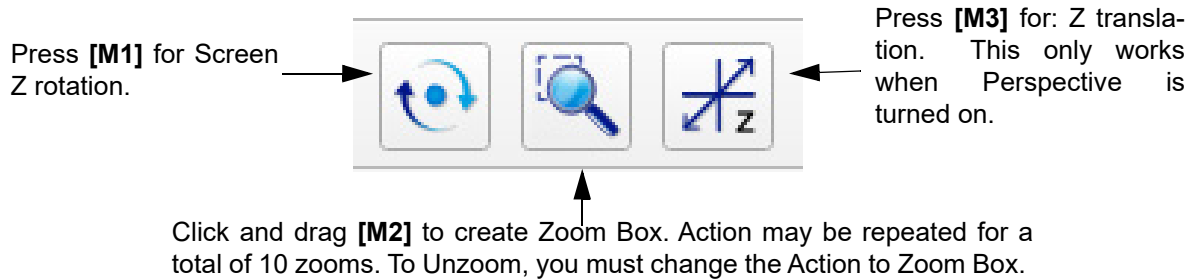
Figure 43 FAST Toggle Transform Icons

Object: World/Dataset Action: Locked Rotation
The transform is specified by clicking on [M1] for Y rotation, [M2] for X rotation, and [M3]
for Z rotation. The rotations are performed relative to screen orientation. When Object is
World, the center of rotation is the center of viewing, which is the center of the screen in
X, Y and at a Z depth set by Locked Translation (see below). When Object is Dataset,
the center of rotation is the local origin of the current dataset.

Press **[M1]** for rotation
about the vertical axis.

Press **[M3]** for rotation
about the screen.

Press [M2] for rotation about the horizontal axis.
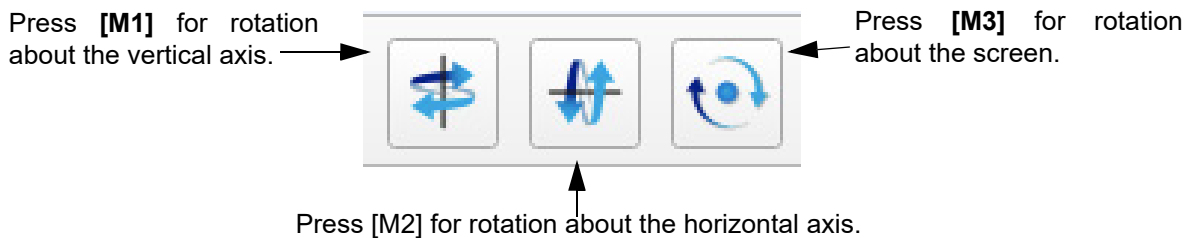
Figure 44 World/Dataset Locked Rotation

Object: World/Dataset Action: Locked Translation
The transform is specified by clicking on [M1] for X translation, [M2] for Y translation, and
[M3] for Z translation. The following actions are performed relative to screen orientation:

Press **[M1]** for X trans-
lation　　　(Horizontal
Movement).

Press **[M3]** for Z transla-
tion.

Press **[M2] f**or Y translation (Vertical Movement).
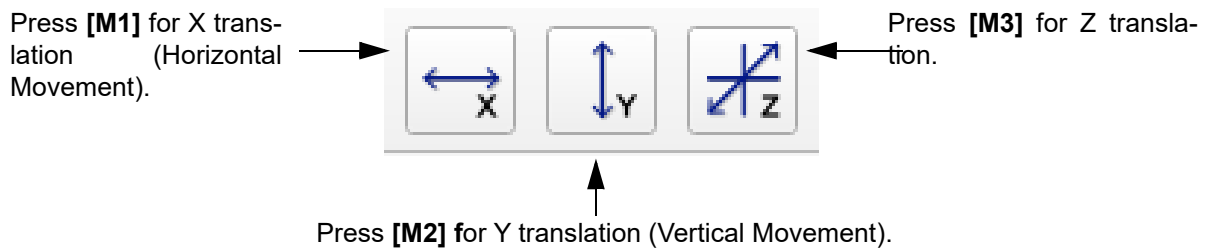
Figure 45 World/Dataset Locked Translation

Note: When Object is World and Perspective is turned off, Z translation is not apparent during interaction. However, this does affect the center of the Object which is apparent when rotating the Object. The Center button will move the World Object back so that the rotation center is at the middle of the World bounding box.

Object: Dataset

Note: When transforming the Dataset in relation to the World, the orientation is changed and there is no undo action to restore the original orientation.

Object: Dataset Action: Dataset Rotation
The transform is specified by clicking on [M1] for Y rotation, [M2] for X rotation, and [M3] for Z rotation. The Axis Marker which appears in the graphics window indicates the orientation of the Dataset's X, Y and Z axes. The following actions are performed about the Dataset's origin and axes:

Press **[M1]** for rota-
tion about the Data-
set's Y axis.

Press **[M3]** for rotation
about the Dataset's Z
axis.

Press [M2] for rotation about the Dataset's X axis.
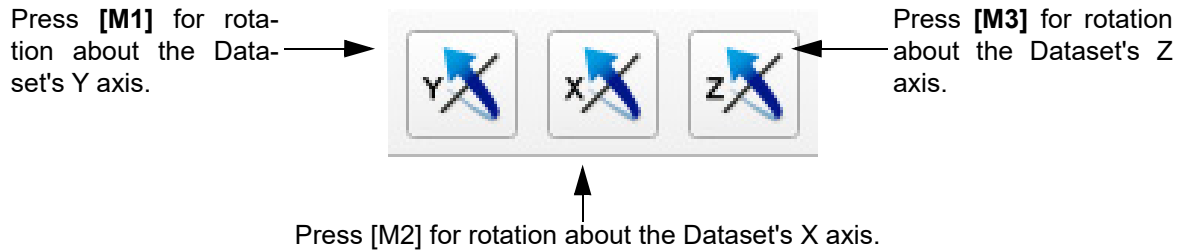
Figure 46 Dataset Rotation

Object: Light

Object: Light Action: Multi-Transform
The transform is specified by clicking on [M2] for XYZ rotation. When Object is Light, a
'light-wand' will appear on the axis marker. This indicates the location of the light source.
Note that **FieldView** uses two light sources separated by 180 degrees, so placing the
light wand behind an object will still lead to lighting from the front. The light wand is
shown in **Figure 47**:
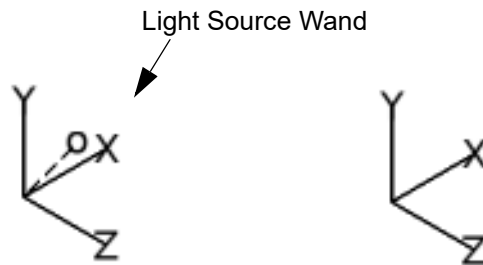
Light Source Wand

Figure 47 Light Source Wand

The following actions are performed about the World's origin and axes:

[M1] is not active. ⟶   ⟵ **[M3]** is not active.

Press [M2] for the XYZ rotation.
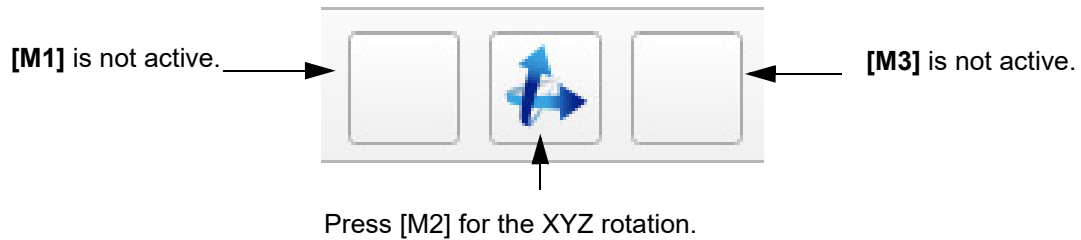
Figure 48 Light Multi-Transform

[M1] is not active. ⟶   ⟵ [M3] is not active.

Press **[M2]** for screen Z rotation.
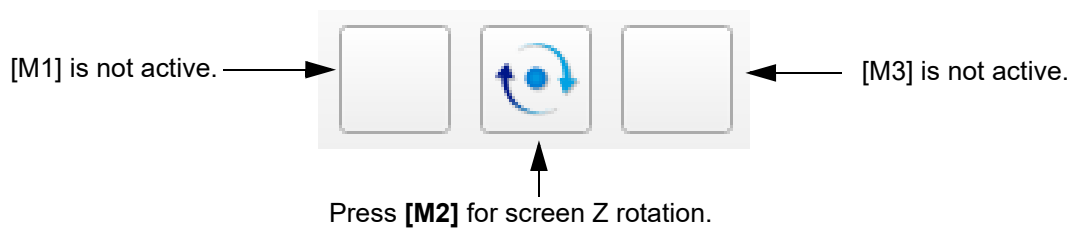
Figure 49 Light Toggle Transform

Object: Plot

Object: Plot Action: Multi-Transform
The transform is specified by clicking on [M1] for XY translation within the plane of the graphics window, [M2] is inactive, and [M3] is used for scaling the embedded plot.

Press **[M1]** for XY
translation (X transla-
tion   is   horizontal
movement, Y transla-
tion is vertical move-
ment).
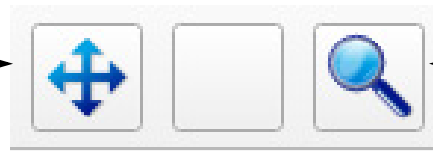
Press **[M3]** for XY scal-
ing. There is a minimum
and maximum bound for
the embedded plot.

[M2] is not active.

Figure 50 Embedded Plot Multi-Transform

Main Toolbar

The Main Toolbar is displayed under the **FieldView** label at the top of the interface as
shown in **Figure 51**. The Main Toolbar provides access to all other panels of the inter-
face.

Press File to access most file input
and   output   operations   including
reading data, saving images and
saving/reading Restarts

Press Visualization Panels to create a surface, 2D
Plot, text annotation or arrow, or Point Probe.

Access online documentation

Press Edit to manage
functions   and   control
scalar colormaps.

Press View to access view-
ing, transform and render-
ing controls.

Press Tools to write XDB files, execute FVX or
Python, or access control panels for Keyframe
Animation, Integration, Flipbook or Transient
Data.
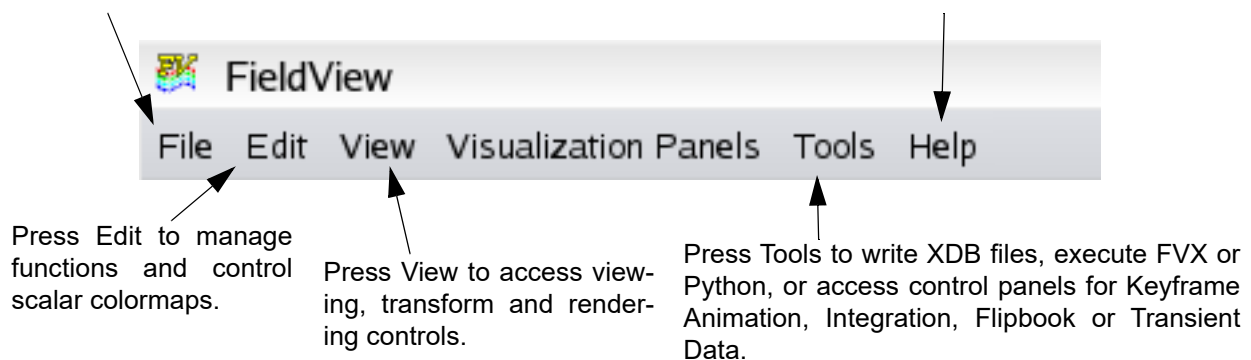
Figure 51 FieldView Main Menu

File

The File pull-down gives you access to the data readers via the Data Input menu, the
ability to save and restore a given state of **FieldView** via the Open and Save Restart
menus, Save Image, Print, Postscript Options and Exit. All of these are described more
fully below.

File > Data Input Menu

The Data Input pull-down menu shown in **Figure 52** offers options which allow you to read in data from a wide variety of public-domain and proprietary formats, including all the major CFD solvers. **Note:** This menu is also where you can select *Local Parallel* for parallel server post-processing, or your own custom server configuration as explained on page 19 in the Installation Guide..
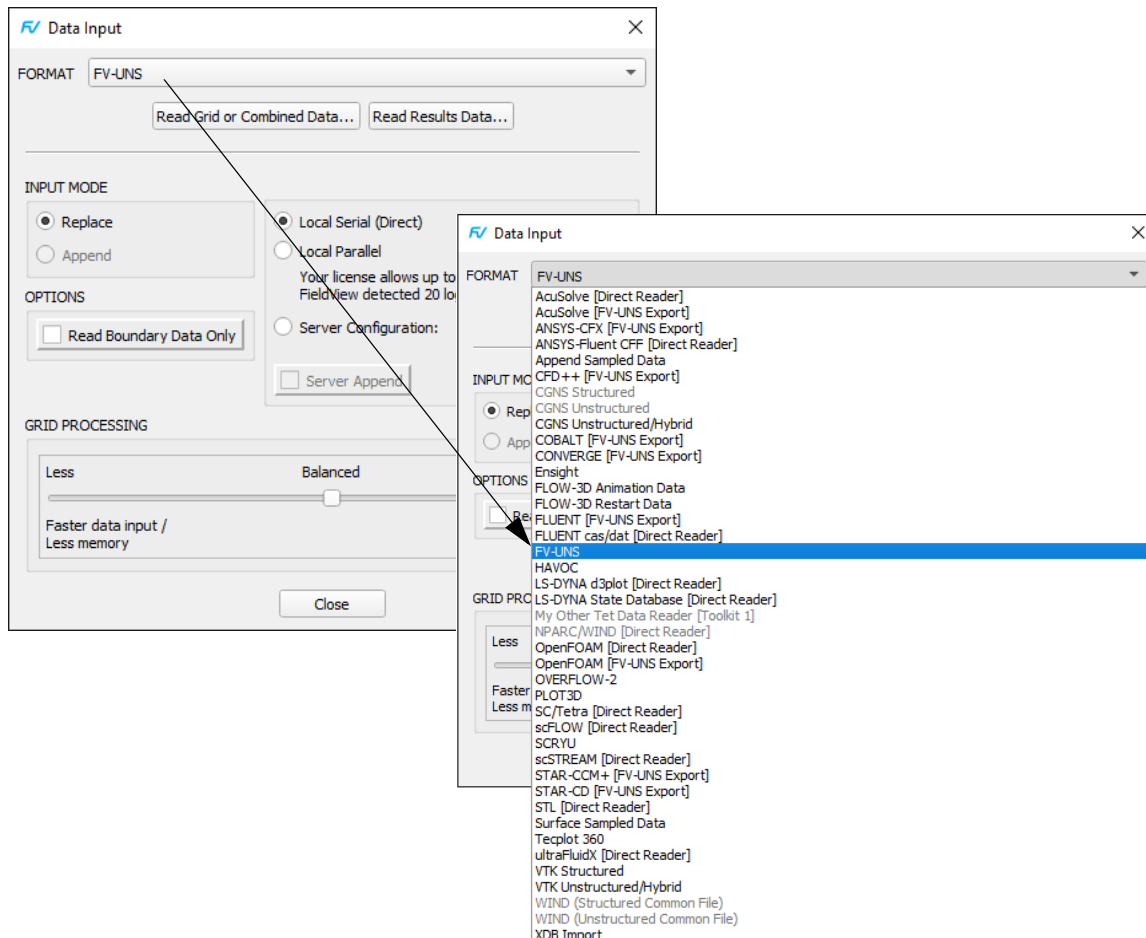
Figure 52 Data Input Pull-down Menu with full options

For a description of the individual readers, see **Chapter 1** of the **Reference Manual**.

File > Open Restart/Save Restart

Complete Restarts of the state of **FieldView** (data read in, surfaces displayed, etc…) can be saved and opened at a later time. Also, restarts of individual surfaces/rakes/views, etc can be created. In this way, a previous view of a particular dataset can be recreated with only one command. Restarts contain information that is applicable to *all* the datasets in memory, not just the current dataset. Therefore, if a Complete Restart or a Computa-

tional Surface Restart is saved, it contains information about all datasets in memory. Separate surface restarts need not be created for multiple datasets (but can, see *Current Dataset* restart type.). See **Chapter 5** of the **Reference Manual** for a complete description of these files. Scripts written in the **FieldView** script (.scr) language are also opened by way of this restart feature. (Note that FVX scripts are run via *Tools.. Execute FVX file*.)
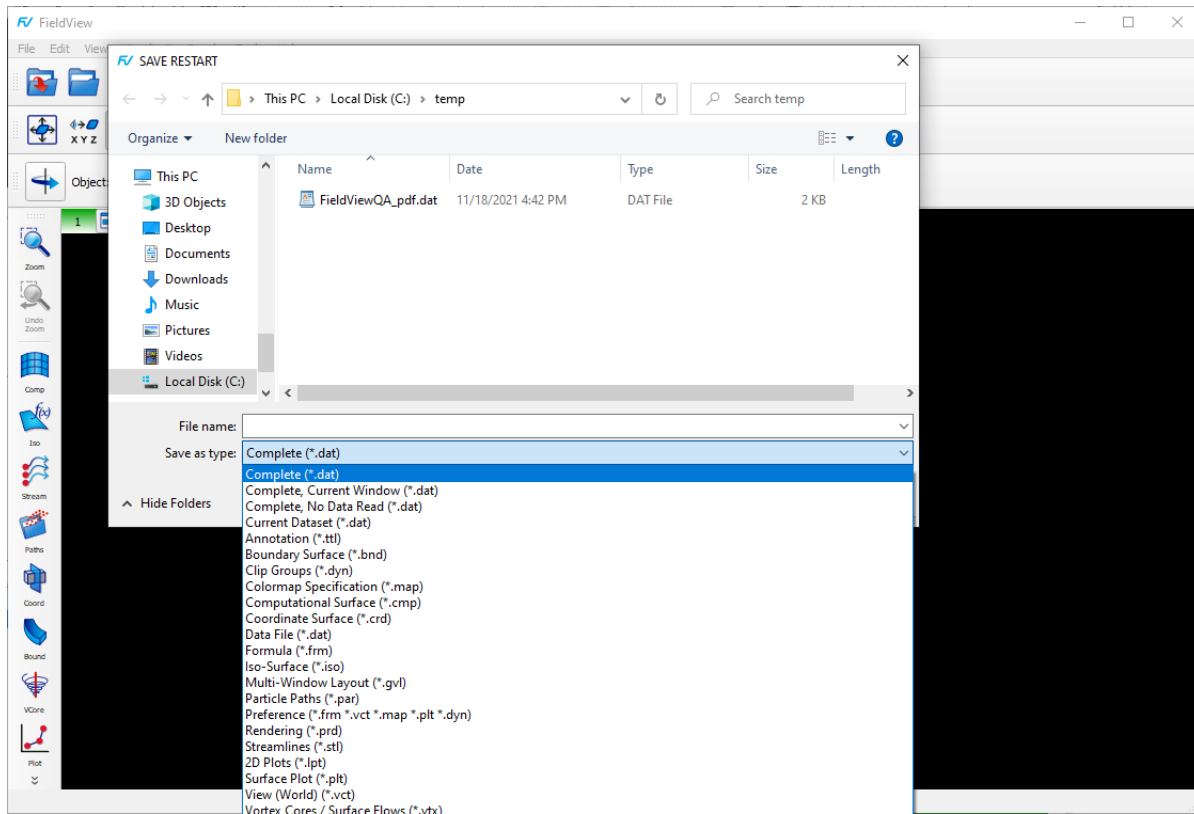
Figure 53 Restart Files Menu

Except for the Complete Restart and Data File Input options in this menu, the names of the datasets that the Restarts are based on are not saved. Thus, using the Restart Files menu, dataset(s) may be viewed in the same way as previous dataset(s) quite easily. See **Chapter 5** of the **Reference Manual** for more information.

File > Save Image

This menu entry lets you save either the contents of the graphics window or the 2D plot window to a file. Several different file formats are supported. Once you have navigated to either the Graphics or Plot sub menu (see **Figure 54**), and have chosen a file format, a file browser is launched letting you specify the name and location of where you want the file to be saved.

Save-Under Operation

FieldView is designed by default to let you save images and flipbooks while the graphics window is covered with another application. For animations that may take some time to create, this gives you the ability to work on other projects simultaneously, thereby increasing your productivity. You can also minimize the graphics window while animations are being created, or while scripts or **FVX** programs are being run.

For WINDOWS users familiar with the FieldView 12 GUI,
This direct Save Image to file replaces the previous action of selecting Print from the pull-down menu, and then selecting the Print to File button on the browser which is launched.
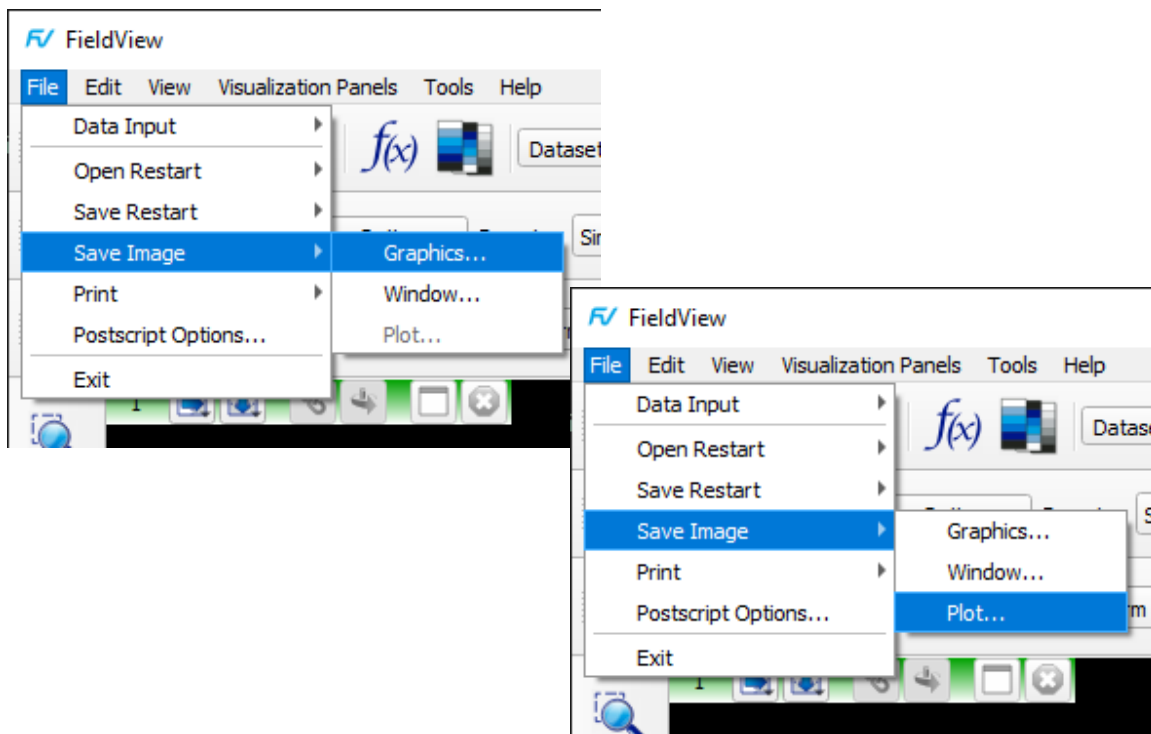
Figure 54 Save Image for Graphics (top) and 2D plot (bottom)

File > Print

The Print pull-down menu is used to output a selected window to a postscript-compatible printer. For LINUX users only there is also a User-Defined Output, if you need output other than postscript. Printing directly to a printer requires additional configuration.
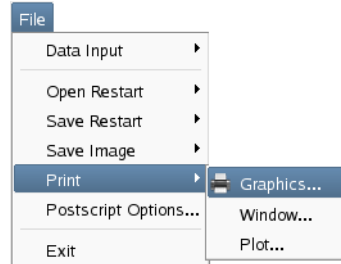
Figure 55 Print menu for direct printing

Two additional options are available for controlling postscript output, and are illustrated in **Figure 56**. These features apply to Print > Window, or Print > Graphics, but not to the Print > Plot option.
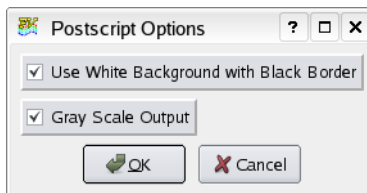
Figure 56 Postscript options

File > Exit

Select Exit in order to exit the **FieldView** program entirely and return to your normal workstation display. Before exiting, the program presents an Exit Confirmation window requesting you to confirm or Cancel. Press the OK button in order to exit the program.

Edit

The Edit menu pull-down is illustrated below.
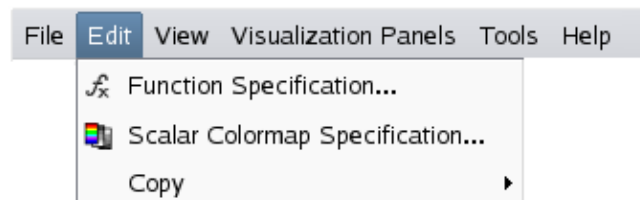
Figure 57 Edit pulldown menu

Edit > Function Specification

The Function Specification button brings up the Function Specification panel as shown below. This panel is used to define which Scalar or Vector quantities you wish to visualize. You may also define your own quantities using the Formula Specification panel, which is available upon pressing the Create button. The Function Specification Panel is redundantly accessible from the Main toolbar (see **Figure 21**).
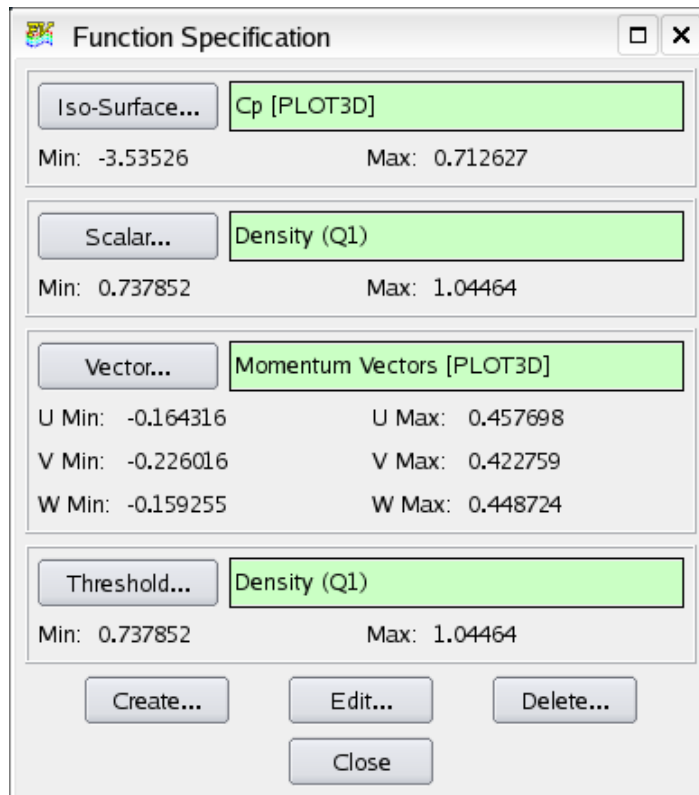
Figure 58 Function Specification panel

Edit > Scalar Colormap Specification

The Scalar Colormap Specification button brings up the Scalar Colormap Specification Panel shown in **Figure 59**. This panel may be accessed using the Scalar Colormap Specification icon from the Main Toolbar (see **Figure 21**). This functionality is also redundantly available on the Colormap tab for all visualization panels; please refer to for additional details. The visibility state (either on or off) as well as its location are saved as part of the **FieldView** preferences on exit, and restored for the next **FieldView** session.
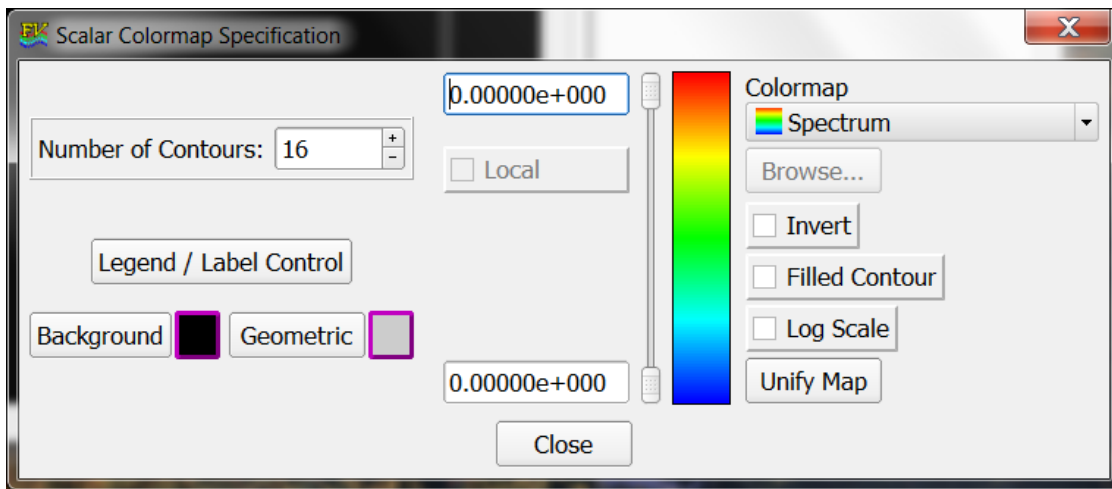
Figure 59 Scalar Colormap Specification Panel

Edit > Copy

This feature allows you to copy the **Graphics Window** or **Plot Window** to the clipboard as a bitmap (BMP file) for use in other image applications, documents, etc.

View

The View pull-down menu (see **Figure 60**) gives you options to adjust the views in **Field-View** in many different ways. Some of the items in this menu bring up separate panels with further controls and choices. These items are indicated by the ellipses after the name (e.g. Defined…). Others are merely checked *on* or off (e.g. Axis Markers). Still others perform operations or turn on special mouse operations (e.g. Center or Set Center of Rotation).

The buttons which launch panels are discussed in detail a little later on. The single action buttons on this menu include:

The Refresh button (redundantly accessible on the Transform Controls Toolbar, see **Figure 22**) lets you refresh the graphics window at any time.

The Zoom Box button (redundantly accessible on the Side Toolbar) changes the operation of the mouse to a rubber box, letting you left-click and drag to zoom into any part of your graphics window. The Undo Zoom undoes the most recent Zoom action.

The Center button (redundantly accessible on the Transform Controls Toolbar, see **Figure 22**) automatically centers the contents of the current window.

When you click on either the Set Center of View or Set Center of Rotation button, you should see your cursor change to a cross-hair. At this point you need to make a selection in the current graphics window by holding the [CTRL] key and clicking with the left mouse button [M1]. When this action is complete, the Defined Views panel is launched. From this panel it is possible to make further changes to the Center of View or the Center of Rotation by using the icons on that panel. The full functionality of this panel is described in detail later in this section.

Locked Transforms lets you toggle the mouse transform to change to the locked transform actions.

Repeat works when the state of Locked Transforms is active and the right mouse button icon on the Locked Transforms toolbar is de-selected. With the repeat button turned on, clicking on any one of the Locked Transform actions cause that action to repeat until the repeat button is turned off again. CAUTION: Selecting a Locked Transform transformation means that your graphics window contents will continue to rotate/translate until repeat is turned off. Therefore, we do not recommend that you use Repeat in conjunction with Locked Transform transformations.

Previously described, Running mouse lets you toggle its state on or off (see **Running Mouse**). Running mouse control is redundantly accessible on the Viewer Options panel.

Axis markers lets you toggle their state of visibility. Axis marker visibility can be redundantly accessed on the Viewer Options panel.

Perspective lets you toggle Perspective on or off. To change the perspective angle, you need to go to the Viewer Options panel, where you can also turn the state of perspective on or off.

Outline lets you toggle the state of the outline either on or off. The outline state can be redundantly controlled from the Viewer Toolbar and the Viewer options panel.

Presentation quality lets you toggle its state. By default this is off. Presentation options can be set by navigating to the Rendering Options panel.

Quick transparency can be toggled on or off. The current default is to use high quality transparency.

Anti-aliasing can be toggled on or off. Anti-aliasing is on by default.
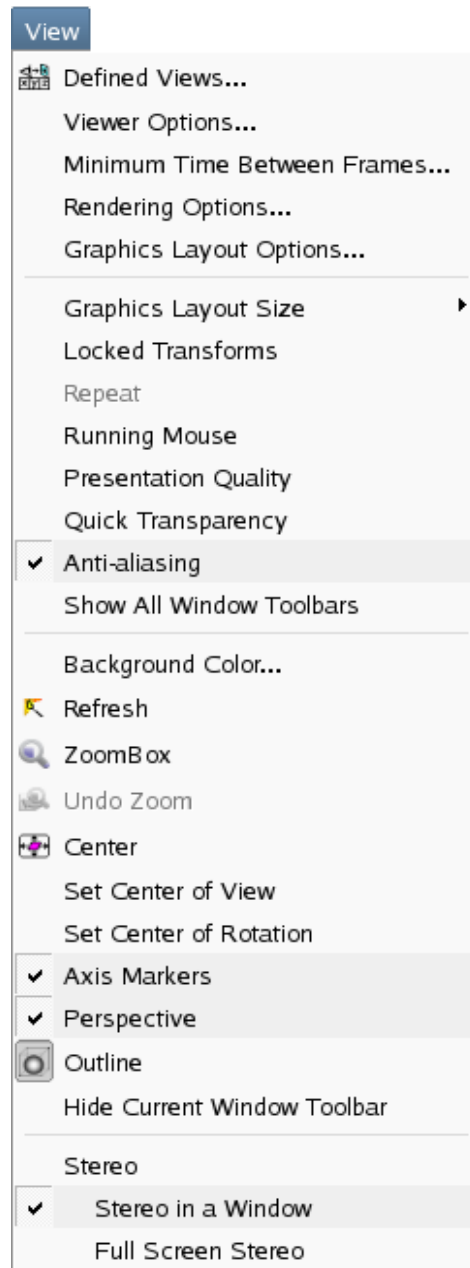
Figure 60 View menu pull-down

View > Defined Views

Defined Views (redundantly accessible on the Transform Controls Toolbar, see **Figure 22**) are applied to the current window only. Pre-defined views can be specified by clicking the appropriate buttons under the VIEWING DIRECTION and ISOMETRIC sections of this panel, which can also be used to set the Center of View or the Center of Rotation. To select the center point interactively, click CTRL-M1 (Control key + left mouse button)

in the current window (interactive selection is redundantly accessible directly on the View menu with Set Center of View and Set Center of Rotation, see **Figure 60**). Alternatively, X, Y and Z values can be typed into the fields in the SET CENTER section of the panel. Then click on either the View or Rotation button to make the selection. Setting the Center of View can be performed irrespective of the current Object and will set the World view. Set Center of Rotation will cause the Object: to change to World. If a dataset is 'picked' which is not the current dataset, then **FieldView** will 'beep.'

Press one of the six VIEW-ING DIRECTION buttons for an orthographic view. The sign denotes the direction from which the dataset will be viewed. The center point of the view is preserved in all cases.

The Center of Rotation (of the World only) or Center of View can be entered numerically (in coordinates relative to the current dataset) into these fields, or chosen by clicking in the graphics window with CTRL-M1 and then choosing with View or Rotation.

First, choose a point in the graphics window that you wish to use as the new center. This is done with a CTRL-M1 at the desired location. Then press View to make the selected point the center of view. This has the same effect as the Center button but operates on the newly chosen center of view. Exact values may also be typed into the fields.
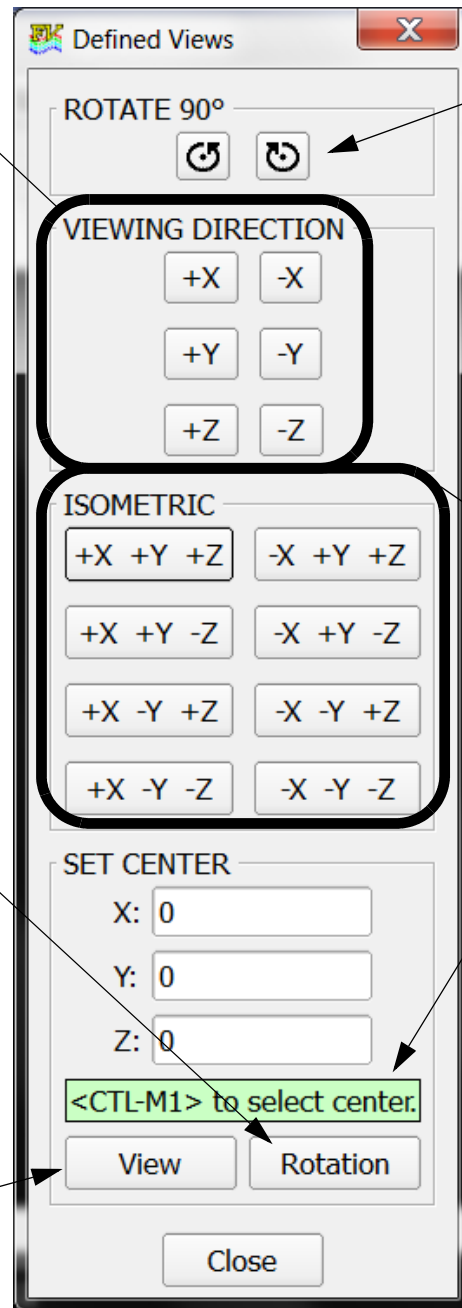
To facilitate user interaction, these buttons rotate - or + 90 degrees around the screen axis. They can be used at any time and apply to any view, and are especially useful for changing which axis is up after applying a "VIEWING DIRECTION" action.

Each of the eight ISOMETRIC buttons have three axes specified and create an isometric view from the specified octant. The default view in **FieldView** is an isometric view from -X, +Y, +Z, which means the vector -1, +1, +1 points out of the screen.

Use the Control key and the left mouse button in the graphics window to interactively set the Center of View or Rotation.

Pressing Rotation after having chosen a new center (with CTRL-M1 in the graphics window, or typed in) will change the center of rotation for the World.

Figure 61 Defined Views Panel

View > Viewer Options

The Viewer Options panel is shown in **Figure 62**. Redundant access to the Running Mouse, Axis Marker and Outline toggles are provided here. They are also available
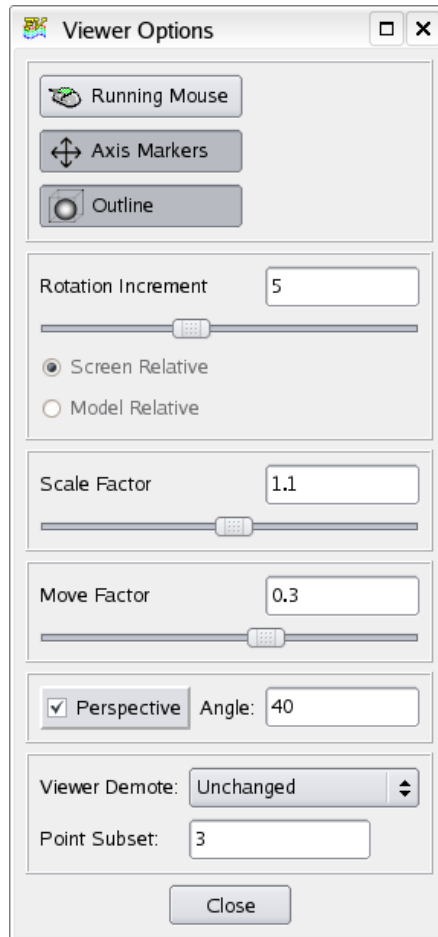
Figure 62 Viewer Options Panel

directly from the View menu pull-down. The Rotation Increment, Scale Factor and Move Factor settings are designed to work with Locked Transforms.   With the Right Mouse button de-selected on the Locked Transforms panel, clicking any one of the icons causes the contents of the graphics window to rotate, scale or move (translate) according to the factor specified in this panel with the appropriate slider. All of these sliders use a nonlinear scale to permit fine-tuning of the position of your dataset.

The state of Perspective (either on or off) is controlled here, in addition to the View menu pull-down. The degree of perspective is specified here. The Viewer demotion is also specified here. The point subset used for the point demotion mode can only be specified here.

View > Background Color

The Background Color panel will be covered in detail in **Chapter 3**.

View > Minimum Time Between Frames
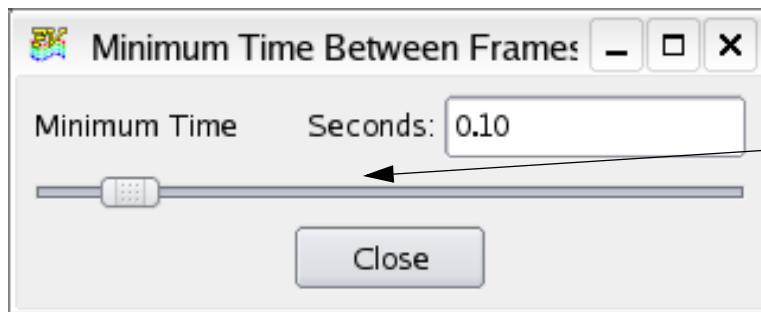
Moving the slider, or typing in the Seconds field controls the minimum time delay between frames of an animation. This control is used to slow down an animation (such as sweeping a surface) that is moving too quickly. The time shown on this panel is the minimum time delay, the actual time between frames may be higher due to system resources and usage.

Figure 63　Minimum Time Between Frames

View > Rendering Options

Selecting Rendering Options will open the Rendering Options panel (**Figure 64**).

Presentation Quality
These controls allow you to create more realistic images by providing a variety of lighting options. You can add "shine" (highlights) to surface types of your choice as well as making streamlines/rakes appear more 3D. For more information about Presentation Rendering with streamlines/rakes, see **Chapter 6**.

.

Shine controls:

Intensity: This slider sets the amount of shine: 0 being no shine, 1 being maximum shine.

Highlight size: This slider sets the area extent covered by the highlight; 0 means no area and 1 means maximum area. The variation of area highlighted is proportional to the amount of curvature. Flat surfaces will be entirely highlighted.

These checkboxes control whether Intensity and Highlight Size apply to a particular surface type, or object type. If the box is checked, then the options are applied. For example, if Iso-surface is checked, then all visible Iso-Surfaces will be rendered with specified SHININESS and Highlight Size.

Any changes that are made to the parameters or surface/rake types activation toggles require that the Refresh button be pressed. Alternately, the Refresh button on the Main Toolbar can be used.
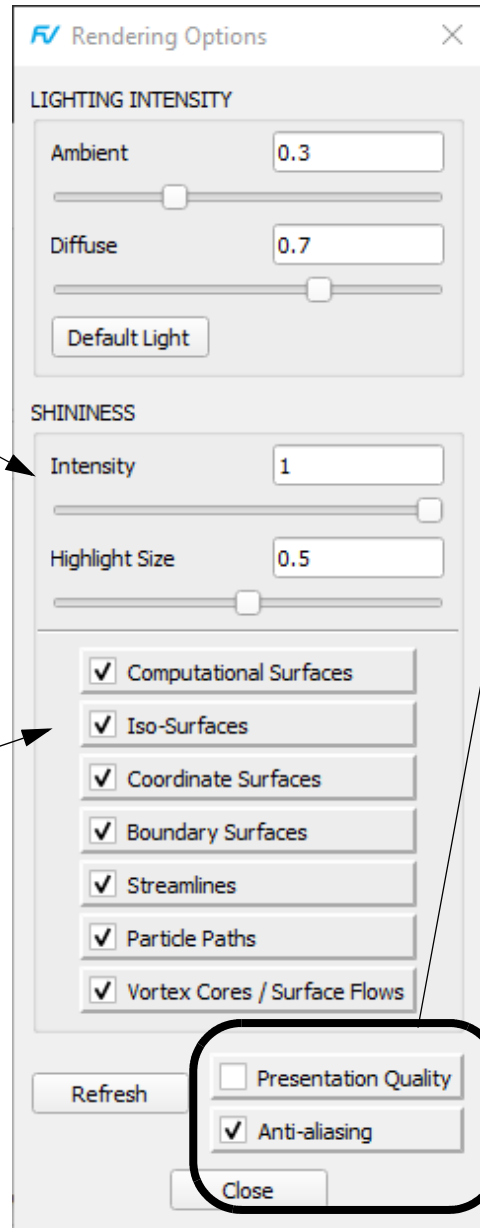
To activate the Presentation Rendering feature, the Presentation Quality button needs to be checked. This check item is also on the View pull-down menu.

These two toggles do not require the Refresh button. Their settings overwrite settings of seven toggles above for particular object types.

Figure 64   Rendering Options panel

Lighting Intensity

Two lighting parameters provide for control over lighting intensity or brightness. These intensity parameters affect the Ambient and Diffuse lighting components and along with the Specular lighting component (controlled by the Shininess Intensity and Highlight Size parameters), determine the pixel color value in the rendered image.

Note that these lighting controls will not change the appearance of constant shaded surfaces which are rendered using the scalar or geometric color.

The figure at top right (see **Figure 65**) consists of a series of renderings with varying diffuse and ambient intensity. As the diffuse light is increased, the sphere appears as it would when it is illuminated by a brighter light. As the ambient light is increased, the overall scene, including the sphere appears brighter.

In the figure at bottom right (again, see **Figure 65**), increasing the specular lighting will increase the size of the shiny spots on the sphere. As in the figure above, increasing the diffuse lighting serves to increase the illumination of the sphere.

In both of the examples it should be noted that a change to any one of the ambient, diffuse, intensity or highlight size parameters (using either the slider or a type-in value) will lead to a change in final rendered appearance of the image.

Note that the values of ambient, diffuse, intensity and highlight will have no effect on the background color or a background image.
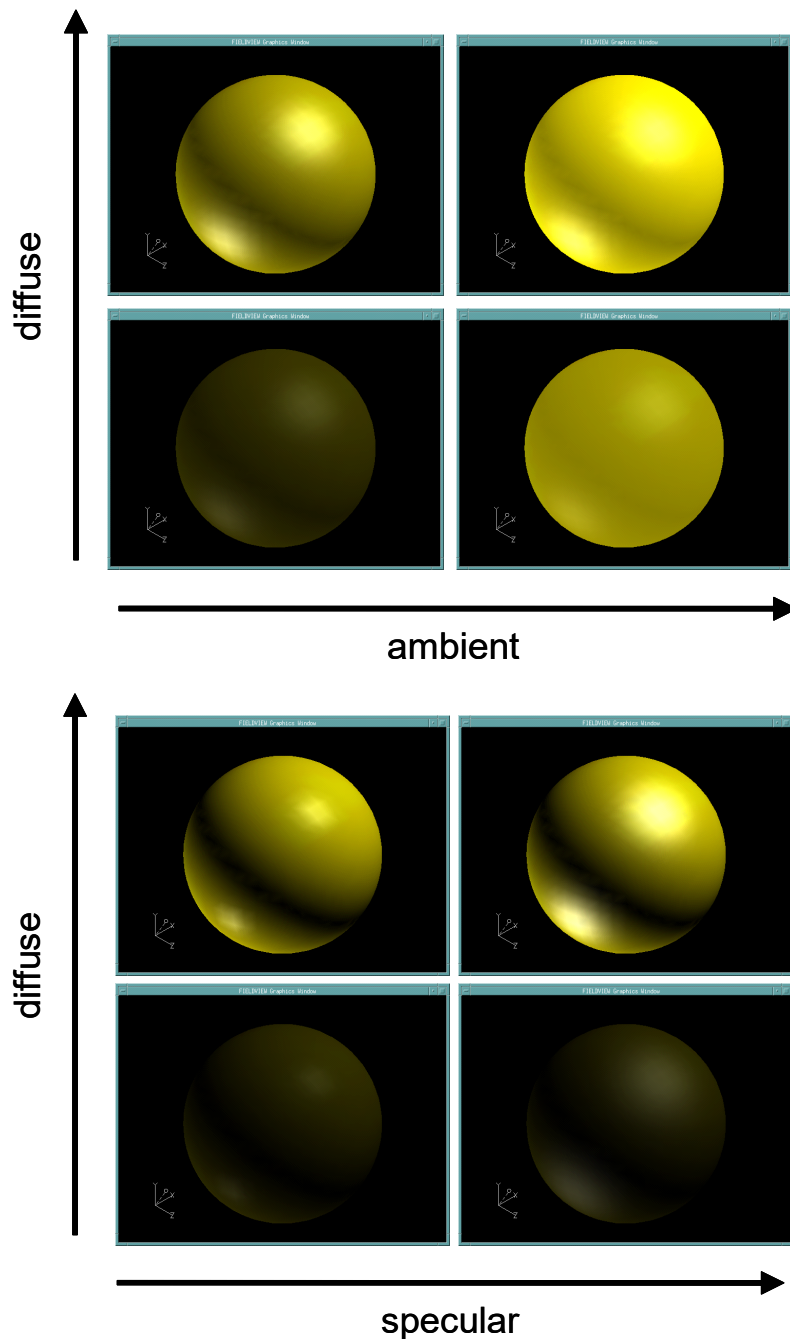
Figure 65   Examples of lighting effects

When the Rendering Options panel is opened for the first time, the default values corresponding to the fixed settings previously used in **FieldView** for Ambient (0.3) and Diffuse (0.7) levels are applied. Presentation rendering restarts from previous versions of **FieldView**, when read, will apply these lighting defaults. The values that you set for Ambient and Diffuse will be saved as part of the presentation rendering restart, and will be restored when the restart is read back into **FieldView**.

Ambient Lighting

Ambient light is a form of indirect illumination resulting from light reflecting off objects in an environment, and is generally described as an overall or uniform light level. **FieldView** simulates this component of lighting by combining the color of the rendered surface with an Ambient Intensity value which is user-adjustable via a slider and corresponding type-in field in the Rendering Options panel. Valid parameter values range between 0 and 1, where 0 sets the darkest Ambient Intensity and 1 sets the brightest. Note that adjusting this setting may decrease the contrast range of the surface shading, and use of extreme values may produce renderings that appear under- or over-exposed. The default Ambient Intensity of 0.3 matches the value used in previous versions of **FieldView**.

Diffuse Lighting

Diffuse light is a form of direct illumination in which the shading of a rendered surface is determined by evaluating the surface color with the light intensity based on the angle between the surface and the light. The Diffuse Intensity is user-adjustable via a slider and corresponding type-in field in the Rendering Options panel. Valid parameter values range between 0 and 1, where 0 sets the darkest Diffuse Intensity and 1 sets the brightest. The default Diffuse Intensity of 0.7 matches the value used in previous versions of **FieldView**.
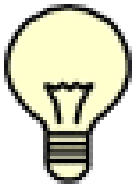
Note: You will need to press the Refresh button on the Rendering Options panel in order to see the settings applied for Ambient and Diffuse light level changes after they have been changed.

Transparency Rendering in FieldView

There are two types of transparency rendering in **FieldView**: Quick, and the default, Standard transparency. The rendering method known as Precise in versions of **FieldView** before 13 is now the standard method for rendering transparency.

Quick transparency is designed to create images as quickly as possible and to require the minimum amount of graphics board resource.   Technically speaking, it is a simple screen pattern method that does not do any sophisticated sorting of surfaces. (Before **FieldView** 13, the default transparency method used a type of hardware sorting that is being phased out in graphics boards.) Without sorting, the blending of object colors

when you have multiple transparent surfaces can sometimes be inaccurate and misleading. However, this is the method that almost all CAE software uses for transparency.
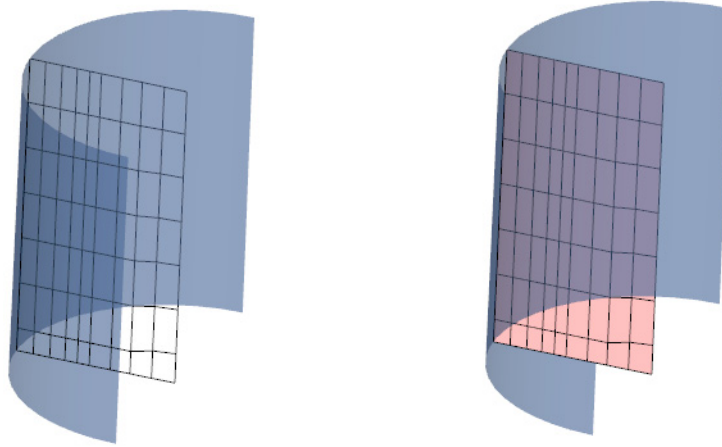
Figure 66 Typical Error with Multiple Overlapping Transparent Surfaces

In **Figure 66** above on the left, a blue half cylinder is made transparent. You can see the darker blue where the cylinder passes in front of itself. When the surface inside the cylinder is changed from mesh to a transparent red surface, you can see the cylinder through it, but the section of the cylinder where it passes in front of itself is no longer visible as dark blue. This is typical of the type of error that occurs with multiple overlapping transparent surfaces.

The standard transparency rendering in **FieldView** 13 is a GPU accelerated version of the Precise transparency found in previous versions of **FieldView**. This method can accurately account for blended colors for up to eight overlapping transparent surfaces. Because graphics hardware is used, there should be little effect on interactive rendering time for high quality scenes.
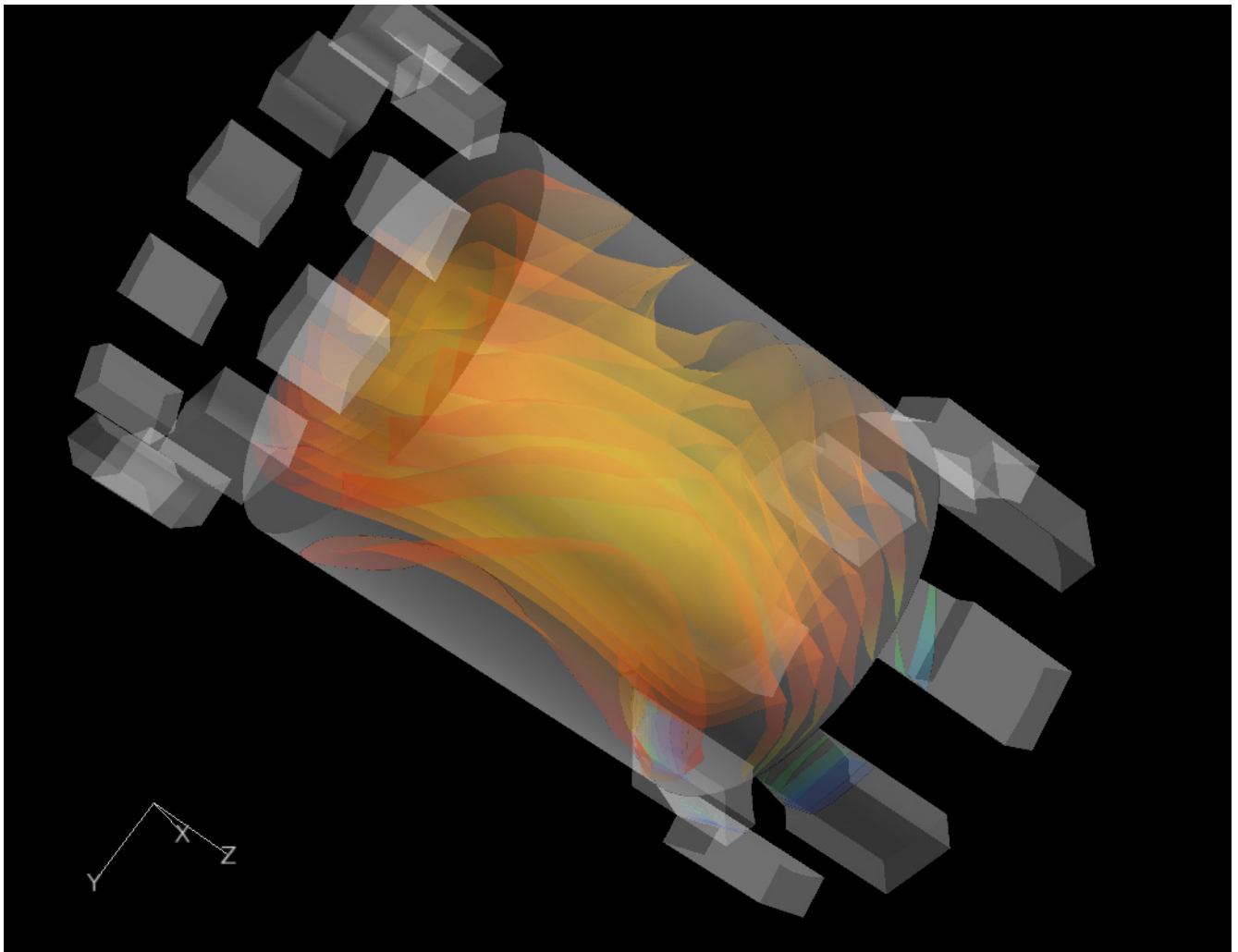
Figure 67 Standard Transparency Rendering in FieldView

The example above shows multiple overlapping smooth-shaded iso-surfaces, with different colors on each. The image clearly shows those surfaces in an almost 'soft' looking accurate rendering.

A side effect of this rendering method is the need to allocate extra memory within the graphics card for each layer of transparent surfaces. The number of layers can be in controlled with an environment variable, FV_TRANSP_LAYERS, which has a range from 2 to 8. This can be used to accommodate older legacy graphics boards with less than 128Mb of memory.

View > Anti-aliasing

Support for anti-aliasing is available for on screen and hardcopy rendering. The effect of on-screen anti-aliasing is illustrated in the figures below:
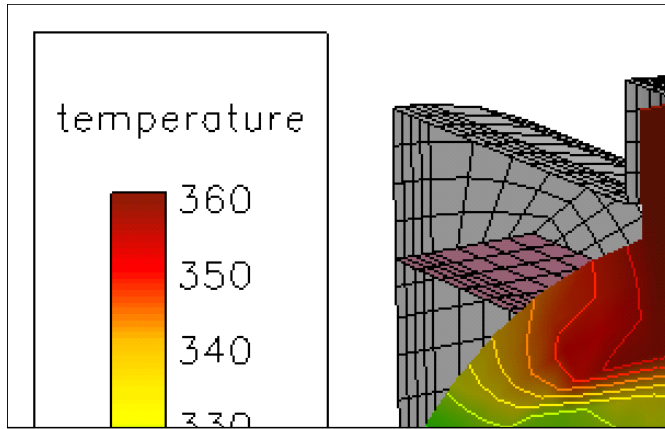
(a) Anti-aliasing is OFF
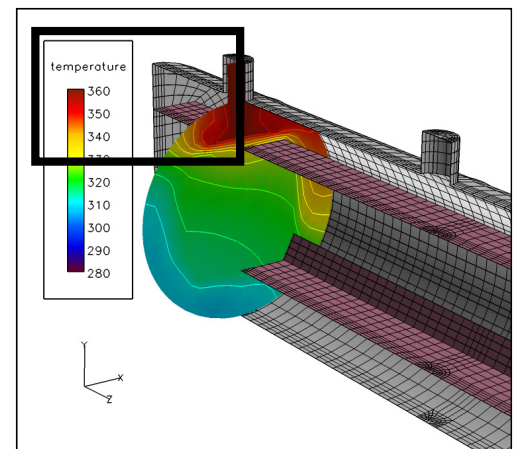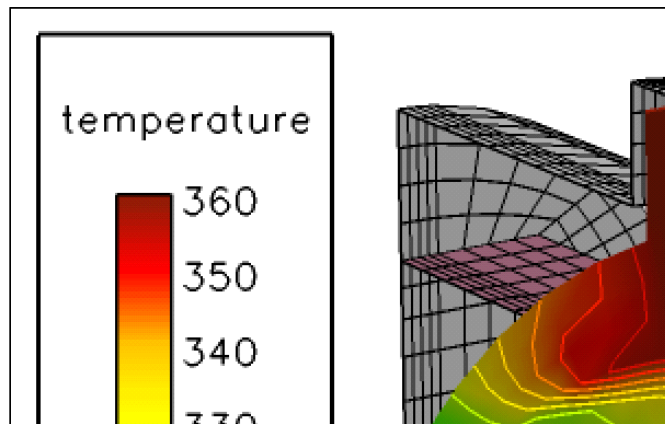
(b) Anti-aliasing is ON

Figure 68   Dataset displayed with Anti-aliasing OFF (a) and ON (b)

When anti-aliasing is on, all lines and vertices will be displayed thicker and will be smoothed. Objects within the **FieldView** graphics window that will be subject to anti-aliasing include things such as mesh lines, contour lines, outline edges, vectors (2d and non-presentation render 3d), fonts and vertices. Vertices look sharper when anti-aliasing is turned off, while turning it on contributes to a "point cloud" effect (see **Figure 69**). On-screen anti-aliasing can be controlled either from the View pulldown, or from the Rendering Options panel (see **Figure 64**). By default, screen anti-aliasing is turned ON.
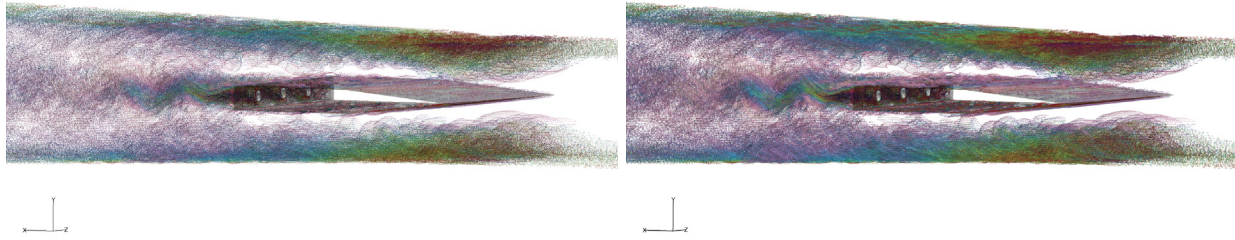
Figure 69 Vertices with Anti-Aliasing Off (left) and On (right)

Known limitations
Anti-aliasing for hardcopy Postscript and emf output will be available regardless of the platform. The quality of Postscript and emf output is controlled by the `FV_DPI` environment variable. If this variable is set to a high value, such as 300 dpi, the time to create Postscript and emf output will increase considerably, and in some instances may fail completely. To disable anti-aliasing for hardcopy output, use the environment variable, `FV_PRINT_NO_AA.`

Full screen images that have anti-aliasing applied (as per the default settings) will render slowly and in some extreme instances **FieldView** may crash due to memory resource limitations. The resources available for rendering will depend on the system graphics card and the size of the image making it difficult to predict ahead of time when problems are likely to occur.

View > Stereo

FieldView supports quad-buffered (active) stereo which derives its name from the fact that four image buffers are used. Quad-buffered stereo can require a higher bandwidth monitor/projector compared to non-stereo operation.

What do I need to use stereo?

1. Your graphics card must support quad-buffered (active) stereo. This information is usually available via the manufacturer's specifications. In stereo mode, the graphics card specifications will also tell you what *screen resolution* and *color depth* is supported for stereo. It is usually less than the full screen, high resolution that may be used on your desktop by default. In order for your **FieldView** pictures to look correct in stereo, you want to choose a stereo mode that provides a 24-bit color depth, also known as *full color*.

2. Your monitor and/or projector must support stereo display refresh rates at a good screen resolution. The manufacturer's specifications should contain information about *monitor refresh rates*. A 96Hz refresh rate is a common requirement for stereo.

This is a faster refresh rate than is used for most desktop applications. Check to see that your monitor supports the *stereo refresh rate* at the *stereo screen resolution.*

3. You will need to acquire the proper stereo glasses and transmitter from a vendor such as RealD for viewing quad-buffered (active) stereo. Alternatively, the active stereo signal can be run through a converter (e.g., Viz3D from www.cyviz.com) for passive stereo projection displays.

Setting up your Stereo Configuration

Stereo is enabled by setting an environment variable `FV_STEREO_ENABLE` in the system settings on your computer.

Running Stereo Display

You can either choose **Stereo** in the **View** pull-down menu, or press the [Enter] key while the mouse pointer is in the graphics window. You have two distinct display modes to choose from:

1. **Stereo in a Window** limits the stereo effect to the graphics window while other GUI elements remain on the screen in non-stereo. Note that 'quick-picking' on surfaces and rakes will work in this mode of operation.
2. **Full Screen Stereo** causes the graphics window to fill the screen and other GUI elements will not be visible. 'Quick-picking' is disabled in this mode of operation.

The appropriate menu item for **Stereo in a Window** or **Full Screen Stereo** must be selected to go into the corresponding stereo mode. Once selected, stereo will be displayed automatically   The default menu selection is for **Stereo in a Window**.

When **FieldView** is running in stereo mode, the visibility of all annotations, legends, embedded plots and the axes markers are turned off.

Stereo will work with the default graphics mode, as well as with the 'Reduced Functionality' graphics mode (this is enabled with the environment variable `FV_RF_GRAPHICS`).

In this case Stereo in a Window has been chosen
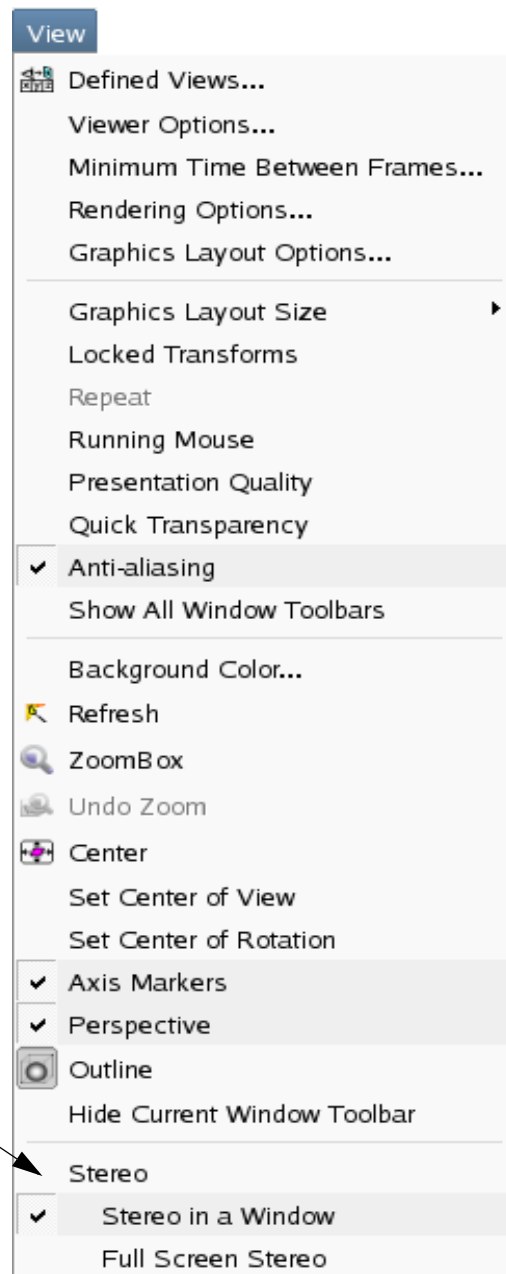
Figure 70 Stereo Control on the View Pulldown menu

Since the **View** menu is inaccessible while the transient sweep is on, the **Stereo** button must be turned on in the **Transient Data Controls** panel, which can be found in the **Tools** sub-menu:
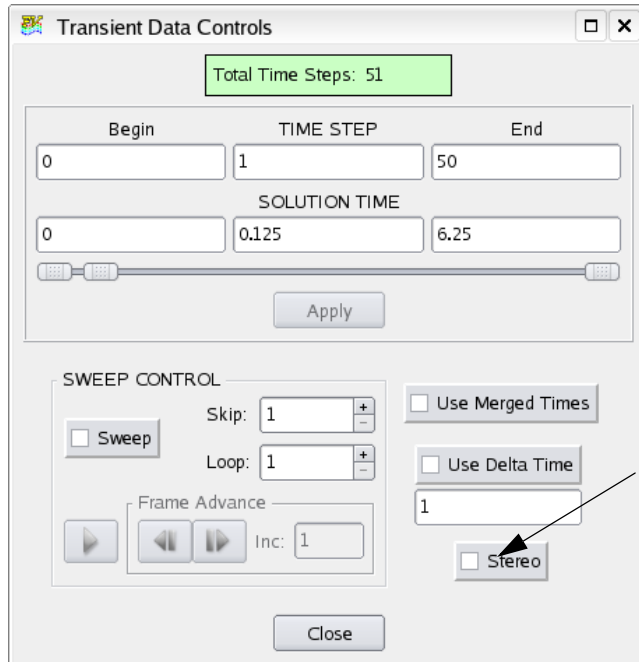
Figure 71 Stereo Menu for Transient Data

Eye Separation

In order to properly visualize the full 3D effects while in stereo, eye separation and screen distance parameters may need to be adjusted. The arrow pad, or numeric pad keys, are able to adjust these parameters. The left/right arrow keys decrease/increase eye separation. The down/up arrow keys decrease/increase the screen distance. Since these values are dataset specific, they are saved and restored to and from data file restarts.

Ending the Session

The [Esc] or [Enter] keys cause **FieldView** to exit stereo mode. These must be pressed while the cursor is in the graphics window and the graphics window has focus.

Visualization Panels

The Visualization Panels pull-down menu offers ten panel options. The Computational Surface panel is visible as part of the default startup display for structured grids, and the Coordinate Surface panel is the default for unstructured grids. Each Visualization panel is discussed in detail in subsequent chapters.

| File | Edit | View | Visualization Panels | Tools | Help |

- Computational Surface...
- Iso-Surface...
- Streamlines...
- Particle Paths...
- Coordinate Surface...
- Boundary Surface...
- Vortex Cores / Surface Flows...
- 2D Plots...
- Annotation...
- Point Probe...
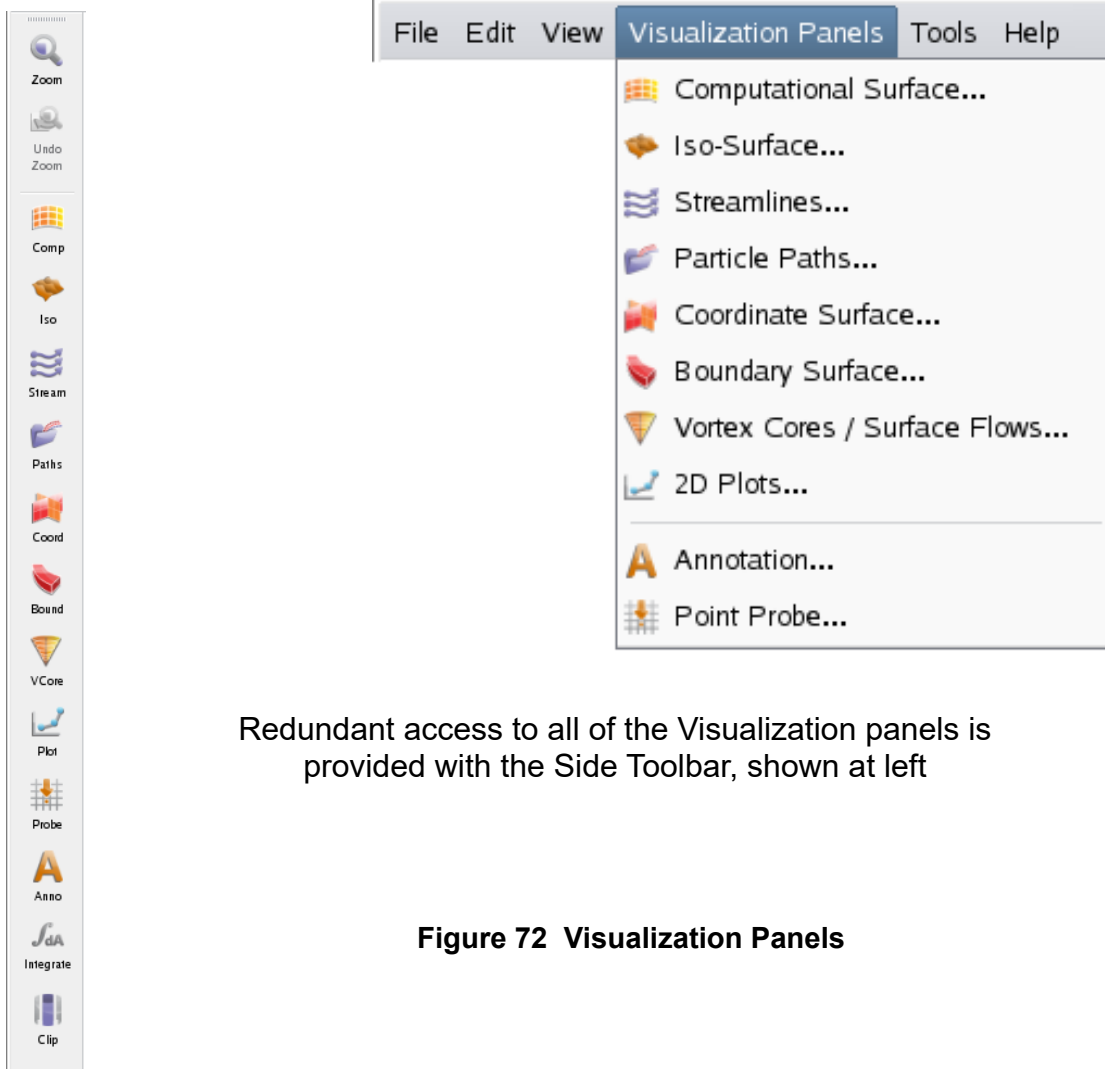
Redundant access to all of the Visualization panels is
provided with the Side Toolbar, shown at left

Figure 72 Visualization Panels

All visualization panels in **FieldView** are designed to work in a similar way, maintaining a
common look and functionality set.

You may select a surface, rake or title interactively in the graphics window by "quick-picking". This is done by double-clicking the left mouse while the cursor is over the surface you want to select. You can also select the surface or rake by its ID number.

The area is used for Creating or Deleting surfaces or rakes. To show any surfaces or rakes, the Create button must first be pressed. Pressing Delete deletes the current surface. Clear All gives a confirmation prompt, and then, if confirmed, deletes all surfaces of this type for the current dataset only. These buttons function the same way on all panels for all objects in **FieldView**. The surface can also be displayed as vectors

Choose the Display Type here. Options for **Coordinate** surfaces (as shown) include: Constant, Smooth, Mesh, Contours, Crinkle, Vertices and Shaded Vertices. Options for other object types differ, and are described in the corresponding sections of this document.

Shows the locations and provides labeling of the minimum and/or maximum values of a scalar on the given surface.

Sets the Transparency of the surface. A value of 0.0 is fully opaque, while a value of 1.0 is fully transparent. Valid values are 0.0, 0.125, 0.25, 0.375,…,1.0.

Thresholding clips parts of the current surface based on the upper and lower bounds for the current threshold function. Any area above the Max or below the Min is eliminated from the display. For example, a surface may be clipped based on a range of X, Y, Z, pressure, temperature, etc.
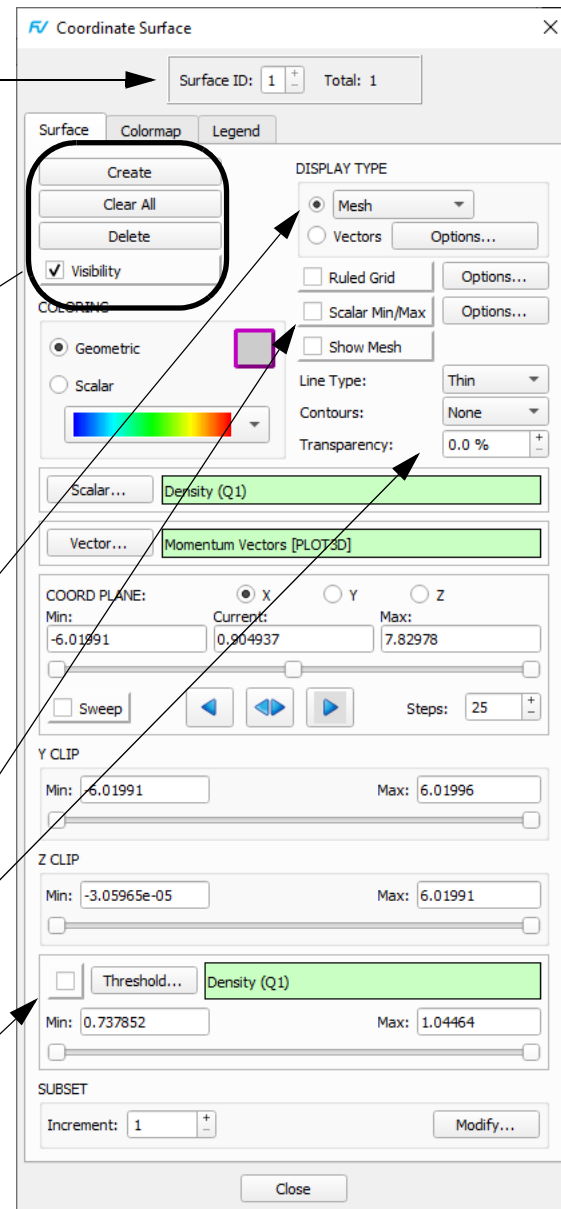
Figure 73 Basic Panel Operations (1)

The Coordinate Surface panel is described above. Panels for other object types (eg. Streamlines, Iso Surfaces, etc.) offer different sets of DISPLAY TYPES among other differences, but are arranged similarly:

The Scalar, Vector and Threshold functions are directly accessible on the Surface or Rake tab of all Visualization panels. The Iso Function is also directly accessible on the Iso-Surface panel.

Surface/Rake visibility can be toggled on and off here

Brings up the Show Min/Max sub-panel. For documentation on this sub-panel, see **Vector Options Sub-panel**.

Brings up the Vector Options sub-panel, which allows for scaling of the vectors, coloring them by a scalar, or showing only one component of the vector. For documentation on this sub-panel, see **Scalar min/max Sub-panel**.

The computational mesh may be shown on top of the current surface by pressing Show Mesh.

Each surface/rake may be colored by the current Geometry color or the current Scalar function, depending upon this selection.

The Sweep controls makes the current surface move in equally spaced steps from its current position to the limits as set by the Min and Max slider positions. The amount that the surface moves in each step is controlled by the Inc: or Steps: field. The Inc: field is the increment at which the possible surface location is sampled (Inc: 2 means show every other position). The Steps: field (if present) indicates the number of equal steps between the Min and Max to take. On the Streamlines & Particle Paths panel, this button is called Animate.

The arrows indicate to Sweep towards Min (<), to bounce (<>) or to sweep towards Max (>). On the Streamlines panel, this button is called Animate. Clicking on the arrow buttons not only changes the direction of the Sweep, but it also moves the position of the current surface 1 step in the desired direction. I.e., pressing the right arrow for the displayed panel will cause the I=16 surface to move to I=17.

Figure 74 Basic Panel Operations (2)

Scalar Colormap Controls

Colormaps for scalar coloring may be selected from the initial Surface/Rake/Group/Feature tab (**Figure 74**) or from the Colormap tab (**Figure 75**), which includes additional scalar coloring options. Both selectors support User defined colormaps in addition to the 8 original and 45 additional predefined colormaps. The scroll bar on the right accesses all options.

On the initial Surface/Rake/Group/Feature tab, the colormaps are aligned horizontally for superior visual appearance. When the mouse hovers over an entry, a "tool tip" (text label) is displayed with the colormap name. The color blue indicates the current selection; the letter "N" highlighted in periwinkle to the right of a colormap indicates a new colormap, i.e., one that has been introduced in this release.

On the Colormap tab, **FieldView** displays an icon in front of each colormap name. New colormap entries are highlighted in periwinkle.
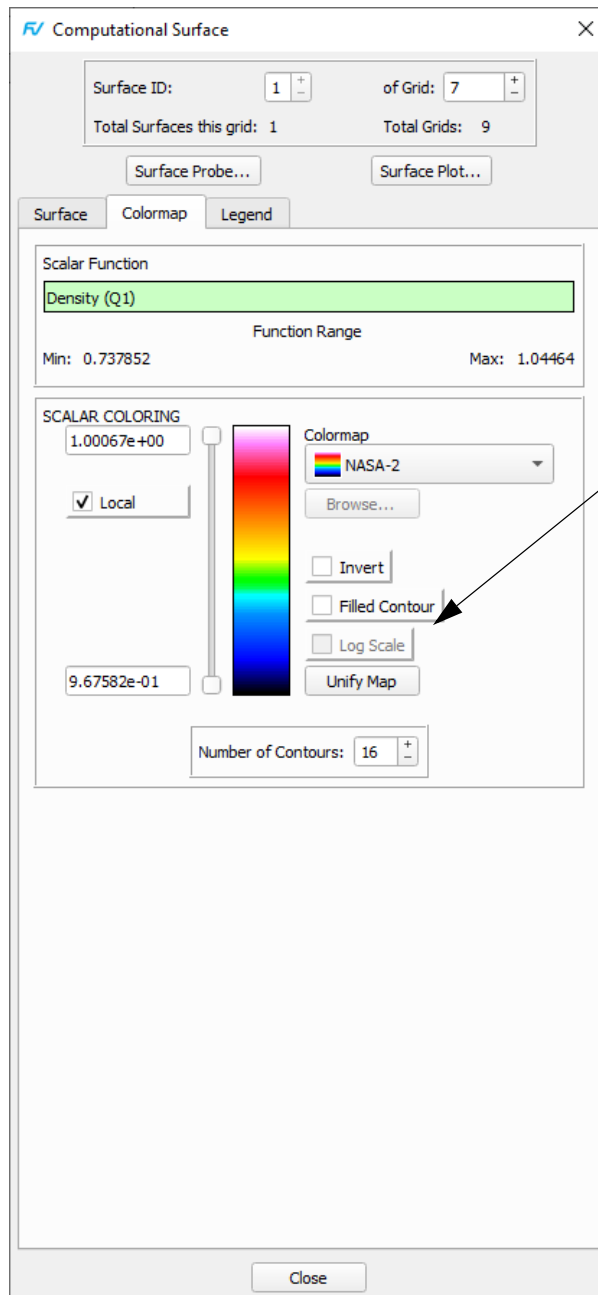
SCALAR COLORING controls appear on the Colormap tab for all surface/rake panels (except Log Scale, which is only supported for surfaces). These controls can be redundantly accessed on the **Scalar Colormap Specification** panel.

Figure 75 The Colormap Tab

Colormaps with a linearly varying lightness, such as Achromatic Vision 1 & 2, inferno, magma, plasma and viridis, present numerous advantages. Thanks to their regular perceived variations in colors, they provide a more accurate representation of scalar fields (see examples in the image below). They're easier to understand for persons affected by

deuteranomaly (also called colorblindness). And they also work when printed in gray scale.
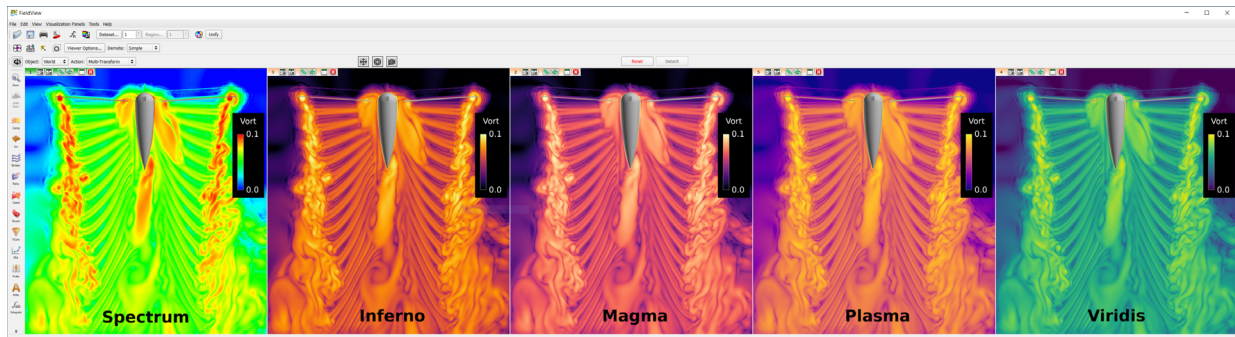
Figure 76 The same Coordinate Surface, colored by the default colormap (Spectrum) and four colormaps with a linearly varying lightness. Note how many more features get revealed in the flow field, especially around the center of the colormap range.

The inferno, magma and plasma colormaps are from Nathaniel J. Smith and Stefan van der Walt. Viridis is from Eric Firing.

User Defined Colormaps

Clicking "User..." on either colormap selector automatically brings up the User-Defined Colormap browser (equivalent to "Browse..." on the Colormap tab). The selection will be inserted directly below "User...", and above the user defined colormaps loaded at **Field-View** startup, if any, and remain unless another browsed selection replaces it or **Field-View** is restarted.

You can also store your colormaps (.col) in user USER directory (the second directory scanned, described below) so that they will be available on the FieldView GUI, alphabetically sorted **with the first will be the default**; otherwise, the default colormap is **Spectrum.** To make a preferred colormap list first (and be your default) you can use clever naming, for example:

```
AAA_my_default_colormap.col
```

Note, however, that Linux and Windows sort differently and can yield a different default colormap on two different desktops owned and operated by the same user, for example:

FieldView Windows:	**FieldView** Linux:
c8hesse.col	Sandy.col
cyan.col	c8hesse.col
fanbiao.col	cyan.col
Sandy.col	fanbiao.col

Note that you can **change the default colormap during your FieldView session**, by using *Edit.. Scalar Colormap Specification* and selecting the desired default map on the resulting Panel. To default to a specific colormap more permanently, you can save a Colormap (.map) Restart and have it load automatically, **See "Preference Restart" on page 12.** of the **Users Guide**.

At startup, **FieldView** will look for user colormaps in two directories. The first directory scanned is

```
<FieldView InstallDir>/data/colormaps
```

On Windows, the default installation directory is

```
C:\Program Files\Tecplot\FieldView 2023
```

On Linux and Mac, the second directory scanned is

```
$HOME/FieldView/colormaps
```
- Under Linux, this is typically `/home/<USER>/FieldView/colormaps`
- Under OS X, this is typically `/Users/<USER>/FieldView/colormaps`

On Windows, the second directory scanned is

```
<USER's Profile Folder>\FieldView\colormaps
```
- Under Windows XP, this is typically
```
C:\Documents and Settings\<USER>\FieldView\colormaps
```
- Under Windows Vista & later, this is typically
```
C:\Users\<USER>\FieldView\colormaps
```

If user defined colormaps are stored in both directories, **FieldView** loads them all, sorted alphabetically. Note, however, that colormap files in the second directory scanned will not be used if their names conflict with files in the first directory scanned, and **FieldView** will print a message to the console or terminal, e.g.:

```
    Skipping Colormap File "/home/jpc/FieldView/colormaps/
c8_hesse.col".
    The colormap name, "c8_hesse", is already in use.
```

Note that if you save a Restart with a user defined colormap in your home directory (the second directory scanned), it will not be available to others who do not have that color-map file. If you attempt to read a Restart with a colormap which is not found, **FieldView** will print the following warning to the console or terminal:

```
    WARNING: User-defined colormap file not found.
    Using default colormap instead.
```

Colormaps will be available site-wide if installed with the **FieldView** installation (the first directory scanned), provided the installation itself is site-wide.

Log Scale for Colormaps
To provide easier analysis of variables with a wide range, **FieldView** can map variable values to colors using a logarithmic scale instead of the standard linear scale. This is especially useful when analyzing derivatives of velocity such as vorticity, helicity, Q-criterion, lambda2, etc. It is compatible with all colormaps (including user-defined colormaps) and works on all surface types.
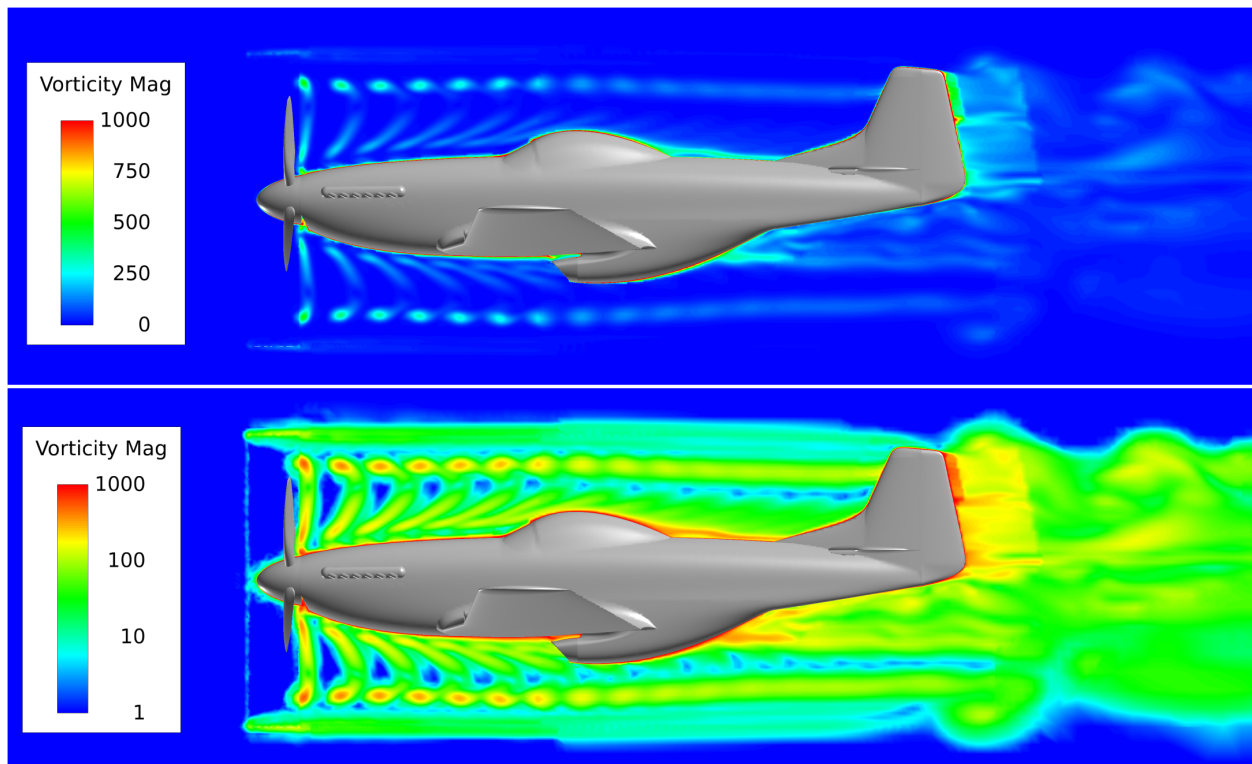
**Figure 77 Comparison of linear (top) and logarithmic (bottom) mapping
of a variable field for the same surface and the same colormap**

The logarithm function is only defined for strictly positive values or negative. Hence, Log Scale will only be turned ON if the current scalar coloring range doesn't include 0. If it does, the error pop-up shown in **Figure 78** will be rendered and Log Scale will not be turned ON.
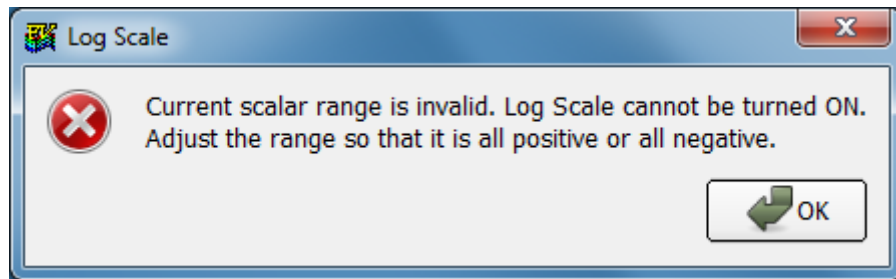
**Figure 78 Error message rendered when an improper scalar interval is used
with Log Scale**

If the scalar coloring range is modified while Log Scale is ON and the new interval
includes 0, the error pop-up shown in **Figure 79** will be rendered and the scalar coloring
interval will be restored.
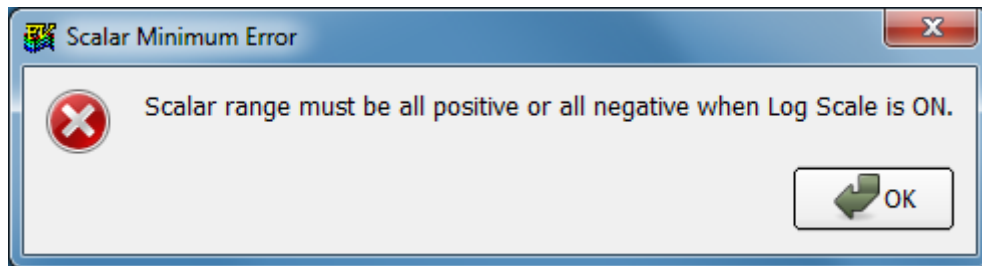
**Figure 79 Error message rendered when Log Scale is ON and the user tries to
set a scalar coloring range that includes 0**

The Log Scale capability has been extended to strictly negative ranges by applying the
same logarithmic mapping but to the opposite of the variable field. **Figure 80** shows the
use of Log Scale on such a strictly negative range.
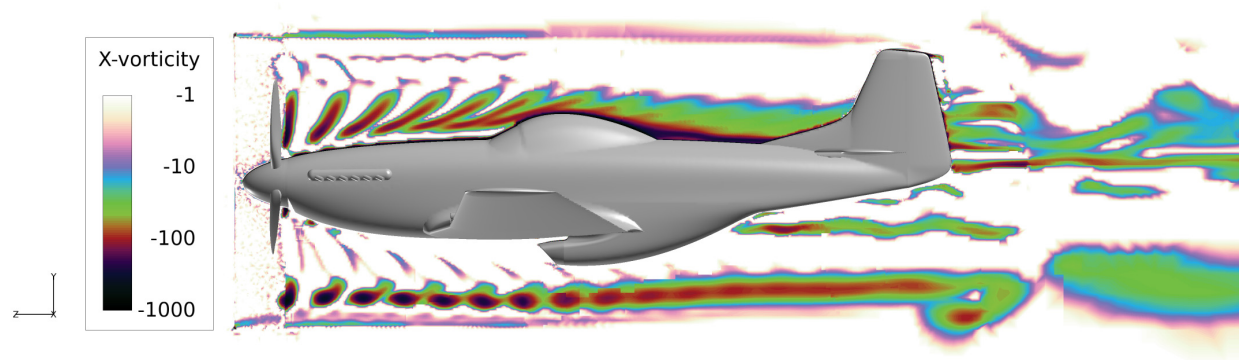
Figure 80 Log Scale applied to a strictly negative range, showing the normal to plane component of vorticity for a range going from -1 to -1,000 s^{-1}

Limitations
Log Scale is currently not supported for objects that are not surfaces (streamlines, streaklines, particle paths, curved vectors, vortex cores and surface flows).

Log Scale is not compatible with automatic local scalar coloring.

Legend Tab Controls

Legend controls appear on the legend tab for all surface/rake panels.
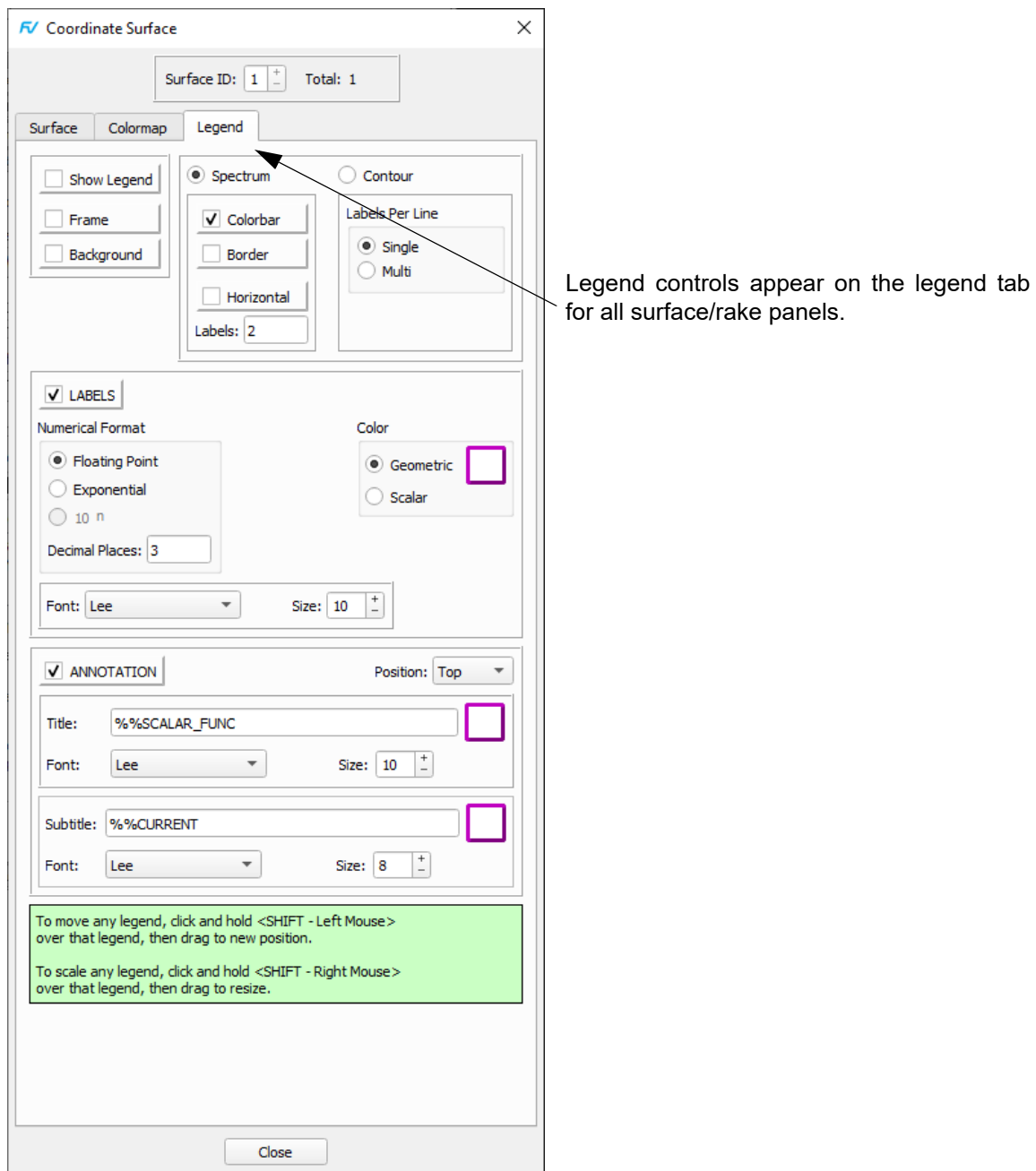
Figure 81 The Legend Tab on Surface/Rake panels

The Legend tab allows you to create Legends and Labels for Spectrums and Contours for scalar-colored objects. Legends have several attributes which can be used to cus-

tomize their final appearance. See also **Legend/Label Control** for redundant access to these features from the Scalar Colormap Specification panel.

Note: Legends always appear in front of any other graphics.

The "Show Legend" button is a toggle to display or hide a legend for the current object. Legends will only display if the object is scalar colored. When the "Frame" button is turned ON, an enclosing frame is drawn around the entire legend. The default for "Frame" is to be turned OFF, that is, no frame is drawn to enclose the legend. When the "Background" button is turned ON, a background fill, using the current background color, is applied to the entire legend area. This means that you cannot "see through" the legend in the graphics window. By default, the "Background" button is OFF which means that you should be able to see objects underneath the legend.

The legend itself can be one of two types: "Spectrum" (the default) or "Contour". The legend type is chosen by making the appropriate selection beneath the tab. A surface or rake can have only one legend type associated with it. When checked, "Colorbar" displays the spectrum colorbar. It is possible to not show the colorbar, and have only the numbers associated with a particular legend. To draw a bounding box around the spectrum colorbar, turn "Border" to ON. If "Colorbar" is not selected, then no border is drawn regardless of the setting of "Border". Legends can be displayed horizontally by pressing the "Horizontal" button. The number of labels shown can also be controlled. The legend range is divided into an appropriate number of equal segments to give the set label number.

If the legend type is "Contour", each contour line gets an alphabetic label to show its exact value. The contour lines either have one label per line, or a series of labels, depending on this selection.

By toggling the "LABELS" button ON or OFF, you can either show or hide the legend numbers. Pressing the color chip to the right of the "LABELS" button brings up a color selector panel. The color you pick is applied to the "Frame" and "Border". If "Color" is set to "Geometric", the color chip selection is also applied to the legend numbers; if "Color" is set to "Scalar", the legend numbers have the color from the spectrum colorbar applied instead. It is also possible to independently control whether numbers are shown in exponential or floating point format, the number of decimal places shown, and the Font type and size for the numbers on the legend.

Powers of 10 numerical format for legends

When Log Scale is used for a surface Colormap, the numerical format "10^n" is available for legend annotations. Instead of annotating the legend with a fixed number of labels including the scalar coloring min/max, only powers of ten which are included in the scalar coloring range are shown, and inputs for number of labels and decimal places are disabled.

The following figure shows the effect of the 10^n numerical format on a legend using Log Scale, in comparison to the default numerical format (decimal).
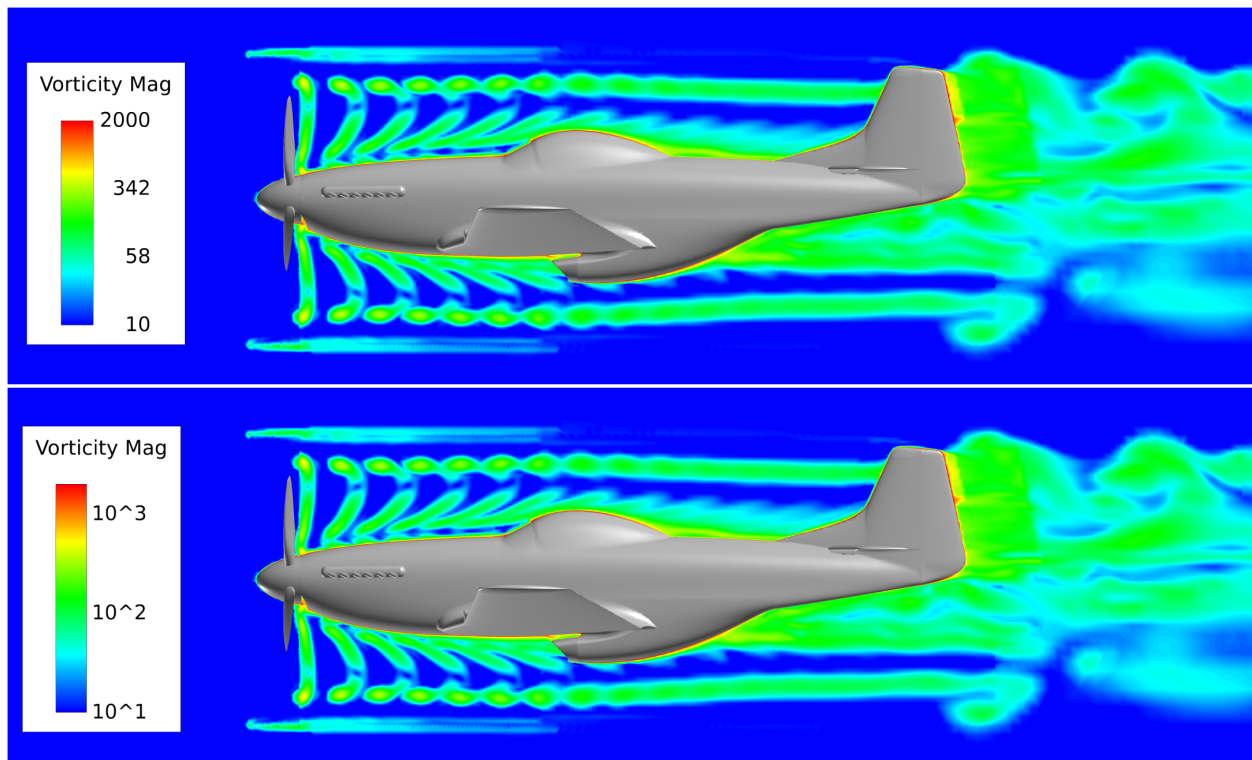
Figure 82 Comparison of the default Floating Point numerical format for legends (top) with the 10^n option (bottom)

It is possible to specify a title and subtitle for each legend using the controls at the bottom of the Legend tab for all surface/rake visualization panels. The "ANNOTATION" button is used to show or hide both the title and subtitle for the legend. When creating new legends, ANNOTATION is ON by default. The "Position" option menu lets you place the title and subtitle at a choice of several fixed locations around the legend: Top, Bottom, Left and Right. When either Left or Right is chosen, the text is rotated 90 degrees, following what is done for 2D Plot left and right axis labels.

By default, the main title has the name of the scalar function for the surface that the legend belongs to. It is possible to independently control both the font, size and color of the title and subtitle. By using the escape sequence, `%%SCALAR_FUNC`, you can change the scalar on the surface, and have your title always correctly reflect the current scalar. It is possible to create an empty title or subtitle. The subtitle field is empty by default.

One useful escape sequence which can be applied to the subtitle (or title) is `%%CUR-RENT`. What this prints on the legend depends on the type of surface which the legend applies to. If it is a coordinate surface, the label prints the location of the coordinate surface slider (ie, either X, Y, Z or R, T, Z or R, T, X depending on whether a region file has been used). If it is an iso-surface, then this label prints the iso-value corresponding to that surface. If it is a computational surface, then the value of either I, J or K for that surface is printed. If the surface gets swept, then the legend title is updated.

The title and subtitle fields also support all the other escape sequences available in **FieldView**, as described in the **Special Numerical Annotation (Escape Sequences)** section of this document.

Scalar min/max Sub-panel

This sub-panel (**Figure 83**) applies when the Show min/max is activated.

Clicking on Options.. will open the Scalar Min/Max Options panel.

Show Min Location and Show Max Location can be activated independently to place markers at the respective locations on the current Surface.

Annotation for your Min and Max locations can be set independently for each location. The default text uses escape sequences for your current scalar function, and its respective value. Numerical formatting escape sequences (to control the formatting of the reported `%%SCALAR_MIN` or `%%SCALAR_MAX` values) can be added, see: Special Numerical Annotation (Escape Sequences)

Font type and size are set here on the panel.
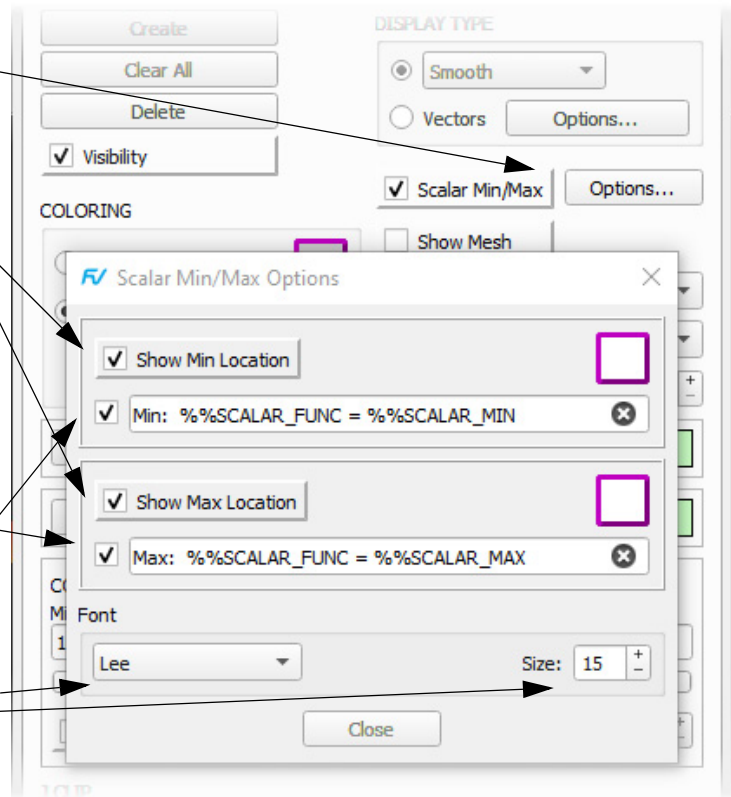
Figure 83 Show min/max Options Panel

Note that for an object with >1 min and/or minimum or maximum values, only a single location will be shown. For example, an X Coordinate surface clipped by Y will actually have min or max values along its edge, but only one location will be shown.

Vector Options Sub-panel

This sub-panel (**Figure 84**) applies when the DISPLAY TYPE is set to Vectors.

The Vector Display Type colors by the current geometry color or by any scalar variable.

Sets the Scale of the Vectors' Length. Increasing this number will make the vectors larger in size.

The Vector Head type may be either 3D or 2D. If 2D, then Presentation Rendering has no effect. When Presentation Rendering is on, Head Scaling will control the thickness of the vector shaft.

The size of the Vector Head is adjustable through the Head Scaling field.

Note: If Head Scaling is used, and the Head Scale is significantly larger than the Vector Scaling value, then no vector head is drawn.

Any single component of the vector may be displayed, or the complete (total) vector may be shown. In addition, the vector may be projected onto an iso-surface or Coordinate Surface.
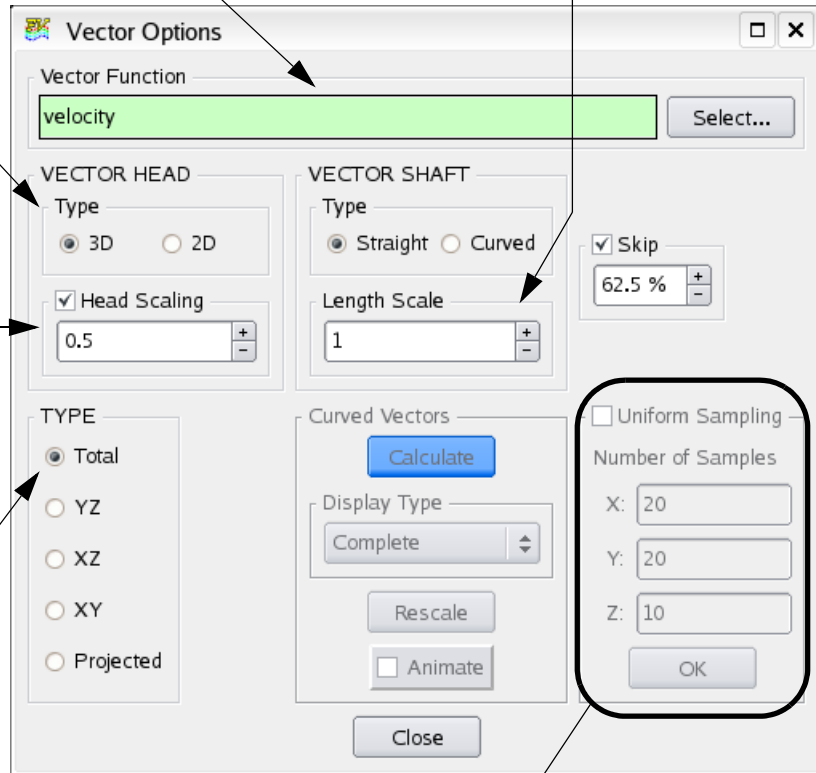
Uniform Sampling is active only for Coordinate Surfaces. For Coordinate Surfaces, vectors are plotted on every polygon created to draw the surface. If Uniform Sampling is turned on, the number of vectors specified in the Number of Samples area are drawn, evenly spaced across the surface. This can be used to reduce the amount of vectors displayed for easier viewing. A different number of vectors may be specified in the X, Y and Z dimensions.

Figure 84 Vector Options Panel

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ fields of the Uniform Sampling section become RTZ (Radius, Theta, Z) fields. For a complete description of the effect of this coordinate change, see **Chapter 3** of the **Reference Manual**.

When the Vector Shaft type is set to Straight (default), the vector length and the Vector Head size are proportional to the magnitude of the Vector Function, shown at the top of

the Vector Options panel. Straight vectors can be uniformly scaled by increasing or decreasing the Length Scale. Vector heads can be set to a uniform size and scaled using the Head Scaling option, as illustrated in **Figure 84**.

Skip factor and Uniform Sampling are mutually exclusive; if one of these options is turned on, the other cannot be used at the same time. The extent of skipping is specified as a percentage, for 9 discrete levels: 0.0%, 12.5%, 25.0%, 37.5%, 50.0%, 62.5%, 75.0%, 87.5% and 100%. A level of 0.0%, the default setting for this option, means that all vectors are drawn. A level of 100% means that all the vectors are skipped; no vectors will be drawn. Skipping can be applied to all surface types when the Vectors Display Type is chosen. Skip settings are independent of all other vector options. For example, it is possible to select a 2D Vector Head type with Head Scaling enabled and vectors projected into the plane, and apply a Skip factor as desired.
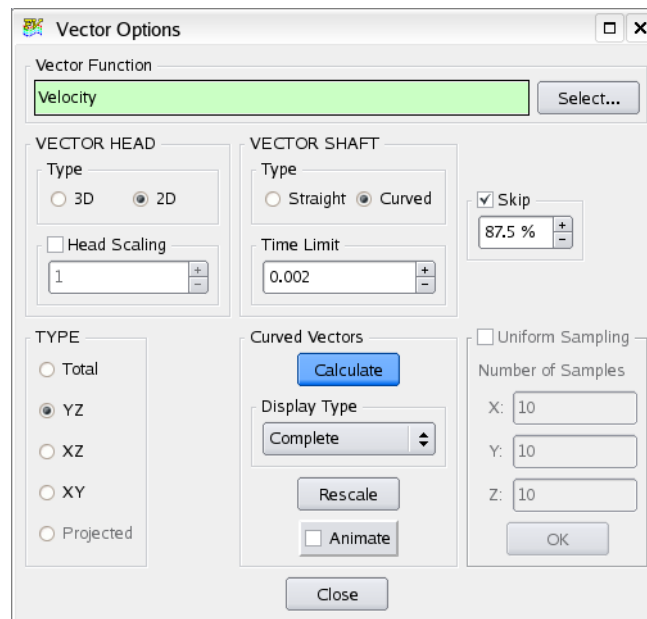
Figure 85 Vector Options for Skip and Curved Vectors

To show curved vectors, change the Vector Shaft type from Straight to Curved. Curved vectors are time limited streamlines with vector attributes. As with streamlines, it is necessary to perform a calculate action in order to generate the curved vectors. The first time that curved vectors are calculated, an estimate for the Time Limit is provided. This estimate is derived from an examination of the global vector components and the extents of the dataset, which can vary significantly from case to case. Since this calculation can be time-intensive, particularly for larger datasets, a conservative, short Time Limit is given as an initial estimate. To increase the 'length' of a curved vector, increase the Time Limit. Although curved vectors are calculated using a common Time Limit, they will have different lengths depending on local variations in the flow field. Vector Head size for curved vectors is proportional to their individual length.

Curved vectors can be shown using one of three display types: Complete, Filament or Growing. The Filament and Growing display types can be animated and saved out as flipbooks.

In **Figure 86** below, we can improve the detail for flow around a cylinder by plotting fewer vectors using the Uniform Sampling vector option.
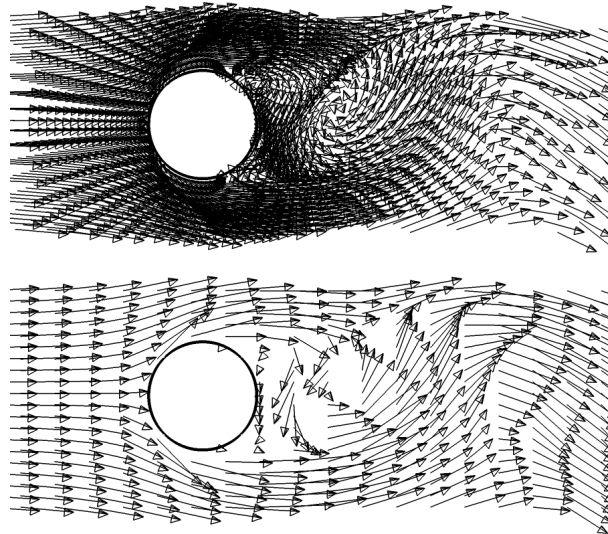
Figure 86 Normal vectors (top), reduced with Uniform Sampling (bottom)
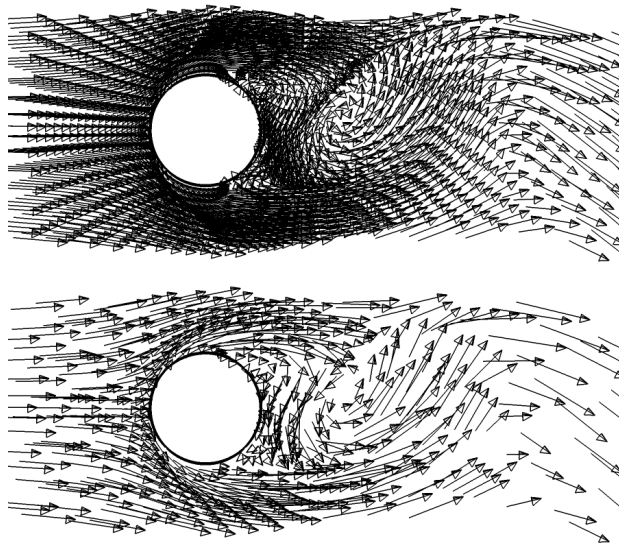
Figure 87 Normal vectors(top), reduced with a Skip factor (bottom)

Uniform Sampling however does not allow you to change the drawn vector density locally, and in this case, the detail in the region behind the cylinder is lost. To address this limitation, the ability to skip vectors is provided as an alternative to uniform sampling.

In **Figure 87** the same region of flow is depicted using a Skip factor.   The local flow variations in the region behind the cylinder where the mesh density is higher are now captured more clearly.

The depiction of the flow downstream from the cylinder is still not ideal. Local variations in the direction of flow are not shown well with straight vectors. In **Figure 88** below, the straight vectors have been replaced with curved vectors, using the same skip factor shown in **Figure 87**.
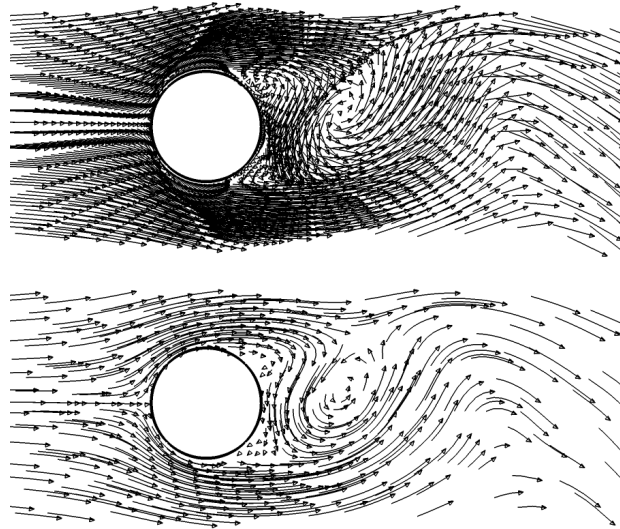
Figure 88 Normal vectors(top), Curved Vectors with a Skip factor (bottom)

Important notes and limitations

Curved vectors are currently available for Coordinate surfaces only.

If the specified Time Limit is too short, curved vectors will not be drawn. In **Figure 89**, the top image shows curved vectors for one time limit, and below, curved vectors for a much shorter time limit.
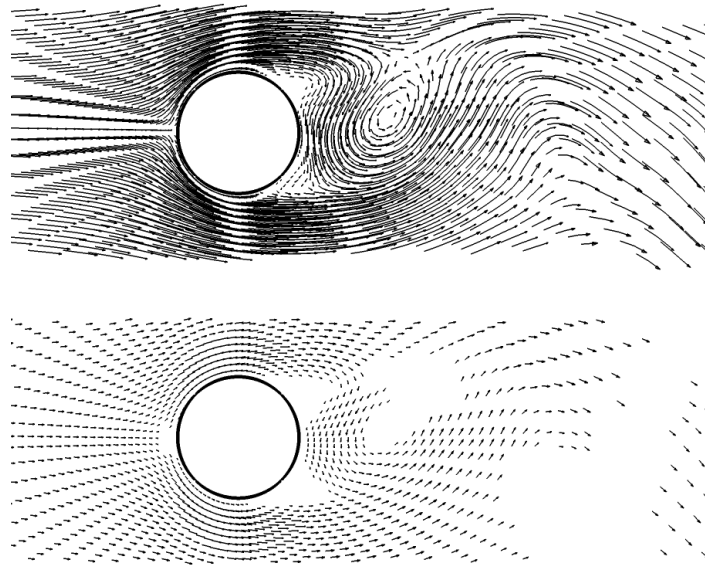
Figure 89 Effect of Time Limit on Curved Vectors

Subset Parameters Sub-panel

This sub-panel (**Figure 90**) is accessed by pressing the Modify… button in the SUBSET section on either the Coordinate or Iso-Surface panel and is used to specify explicit subset ranges in order to reduce the volume of data that is considered for surface calculation. The Region section provides the ability to subset out specific Regions and Blade Rows. Sliders provide for the control of Min: and Max: for each of I, J, or K for any structured dataset in memory. For multi-grid data, the Grid Selection area displays the Total Grids present in the system and specifies the Grid number whose subset information is shown on this sub-panel.

Note: These controls apply only to structured datasets and are inaccessible for unstructured datasets.

By toggling the Grid On/Off button the entire grid of the current dataset may be shut off for the purpose of calculating the current surface by selecting the off button. All grids are on by default.

This section shows the region numbers for the current dataset. The region name as specified in the *.fvreg file is also shown. Deselecting any region (so that it is not highlighted) and pressing OK causes the deselected region (as well as any grid the region consists of) to be subsetted away.

To change the grid whose subset information is displayed, use Grid Selection to either click on the arrow keys or type into the text box Only the grid count for the current dataset is shown.

For each grid, the I, J and K dimensions are displayed here. By adjusting the Increment fields, the number of points used in each direction to create the surface can be decreased. For example, an increment of 2 will use every other point in the selected direction. Adjusting the Min: and Max: fields will clip the current surface in a particular direction.
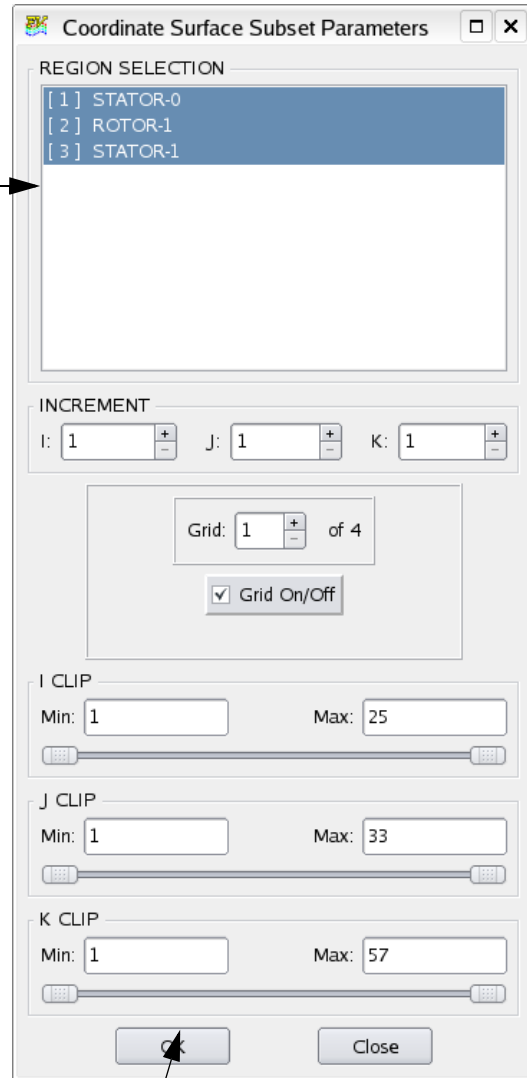
To make any changes on this panel affect the display, press the OK button or exit this panel.

Figure 90 Subset Parameters Panel

Streamlines and Grid/Region Subsetting

Auxiliary Seed Planes consist of a single grid (for structured datasets) but can span several grids for unstructured datasets (X, Y, or Z plane). If an Auxiliary Seed Plane is created for a structured dataset (I, J, or K plane), and the grid (or region) that contains the Auxiliary Seed Plane is subsetted out, then a plane is drawn, and seeds can be placed

on the plane, but no streamlines are calculated, since the grid (region) has been subsetted off. This has been done to avoid the confusion that might result in the creation of an Auxiliary Seed Plane, but no plane being shown due to subsetting.

If selected grids (or region) of a multi-grid unstructured data are subsetted off, then any Auxiliary Seed Plane containing those grids are turned off. If all of the grids that an Auxiliary Seed Plane contains are turned off, then no plane is drawn.

Tools

The Tools menu contains features which allow the user to Export surface and object information, Unify the display characteristics of all surfaces of a given type, create **FieldView** Structured Boundary Files based on the grid file read in, define custom colors with the Color Mixer, open and execute **FieldView** Extension language programs, convert **FieldView** script files to **FieldView** Extension language files, use preset graphics window sizes for images and animations, and dynamically clip datasets.
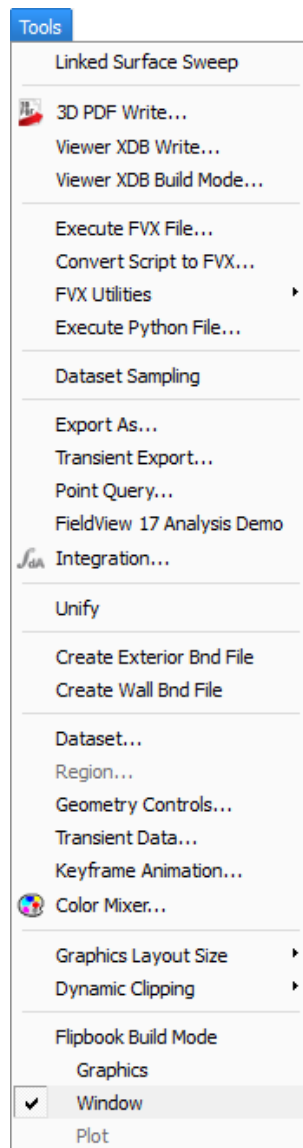
Figure 91 Tools Menu

Linked Surface Sweep

This menu option makes creating animations of multiple surface sweeps easier. With this mode turned on, the sweep controls for the current surface are applied to all like sur-faces. For instance, if you have several coordinate surfaces created for two (or more) datasets, you can use the sweep controls panel for one of the coordinate surfaces to sweep all coordinate surfaces simultaneously. The Flipbook build mode is automatically supported to let you create animations of multiple surfaces with a single button click. More information on how to use the Linked Surface Sweep feature is provided in detail in **Chapter 14**.

3D PDF Write

This menu option is used to export the current window to a 3D PDF format file. For more details, see **3D PDF Write** in **Chapter 14**.

Viewer XDB Write

This menu option is used to create a `.xdb` extract database from **FieldView**. This file can subsequently be read in using **XDBview**, which is freely available from **Tecplot, Inc**. The contents of the `.xdb` file will match what you have created within **FieldView** up to the moment that you decide to write the export. You may control which scalar variables get written to the `.xdb` file interactively or by configuring the `xdb_vars` file. More information on how to create `.xdb` files is provided in detail in **Chapter 14**.

Viewer XDB Build Mode

This command is similar to the previous one in the sense that both are capable of creating a `.xdb` extract file which can be viewed using **XDBview**. The difference here is that this mode enables you to create a `.xdb` extract file based on a sweep of a surface, or a sweep of a dataset through a transient series. More information on how to create `.xdb` files is provided in detail in **Chapter 14**.

Note that the merged transient feature has an impact on the creation of transient `.xdb` files, which are limited to containing only one dataset per file. If the merged transient mode is on, the `.xdb` file created using this build mode contains only the current dataset.

Execute FVX File

This menu option opens a file-selector window with which it is possible to open and run **FVX** programs within **FieldView**. The **FVX** programs are text files with the file name extension `.fvx`. **FVX** programs allow a user to have direct access to **FieldView** features and enable automation of tasks. **FVX** provides more features than the **FieldView** script language. More information on **FVX** is provided in **Chapter 4** of the **Reference Manual**.

Convert Script to FVX

This menu option provides a facility to convert **FieldView** script language-based programs to **FVX** format. This feature allows users to migrate their current **FieldView** scripts to **FVX**.

FVX Utilities

Some **FVX** utilities are provided with **FieldView**. These utilities are placed in the `fvx-_and_restarts` subdirectory of the standard installation. **FVX** utilities are directly accessible from the Tools Menu (see **Figure 92**).
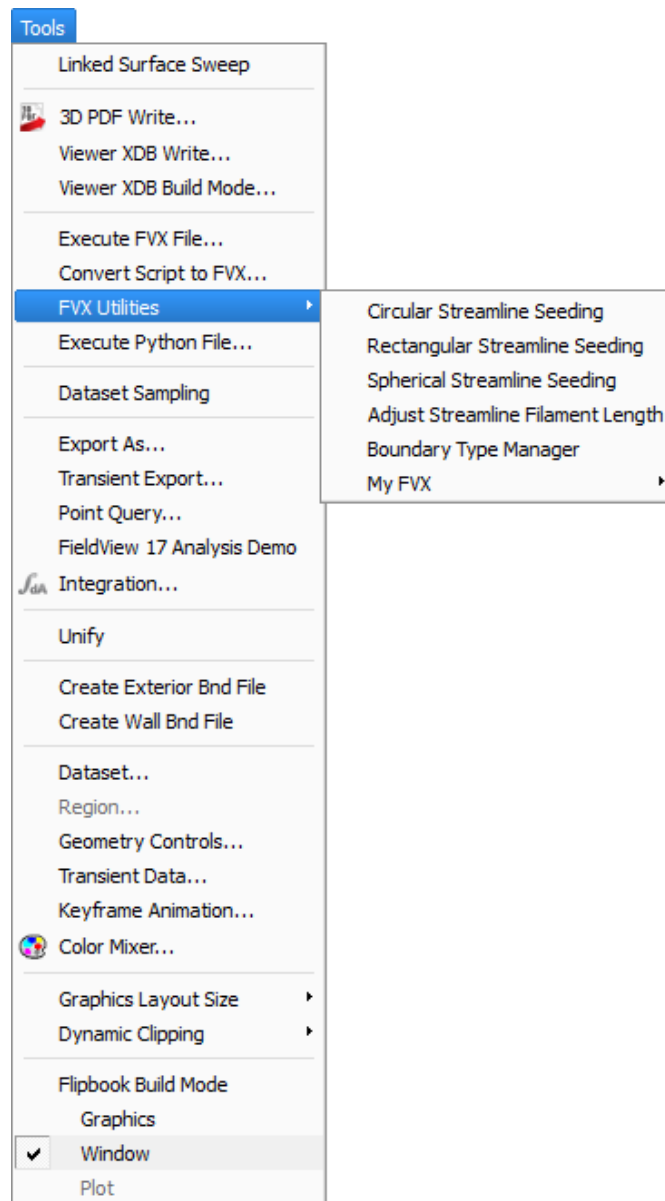
Figure 92 FVX Utilities provided with FieldView

Within this menu, there are references to three tools which are designed to simplify the creation of streamline seeding rakes. A complete tutorial on how these **FVX** programs can be used is provided - see **Chapter 15** of the **Users Guide.** A complete discussion on the 'Adjust Streamline Filament Length' **FVX** program is provided - **See "Length Fraction Controls" on page 195.**

In addition, users can customize an additional menu, accessed thru My **FVX**, to point to their own **FVX** programs. An example of how this might look is illustrated below (see **Figure 93**):

Figure 93 FVX Utilities defined by User

The contents of these menus are controlled using two files, `ReleaseFVX.fvxt` and `UserFVX.fvxt`, both located in the `/fvx_and_restarts` subdirectory. The syntax of both of these files is straightforward. For each menu entry, there must be two lines in the `.fvxt file`. The first entry contains the exact name of how this entry will look on the menu interface. The second line contains the complete path to the location, and name, of the **FVX** file that will be executed when the menu entry is pressed. For example, the `ReleaseFVX.fvxt` file may look something like this:

```
Circular Streamline Seeding
```

```
seed_circ_GUI.fvx

Rectangular Streamline Seeding
seed_rect_GUI.fvx

Spherical Streamline Seeding
seed_sphere_GUI.fvx

Adjust Streamline Filament Length
length_fraction.fvx
```

This feature provides for up to 12 **FVX** utilities associated with the **FieldView** Release, and up to 20 **FVX** utilities associated with My **FVX**.

Execute Python File

This menu option lets you run a Python script within **FieldView**. These scripts are enabled to give you access to all of the existing **FVX** commands. Therefore, it is perhaps useful to consider this feature as an enabled form of **FVX**, which is compatible with the python scripting language. More information on this feature is provided in **Chapter 14**.

Dataset Sampling

The intent of this feature is to give you the ability to compare two completely different datasets, with different grids and different results. The existing Dataset Comparison methodology requires that for formulas to span datasets, their underlying grids must match. To enable this for datasets with different grids, the approach is to create a new 'Results ONLY' or Sampled Dataset. This choice offers the benefit of full interactivity on the sampled dataset, permitting real-time sweeping of surfaces. More details and several examples are presented in **Chapter 14**.

Export

The Export action causes the information from the current surface or rake of the current dataset to output to a file. When the Export button is pressed, the user is presented with a file selector so that an output file name may be given. If the selected filename already exists, the user is given the option of overwriting the file or canceling.

Upon filename selection, the numerical information for the current surface or rake will be output to the file as described below. The variables that are output are the variables currently loaded in the Scalar, Vector, and Threshold function registers. The Iso-surface register will not be output (except if the current panel is the Vortex Cores / Surface Flows panel, in which case the exported file is in the **FieldView** Particle Path format (ASCII)).

More information about Export can be found in **Chapter 14** and the format of the surface or rake exported is given in **Appendix I** of the **Reference Manual**.

Unify

The Unify button is used to make the display style of all surfaces of a particular type (Computational, Iso, Coordinate, etc.) the same for the current dataset. Surfaces/rakes belonging to any other datasets in memory are not affected. The following features are unified: Display Type, Color, Scalar Variable, Scalar min/max (Color map range), Vector display (including vector head type (2D or 3D) and scaling and vector scale factor), Line Thickness, Show Mesh, Transparency and Thresholding. The following items are not affected by Unify: Position/Iso-surface value, Visibility and Subsetting. See **Unify** in Chapter 14 for a additional details about how Unify works.

Create Exterior Bnd File

This tool and the following provide a means to automatically create two different types of structured boundary files. A `.fvbnd` file is generated for a structured dataset loaded in **FieldView** which is current. If an `.fvbnd` file for the current grid file already exists, and the user chooses to overwrite it, the original file is renamed `.fvbnd.old` and the specifications contained in it is merged into the new `.fvbnd` file. Boundaries are renumbered, but the specifications are otherwise be the same. No work is done to assure that the result of merging the generated and existing boundary specifications do not create duplicate boundaries or boundary names. Only the boundaries of the current dataset are created. See **Chapter 14** for more information about the use of this tool and **Appendix H** of the **Reference Manual** for a description of the file format.

Create Wall Bnd File

This tool creates a file named `gridfilename.fvbnd` with a boundary type "walls" for all no slip walls, and boundary types "wall-grid" per grid (if the dataset is single grid the per grid boundary names are omitted). These boundaries specify the exteriors of each grid where velocity is zero. When running interactively, the user is prompted to confirm that velocity magnitude (in whatever form that exists) is currently loaded as the current Threshold function prior to calculating the walls. Only the boundaries of the current dataset are created. See **Chapter 14** for more information about the use of this tool and **Appendix H** of the **Reference Manual** for a description of the file format.

Color Mixer

The Color Mixer allows the user to change 8 of the 10 colors that **FieldView** uses for Geometric and Background colors. Black and White are not editable. The set colors are global over all objects.

The panel initially displays the Geometric colors as both the "Old" and "New" color chips, and its Red/Green/Blue (RGB) value is displayed in the RGB sliders and type-ins. To choose a different color chip to edit, press on the desired color chip to select it. As the RBG values are changed, the "New" color chip and the "Selected Color for Mixing" chip update to reflect the new color. The "Old" color chip continues to display the old color (see panel description on next page).

The user may change more than one color at a time, before applying the changes. Pressing the "Apply" button commits all color changes made to any chip. The colors committed would be those displayed in the color chip selector in the upper portion of the panel.   See **Chapter 14** for more information concerning the use of the Color Mixer panel.

Graphics Layout Size

Please refer to **Graphics Layout Size** in **Chapter 14** for information.

Help

The Help menu (see **Figure 94**) contains an entry entitled "**FieldView** Customer Center". When selected, the default web browser will be launched and will navigate to the Customer Center login page. This section of our website is accessible to all registered **FieldView** users. It contains **FieldView** downloads, tutorials and over 150 FAQs.
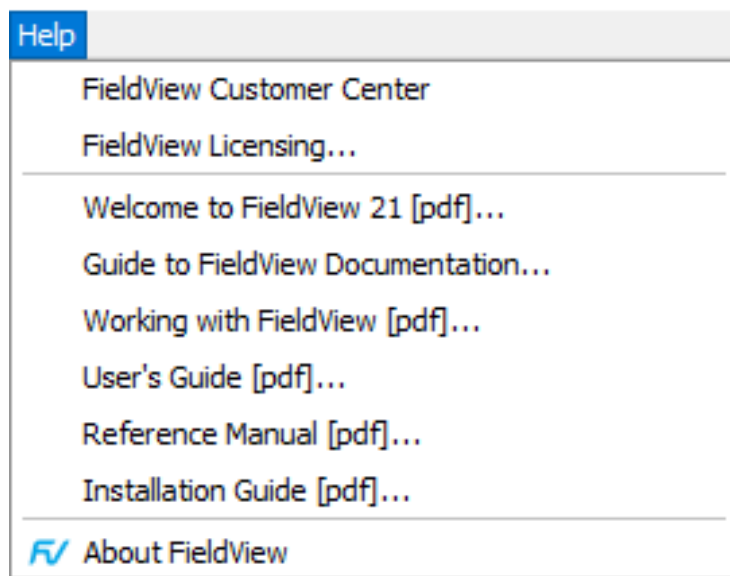
Figure 94 Help Menu

Online Help Menu Customization

For users with no Internet access or who maintain a **FieldView** documentation page on their internal network, the Help menu can be customized to include a link to a user-defined URL. This is done by setting the following environment variable as a text string:

```
FV_USER_HELP_URL
```

By default, the new menu entry will use the first 33 characters of the provided URL. To specify different text for the menu entry label, also set the following environment variable:

```
FV_USER_HELP_ENTRY
```

Figure 95 illustrates an example in which both environment variables have been set.
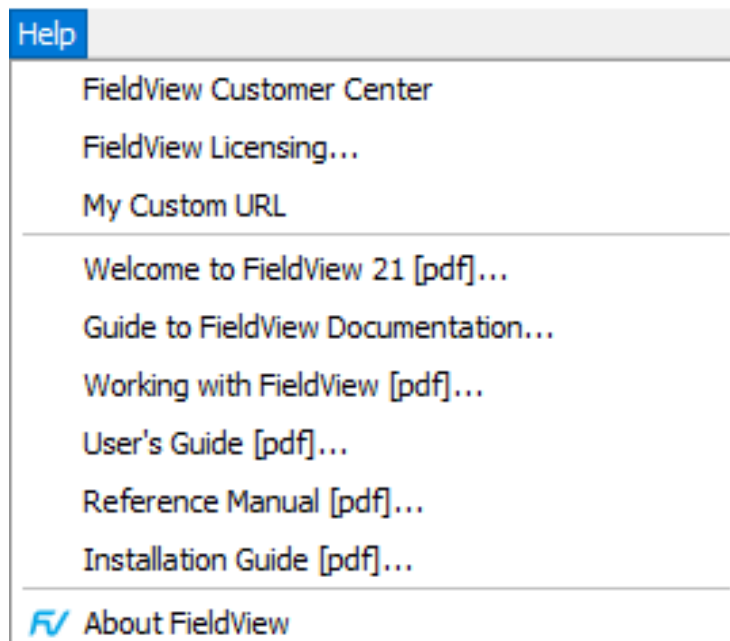
Figure 95 Help Menu with Customized Entry

Limitations
FieldView will not attempt to confirm the validity of `FV_USER_HELP_URL`.

`FV_USER_HELP_ENTRY` will have no effect unless `FV_USER_HELP_URL` is also set.

The number of characters displayed in a Help menu entry will be truncated at 33.

Viewing PDF Help Documents

The Help menu gives you direct access to a collection of PDF document files located in the `/help` directory of the **FieldView** installation, including **Welcome to FieldView 2023**, **Working with FieldView**, **User's Guide**, **Reference Manual**, and **Installation Guide**. The "Guide to **FieldView** Documentation" entry is a main PDF file with descriptions of and links to these documents.

Also on the Help menu is the About **FieldView** entry which provides copyright information.

The **FieldView** documentation is composed of several PDF files, spanning the topics of installation, general reference (including scripting and **FVX** programming), and basic user guidance (with several tutorials and complete descriptions of all panel operations).

On WINDOWS systems, Adobe® Acrobat® is widely used, and **FieldView** is designed to work with this PDF Viewer only. On LINUX systems however, there are several PDF Viewer options. For our supported LINUX platforms, when you select one of the PDF files from the Help pulldown menu, **FieldView** will automatically search for PDF viewers in the following order:

1. Adobe® Acrobat® (or acroread)

2. Evince

3. Kpdf

4. Xpdf

Once a PDF viewer has been successfully located, **FieldView** will launch the selected PDF Help document. Note that your PATH environment must be set so that at least one of these PDF Viewer applications can be located.

It is also possible to configure your own PDF Viewer to read the PDF documentation. This can be done by setting the following environment variable to point to the full path and name of your PDF Viewer. For example:

```
FV_PDF_VIEWER=/opt/kde3/bin/kpdf
```

If the `FV_PDF_VIEWER` environment variable is set, **FieldView** will try to launch this application first. If this fails, it will try to find a PDF Viewer following the order noted above. If **FieldView** fails to find a PDF Viewer, you will see the following error message:
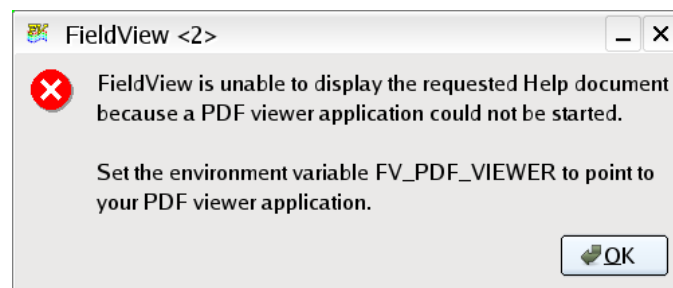
Figure 96 PDF Viewer Warning Message

Modifications to the documentation source will let you view PDF files using PDF viewers such as Evince which do not work with secured documents.

Chapter 3

Basic Oper-
ations

3

To make **FieldView** easy to use, the basic layout of all of the visualization panels for sur-
faces and rakes is essentially the same. As a result, many of the text fields/buttons/
menus are repeated on each of the panels. These options are all documented within this
chapter.

Working with File Selectors

File selector panels (or File Browsers) are used whenever the need for specifying an
input or output filename arises. The file selector panels are meant to make this task eas-
ier for the user. A typical file selector panel and its functions are shown in **Figure 97
Example File Selector Panel**. This particular panel is for a Complete Restart, but exhib-
its the features found in all of the file selector panels encountered in **FieldView**.

Note that on Windows, FieldView uses the operating system's default browser, except
when selecting a dataset in Client-Server mode. The file selector shown below is what is
presented on Linux and macOS systems.

Each item on the pull-down menu brings up a sub-panel with the following contents:

This entry box displays the directory where a file will be read from or written to.

Displays the current Files available. File read action can be accomplished by double clicking on file-names or clicking once and using the Open button.

Displays the current Directories available. It also allows the user to traverse up and down the current directory structure.

To specify the file name, type directly into the entry box, followed by a carriage return.

The selected file appears here.

Press Open to read the file or set of files specified above. Reading a restart file that corresponds to a single panel will make the corresponding panel visible (with the exception of the Data File Input panel).

Exits the file selection panel. No action will be taken.
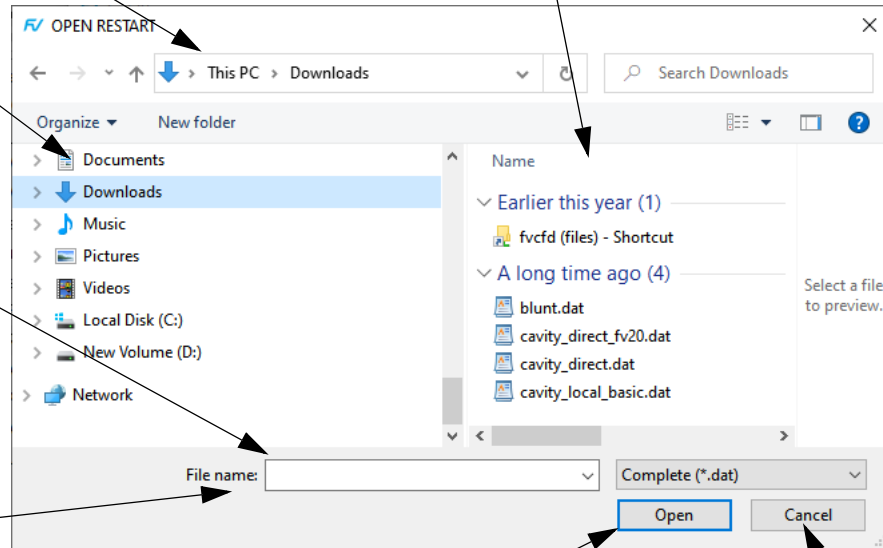
Figure 97 Example File Selector Panel

File browser preferences save and return to the last directory browsed, individually, for all file selectors. This information is stored in `FieldView.ini`. Its location depends on the system where **FieldView** is being run.

On Linux,
`/home/user/.config/Tecplot/FieldView.ini`

On WIndows
`C:\Users\user\AppData\Roaming\Tecplot\FieldView.ini`

We recommend that you *do not* edit this file.

This implementation has introduced a potential operational conflict with server config files. By saving the last directory browsed during a remote data input action, on the next start up of **FieldView**, the remote file browser returns to the last directory instead of the

StartDirectory specified in the server config file. The server current working directory is echoed to the console window along with other server diagnostic information.

A benefit of having file browser directory preferences for LINUX users is that you can start **FieldView** from any point in the console, and return back to where you working the last time you left **FieldView**. This is also true for remote operation, working with **Field-View** servers. Another benefit is that you can store things such as commonly used Formula RESTARTS or colormaps in a standard directory and easily navigate back to them at any time.

If you do not wish to return to previously browsed directories as a preference, you can revert to previous **FieldView** behavior by setting the environment variable,

`FV_OLD_BROWSER_BEHAVIOR`

When this is set, the settings from the [FileDialogs] and [Servers] sections of the `Field-View.ini` file are NOT used. For the local file dialogs, each opens in the current working directory at the time the dialog is opened. For the servers, the remote file browser opens in "StartDirectory", or the home directory of "UserName", if specified in the server config file. Note that even if the environment variable, `FV_OLD_BROWSER_BEHAVIOR` is set, **FieldView** still writes a preference file, saving the location and size of the main window, the state of the toolbars, and the locations of all visualization panels.

In some cases, it may be desirable to turn off the default behavior which saves and restores user preferences, such as the GUI layout and last file locations browsed. This is controlled by setting the following environment variable:

`FV_NO_PREFERENCES`

When this environment variable is set, the **FieldView** preference file (`FieldView.ini`) will neither be read on start up of nor written on exit from **FieldView**.

Setting `FV_NO_PREFERENCES` is similar to setting `FV_OLD_BROWSER_BEHAVIOR`, which suppresses reading file browser settings from `FieldView.ini`. The behavior of file browsers is affected as follows:

1. For local file browsers, when the file dialog is first opened, it will be viewing the current working directory.

2. For remote browsers, the "StartDirectory" field in the server config (`.srv`) file will be used, if it exists; otherwise the remote browser will open in the your home directory.

With `FV_OLD_BROWSER_BEHAVIOR`, however, **FieldView** does still write file browser settings to the `FieldView.ini` file.

The StartDirectory keyword, CWD, can be set in the server config file:

StartDirectory: CWD

Use this (case insensitive) setting to start **FieldView** in your current directory when running in Client-Server or Parallel mode. This is particularly useful if you routinely navigate through many directory levels below your home directory. This setting over-rides the saved preferences behavior in **FieldView** which store and return you to the last directory visited by default. In other words, this setting if present, take precedences over the stored preferences which are saved at the end of every **FieldView** session. Note that within a **FieldView** session, if you navigate away from your current working directory, this is stored and the location becomes the new 'remembered' location upon exit. This server configuration setting only applies to the initial view in your file browser.

Working with Filters (Search Bars)

Because some datasets have extensive lists of items, scrolling and selecting the ones you want to use for a particular visualization can be unwieldy. The search bar included on many **FieldView** panels facilitates selecting from lengthy lists using a text string filter, which also expedites locating multiple entries with similar names.

A search bar consisting of a type-in field and a tool button appear on the panel under the selection list. By default, the type-in field is empty, the entire list is visible, and the tool button displays a magnifying glass icon, as illustrated in **Figure 98**. When text is typed into the field, filtering is applied automatically, with only entries matching the filter string remaining visible, and the tool button displays an "X", as illustrated in **Figure 99**. Pressing the "X" will clear the search and restore the default state.
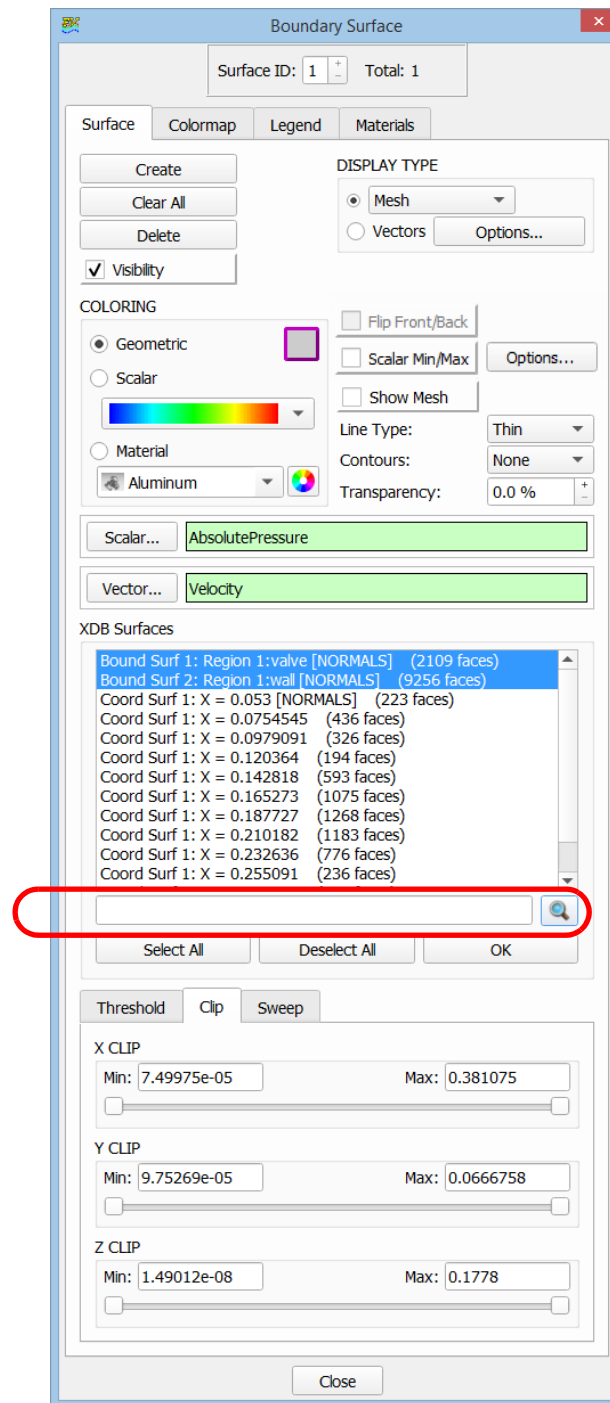
Figure 98 Boundary Surface panel with filter (search bar)
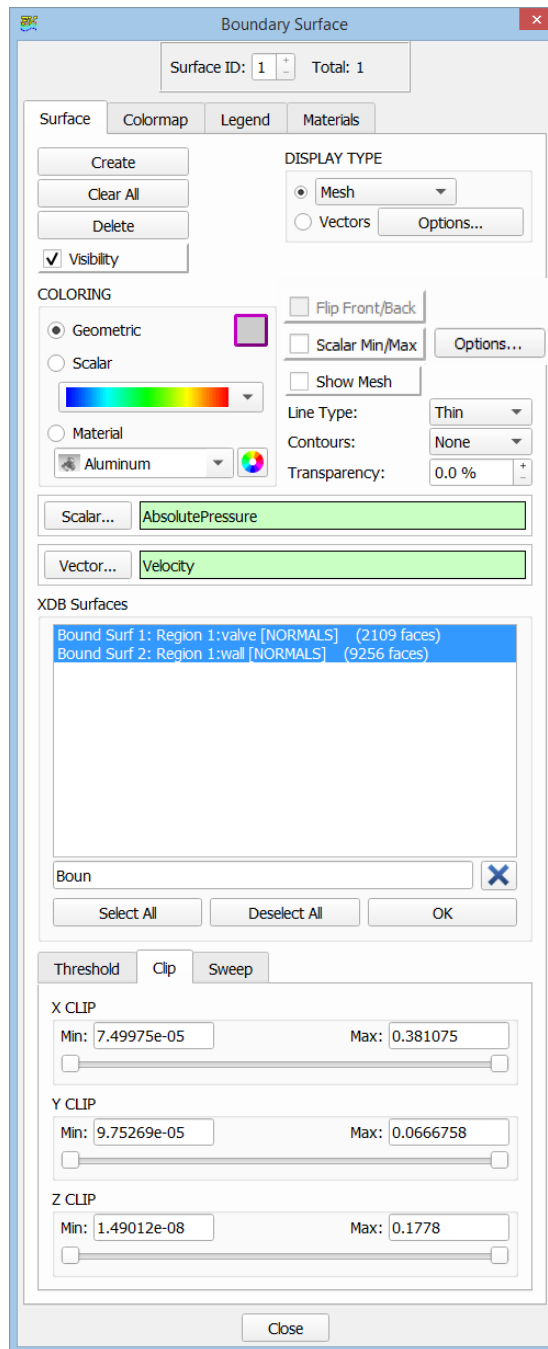
Figure 99 BOUNDARY TYPES list filtered by text string "Boun" after pressing Select All

Filtering is case insensitive and uses simple wildcard matching:

c   Any character represents itself apart from those mentioned below.

Thus c matches the character c.

? Matches any single character.

* Matches zero or more of any characters.

[...] Sets of characters can be represented in square brackets. The filtering will be applied after the brackets are closed, and the contents within the brackets will be treated as a set of characters rather than a character string. For example, if "[Wheel]" is typed, the filter will display names containing the letters "W", "h", "e" and "l" instead of the word "Wheel".

\ The character '\' escapes the wildcard character.

The search will be cleared when:

- the user presses the "Clear Search" button ("X")
- the current surface on the panel changes, including when surfaces are cleared, deleted, or created
- the panel is closed
- the current dataset changes

Interacting with a Filtered List

- A selection will remain selected even if it is no longer visible (i.e., filtered out).
- Select All and Deselect All will act only on the visible entries. The selected state of all non-visible entries is not affected.

Dataset Controls

The Dataset Controls panel (see **Figure 100 Dataset Controls Panel**) can be brought up by clicking the Dataset... button on the Main Toolbar or selecting Dataset... from the Main Menu bar Tools pull-down. The Dataset Controls panel allows you to select a dataset to transform by scaling, rotation and/or translation. This panel also allows for Mirror, Rotate and Translate Duplication of the selected dataset.

Multiple Datasets

FieldView allows for multiple datasets to be read into memory through the Append button on the data input panels. When this is done, every function, surface, rake, groups of particle paths, 2D plots and/or Vortex Cores / Surface Flows in **FieldView** are attached to a particular dataset. The properties of these objects can only be changed if the dataset that they are attached to is the current dataset. In order to adjust the display parameters

of any object in **FieldView**, the dataset which that object is attached to must become the current dataset. This can be accomplished in any of the following ways:

1. On the Dataset Controls panel (see **Figure 100 Dataset Controls Panel**), type into the Dataset: field or click + or - to change the dataset number.
2. On the Main Toolbar, type into the Dataset: field or click + or - to change the dataset number.
3. "Quick-pick" on the object.

If the object (rake, surface, etc.) belongs to a different dataset than the current dataset, then the current dataset will change. Usually, this change will not be very noticeable to you. The dataset number on the Main Toolbar will update and the picked object's attributes will be displayed in the appropriate panel.

Dataset Sweeping

In addition, you can use the Sweep button to toggle the visibility of the datasets that you have loaded into **FieldView** memory. If you have several datasets visible in the graphics window, turning on the Sweep button will first cause all of the datasets to become invisible (visibility off). Then, starting with the current dataset (controllable through the dataset number field on the Dataset Controls panel), the datasets will, in turn, have their visibility turned on and off. That is, if you have 4 datasets in memory and the current dataset is #2, then turning on the Sweep button will cause all the datasets to turn off, then dataset #2 will have its visibility turned on, then off, then dataset #3 will have its visibility turned on, then off, etc. The sweep is continuous, so this visibility toggling will occur until the Sweep button is turned off.

This area displays the current dataset and the number of grids it contains. To change the current dataset, either type into the text field or click + or -.

The current dataset may be turned invisible in entirety by turning off the visibility button.

Dataset Sweep will cycle appended dataset's visibility. During the dataset sweep, only the dataset number on this panel will be updated. The other controls on this panel (and the Dataset number on the Viewer Toolbar) will not update during the sweep. When the sweep is stopped, the entire panel will be updated to display all the information for the current dataset at the time the sweep ended. The dataset number on Viewer Toolbar will also be updated. This button will change to *Build*, when you enable Flipbook.

These display-only fields list the names of the files used as input for the current dataset.

By default, there is no **Duplication.** The current dataset may be either mirrored across one or more axis, or copied and rotated about an axis, or copied linearly, in any or all of three axis. Selecting Mirror Rotate or Translate, the appropriate parameters panel will be displayed to make your settings, and will provide defaults based on your dataset. The corresponding panels are shown in **Figure 101 Dataset Controls Panel - Duplication Modes on** the next page of this documentation.

Any of the dimensions of the current dataset can be **Scaled**. This affects how the data appears visually only.

Translate allows for exact translation along one or more axes through the type-in fields.

Rotate enables up to 3 rotations in arbitrary axis order to be applied to a given dataset. There are 3 pairs of specifications fields. Each pair consists of an option menu of the rotation axis selection (X, Y, or Z), and a type in field for an angle in degrees. The rotations can be "ganged" by rotating about a single axis 2 or 3 times. The axes marker will change to reflect the rotation.

This feature allows more precise control over rotations than allowed through mouse interactions. Note: These settings are not reset by the Reset button on the Viewer Toolbar. They may, however, be reset by clicking the Defaults button.

Figure 100 Dataset Controls Panel

Mirror - Toggle on one or more of these buttons to **mirror** about the X, Y and/or Z axes. Note: Selecting 2 axes will create 3 copies, while selecting all 3 axes will create 7 additional copies.

Rotate - Select one of these buttons to rotate about the X, Y or Z axes.

Enter the total number of copies, and total sweep (degrees) to sweep the copies. For example, two copies with a total sweep of 360 degrees will produce one instance every 180 degrees.

Translate - Translate allows for exact translation along one or more axes through the type-in fields. The *Delta* fields will default to values dedrived from your CFD case, with *Copies* defaulting to 2.

FieldView's DUPLICATION options allow you to instance smaller periodic solutions to represent a complete system. The *Translate* duplication mode, in particular, will obtain default offsets based on the extents of your data in Cartesian space. For Rotate and Translate duplication modes, streamlines will integrate across your instances automatically, given that the settings used are appropriate fo the given CFD case. Pressing the *None, Mirror, Rotate* or *Translate* button in the left column, will immediately execute the operation you choose, using defaults, updating the panel with the appropriate settings for the action chosen.

Figure 101 Dataset Controls Panel - Duplication Modes

Dataset Scaling

Dataset Scaling can be used to make the geometry easier to visualize. It can be used to stretch or shrink the dataset in any of the three geometric directions. Geometries can be vertically stretched for better visualization of boundary layer details. Channel/pipe flows with very large aspect ratios can be shrunk for larger images in the graphics window, etc. Note: Any such dataset scaling that takes place affects the *visual* look of the data in the graphics window. The actual dimensions of the data are not unchanged. Data extraction functions, such as Point Probe, Integration, etc., will yield the same result whether the data has been scaled or not.

Axisymmetric Visualizations

If you have axisymmetric data (with a data domain containing a symmetry plane mirrored around X, Y or Z), you can create a visualization consisting of a surface (Boundary, Computational, Iso or Coordinate) which is mirrored on both sides of the symmetry plane. If you wish to display different scalars, on each side of a surface mirrored across a symmetry plane, dataset scaling will be needed to perform this visualization. To do this, use a negative number for the dataset scale of the second dataset. This will cause the view of the second dataset to be reflected about the symmetry plane. For example, suppose your data is symmetric about the Z-axis. Read in the dataset a second time and change the value for the Z Scale to -1. This will produce a view similar to Z-Axis mirroring. However, since you now have two datasets in memory, your can visualize them separately. You can create an X=5 Coordinate surface on the first dataset (right-hand side) scalar colored with pressure, and a X=5 Coordinate surface on the second dataset (now the left-hand side) scalar colored with temperature. Important Note: To get the appearance of the model appearing on both sides of the symmetry plane, the data must be read-in twice (with the Append feature) - one dataset has positive scale, the other is negative.

Region Controls

Regions are sub-volumes of datasets. Regions are a hierarchical level in **FieldView** below Dataset but above grids, surfaces, and rakes. The Region definition can be used to group grids, change the coordinate system available for the dataset in **FieldView**, and change characteristics of the dataset coordinates. In addition, the file is used to pass blade row information for turbomachinery applications.

The Region Controls panel (see **Figure 102 Region Controls Panel**) can be brought up by clicking the Region... button on the Main Toolbar or selecting Region... from the Main Menu bar Tools pulldown. The Region Controls panel allows you to select a Region similar to the way that datasets can be modified with the Dataset Controls panel. This item will only be available if a valid FVREG file has been defined for the current dataset.

Note: There is no requirement that regions be used. If no region information is present, then the region fields are grayed out and inaccessible.

The top section of the *Region Controls* panel shows the Total Regions for the current dataset as well as indicate the current region number. The region name (as specified in the (*.fvreg) file is also shown. The Select… button allows you to select a specific non-empty region by name (see **Chapter 3** of the **Reference Manual**). Visibility can also be controlled on a region by region basis.

The body of the *Region Controls* panel is nearly identical to the *Dataset Controls* panel, shown above. It provides Region attributes in exactly the same way as the Dataset Controls panel, with the exception of *Rotate* parameters and its defaults. For more information on specifying rotational parameters for Regions, please see **Chapter 3** *Region Files* in the Reference Manual for a complete description on the usage of this panel to duplicate regions.

Figure 102 Region Controls Panel

Cylindrical / Cartesian Coordinates:

The Region file can be used to convert XYZ data to RTZ (Radius, Theta, Z) data in **FieldView**. This affects many of the panels and fields in **FieldView** and is described in detail below (as well as mentioned in the specific panel chapters to follow). Note that the input grid(s) values must be Cartesian. **FieldView** converts this Cartesian data into RTZ coordinates upon read-in. **FieldView** does not recognize non-Cartesian grid data (some

readers may already convert cylindrical coordinates to Cartesian for **FieldView**. Note: any data format supports `.fvreg` files.).

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ sliders on the Coordinate Surface panel become RTZ (Radius, Theta, Z) sliders. For a complete description of the effect of this coordinate change, see **Chapter 3** of the **Reference Manual**.

Selecting and Creating Functions

This panel (**Figure 103**) is used to define which variables are to be used when creating surfaces, as well as to define any new variables. If you press any of the Iso-Surface, Scalar, Vector or Threshold buttons, the Function Selection panel is presented. The Function Selection panel is used to pick the variable to load into the selected register. See the section **Working with Filters (Search Bars)** for information on how to use the search bar on the Function Selection panel to filter long lists of function names. Note that for all registers other than Vector, only scalar variables are displayed. For the Vector register, only vector variables are displayed.

The Scalar, Vector and Threshold functions are directly accessible on all Visualization panels. The Iso Function is also directly accessible on the Iso-surface panel.

The Create and Edit buttons are used to define new or modify existing variables, using the Formula Specification Panel. This panel, also called the "CFD Calculator", is shown below. The new variables may be defined from any of the variables currently in memory, constants, and any of a series of pre-defined math functions. See **Chapter 2** and **Appendix A Built-In Functions** of the **Reference Manual** for more information.

Iso-Surface... loads a scalar into the iso-surface register. Iso-surfaces are surfaces of constant value, such as regions of constant temperature and pressure.

Threshold... loads a scalar into the Threshold Register. This function is used to limit the boundaries of surfaces to areas within a range of values for a quantity, such as temperature or pressure.

Scalar... loads a scalar quantity, such as temperature or pressure into the scalar register.

Vector... loads a vector function into the vector register. Vector functions are used for streamline integration.

Create... brings up the Function Formula Specification panel (**Figure 104 Function Formula Specification Panel**). This panel is used to create a new function based on any previously defined functions, constants, functions read from an external file, or predefined math functions.

Edit... edits a formula that has been previously defined with the Function Formula Specification panel.

Delete is used to delete a formula. Note that a formula may not be deleted if it is "in-use" which means that it is currently loaded into a register, is being used to define one or more surfaces, or is used in another formula.

Figure 103 Function Specification Panel

The Constants section contains values that **FieldView** obtains from certain solver results files. For some solvers, the number of constants available may be fewer or greater than the number shown here.

The Q-Criterion function is based on the general form of the criterion proposed by Hunt in 1988. The function is generalized such that no assumption is made on compressibility.

The Operations and Keys are the built-in functions supported by **FieldView**. Any formula that can be expressed in terms of these operations and solver variables can be created and saved.

When any of the Constants, Operations, or Quantities are selected, the Formula created is displayed in this area. This area may also be edited so that functions can be typed in directly.

Lists the Name of the function that is being edited. It will be blank at other times.

Lambda2 (λ_2) criterion is based on the observation that, in regions where lambda2 is less than zero, rotation exceeds strain, and in conjunction with a pressure minimum, indicates the presence of a vortex.

This toggle allows switching Dataset Comparison mode on and off.

Quantities lists all of the functions that are valid for the current dataset, including all of the geometric grid-based functions.

Figure 104 Function Formula Specification Panel

Scalar Colormap Specification

The Scalar Colormap Specification panel (**Figure 105**) can be selected from the icon on the Main Toolbar or the Edit menu. The panel is used to set a color spectrum for the current surface or rake, scalar function minimum and maximum values for colormap interpretation, and the specific colormap selection (see also for redundant access to these features from all visualization panels). The Scalar Colormap Specification panel can also be used to specify the graphics screen background and geometry color. Legend/Label Control allows you to create and modify Legends and Labels for Spectrums and Contours.

The number of contour lines to be displayed can be changed by editing the Number of Contours or using the arrow keys. There is a limit of 500 contours.

The Minimum and Maximum scalar values are displayed here. These values map to the bottom and top of the colormap. To change the values, edit the type-in fields or move the slider bars.

This area displays the current Colormap that is being used for mapping scalar function values.

The type of colormap to be used is selected here. Options include Spectrum, NASA-1, NASA-2, Gray Scale, Color Striped, Black & White, Striped, Zebra and others. User-defined colormaps may also be input. The colormap may be inverted by pressing the Invert button.

For information, see **Log Scale for Color-maps**.

The Legend/Label Control button will bring up the Legend sub-panel.

The color of any Geometric object, or of the Background is controlled with these options. Clicking on the color chip brings up a menu of possible color choices.

Local automatically sets the Local Min Max of the scalar values that are displayed on the selected surface/rake.

The Filled Contour button, when on, controls the way that surfaces are colored. Vertex coloring is replaced with a constant color per polygon of the surface. If the contour value changes within a polygon, that polygon will be split into two or more polygons resulting in individually colored polygons for as many contour values as are required for that polygon.

The number of contour levels is determined by the Number: field on the Colormap Specification panel. The limit is 100 contour slices.

Figure 105 Scalar Colormap Specification Panel

All values of the current Scalar Function which are less than or equal to the Minimum will map to the color at the bottom of the color bar. All values greater than or equal to the Maximum will map to the color at the top of the color bar. All scalar values in between will map linearly (unless Log Scale is turned on) to intermediate positions in the color bar.

Filled Contour Mode
When Filled Contour mode is enabled the valid range for number of contour lines is 2 through 101 (number of contour slices is equal to the number of contour lines minus 1). The number in the Contour Control section of the panel indicates the number of contour lines. The number of contour bands is the number of contour lines minus 1.

Low-Fidelity Colormaps

Low-fidelity colormaps such as ColorStriped, Black & White Striped and Zebra may yield poor sampling results for use with filled contour rendering. Continuous colormaps such as Spectrum exhibit better behavior.

Exceeding Max/Min Ranges

It is possible to type in values that exceed the maximum or minimum scalar function value range. If an out of range value is typed into either entry box, the slider and the value are reset to be equal to the new value. If you wish to reset the maximum and minimum scalar function values to their original values from the Scalar Function Register, you must recalculate the scalar function.
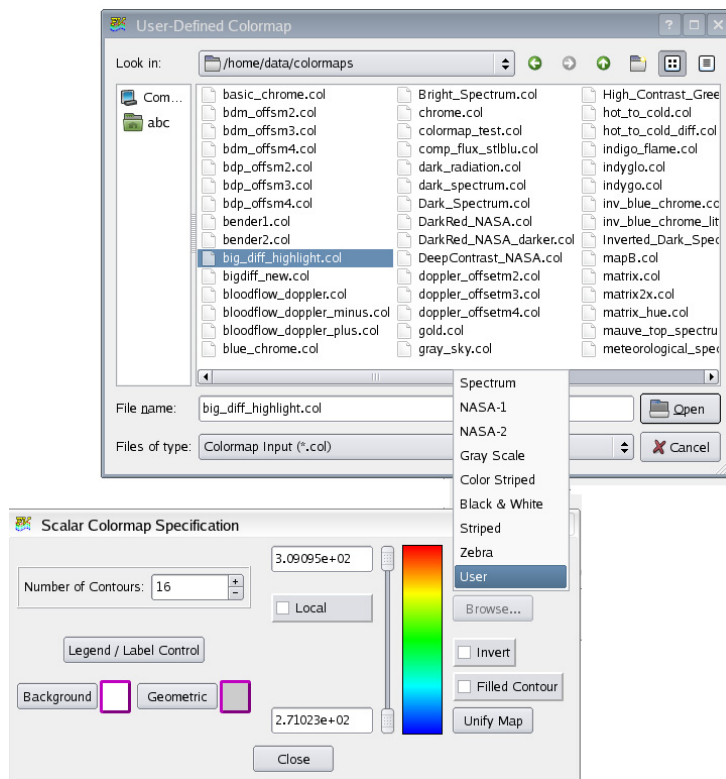
Figure 106 Selecting a Colormap

Several colormaps are built into **FieldView** already and can be selected from the Colormap Specification panel (see **Figure 106**). When the user option is selected, a file browser is launched permitting the selection of any User definable colormap. Double-click the colormap file name to keep the browser up while previewing the spectrum bar in the Colormap Specification panel. Several user defined colormaps are included. For more information on the format of colormaps, please refer to **Appendix E Colormap File Format** in the **FieldView Reference Manual**.

Multiple Colormaps

Colormap attributes are an attribute of the individual visualization objects (i.e., boundary, coordinate, computational and iso-surfaces, and, streamlines, streaklines, particle paths and vortex cores/surface flows). The colormap attributes are:

1. The choice of Colormap spectrum to show the scalar range.

2. Whether the spectrum is inverted or not. By default, Invert is OFF.

3. Whether filled contours are applied to the current visualization object. By default, Filled Contour is OFF.

4. Whether local scaling of the scalar associated with the visualization object is applied. By default, Local scaling is OFF.

An important implication of this feature is the ability to use multiple colormaps in the same visualization. Examples using two colormaps are shown in **Figure 107 Illustrations using Multiple Colormaps**. A grayscale colormap is used to emphasize the shockwave reflections inside the scramjet (at left) distinguishing them from the streamlines where flow speed is displayed in color. At right, a colormap limited to a range of black and light blue is used to show Schlieren contours in the region of an ejector nozzle.
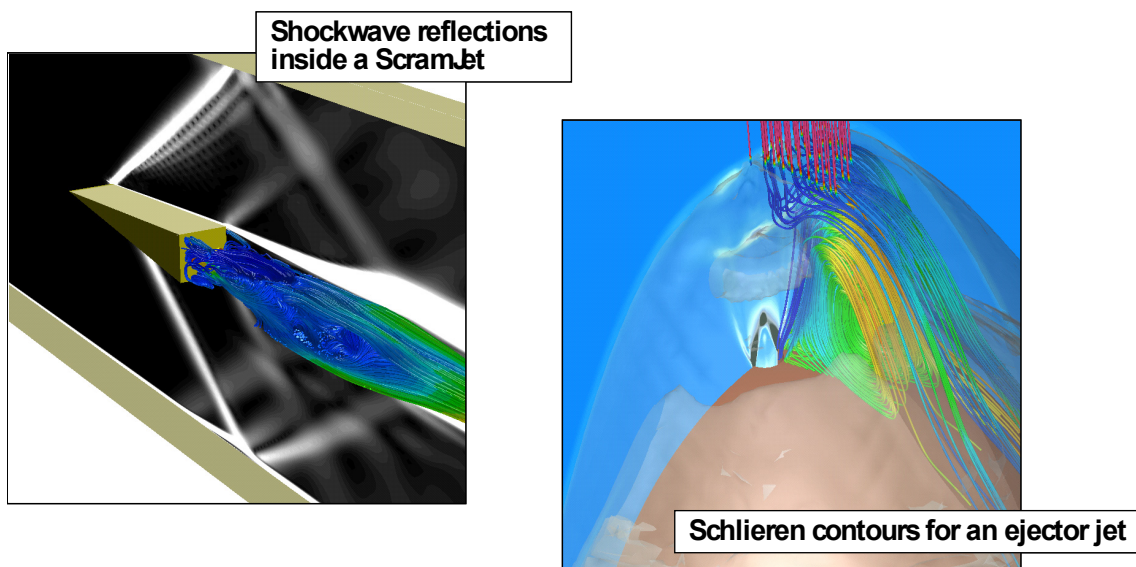
Figure 107 Illustrations using Multiple Colormaps

When a surface is 'current', the Colormap specification panel displays the colormap attributes that match the current surface. Changes made to the colormap attributes in the Colormap Specification panel is applied to the current object *only*. These attributes are also saved as part of the RESTART associated with that visualization object type. Changing to another surface type by bringing up a different Visualization Panel, for the case in which there are no existing objects of that type, are reset the colormap attributes in the Colormap Specification panel to the session defaults. The session defaults are initially applied when a new surface is created and may be modified. When a second surface of any type is created, the attributes for that surface, including the colormap attributes, are passed along to that surface.

Note: Colormap attributes are only visible if scalar coloring is used for the current object(s) and/or contour lines.

There are three conditions for which the Colormap Specification panel will display the global or session default colormap.

Condition 1 - When **FieldView** is initially started and no visualization objects have yet been created.

Condition 2 - When no object visualization panels (ie boundary, coordinate) are displayed.

Condition 3 - When the object visualization panels are of the type 2D Plots, Point Probe, or Titles/Annotation. For these panels, scalar coloring cannot be applied.

The colormap which is displayed under these conditions is saved as part of the Colormap Specification RESTART (`.map`) file.

Note: If you save a RESTART when a visualization panel is open, the colormap displayed in the Colormap Specification panel will NOT be saved as part of the Colormap Specification RESTART (`.map`) file. To change the session defaults, close the current Visualization Panel, modify the colormap attributes in the Colormap Specification panel and save a Colormap Specification RESTART.

Restarts from previous versions of **FieldView** all share the same colormap attributes. When reading a complete restart, the attribute settings come from the colormap restart file (.map). When reading a panel restart, the settings are picked up from the current default settings, whatever they may be.

Figure 108 Warning Message for Unify Map

The "Unify Map" button applies the current colormap attributes to all objects of all types across all datasets. Before this action is applied, a warning message is presented to let you cancel this operation (see **Figure 108 Warning Message for Unify Map**). Note that exiting a Visualization Panel after a Unify Map operation resets the session default colormap attributes in the Colormap Specification panel and may no longer match those that were just unified.

The "Unify Map" operation and the "Unify" option in the Tools menu do not do the same thing. The Tools menu "Unify" command applies the colormap attributes of the current object to all objects of the same type in the current dataset (for example, to all Coordinate Surfaces if the current object is a Coordinate Surface). The Unify Map operation is applied to ALL surfaces of ALL types in ALL datasets.
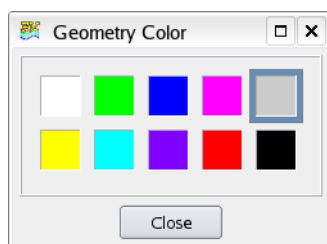
When the Geom color chip is selected in the Colormap Specification Panel, a sub-panel of color options allows the selection of a new geometry color. As shown, the panel allows you to change the current geometry color. The geometry color controls Geometric colored surfaces and rakes and also Legend and Title text.

Figure 109 Geometry Color Panel

Background Color and Images

When the 'Background' button or color chip is clicked in the Scalar Colormap Specification Panel, the Background Color sub-panel allows the selection from color options or the insertion of custom background images into the **FieldView** graphics window. To place a background image, two steps are required. First the background image must be

selected, and next it must be turned on. To specify a background image, perform step 1 to click the Select... button in the Background Color GUI (illustrated below in **Figure 110 Steps to select and display a background image**). This brings up a file browser. From this file browser window, you can select a background image. Officially supported formats include `.bmp`, `.jpg`, `.png`, and `.gif`. On most platforms, we expect that other file formats will work, depending on whether they have been created or saved using compatible byte ordering and/or compression. When an image has been selected, the file name appears in the Background Color GUI. The image in the graphics window can be toggled ON/OFF using the Background Image check button. Note that all visual objects created by **FieldView**, such as surfaces, rakes & titles always appear in front of the background image. There is no way to control the layer level of the background image.
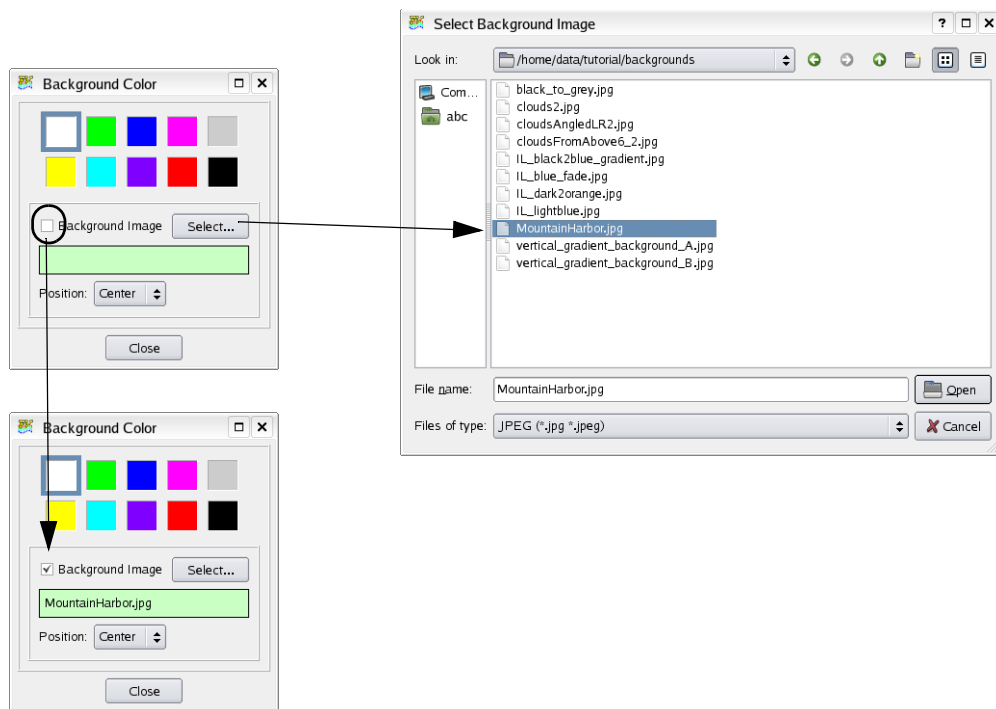
Figure 110 Steps to select and display a background image

The Position Pop-Up selector offers three options for positioning the background image in the graphics window, namely, Center, Stretch and Fit. The outcome for each choice is illustrated below:

Position: Center
This option aligns the center of the background image with the center of the graphics window. If the background image is larger than the graphics window in either or both dimensions, it will be resized until either the vertical or horizontal dimension matches the graphics window. If the background image is smaller than the graphics window in either or both dimensions, the background color set in the Colormap Speci-

Figure 111 Applying Position option to a background image

fication panel will appear as a border between the edges of the background image and the graphics window in the dimension(s) falling short of the graphics window boundaries. In the example illustrated, the background color is 'white', and the original image is smaller than the graphics window.

Position: Stretch
This option will scale the background image to exactly fill the graphics window in both the horizontal and vertical dimensions. Note that if the aspect ratio of the background image is *not* the same as that of the graphics window, this will distort the background image. For many backgrounds, some distortion may not be objectionable, for example, images of generic clouds or terrain.

Position: Fit
This option will uniformly scale the background image to fill the graphics window in one dimension, maintaining its aspect ratio and preventing cropping or distortion. If the background image has a wider aspect ratio than the graphics window, it will be "letterboxed" with the background color set in the Colormap Specification panel at the top and bottom of the graphics window. If the background image has a narrower aspect ratio than the graphics window, the background color will appear at the sides of the graphics window. When the graphics window is smaller than the background image, there is no difference between the 'center' and 'fit' position options.

The background image below (see **Figure 112 Fit vs Stretch for a tall, narrow background image**) is narrower than the (default) graphics window. When Position: Fit is chosen, the background image is scaled until it meets the bottom and top of the graphics window. Using the Position: Stretch option, the background image is distorted to match the dimensions of the graphics window.
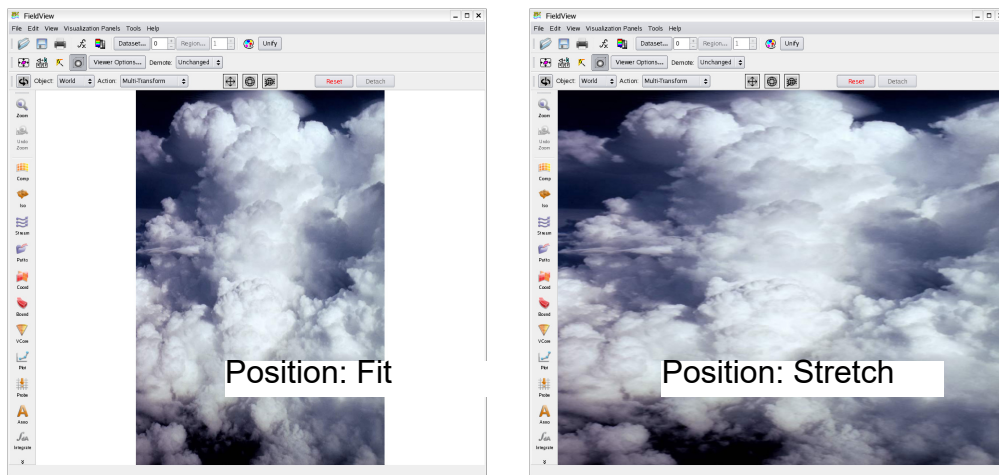
Figure 112 Fit vs Stretch for a tall, narrow background image

If the graphics window is rescaled at any time, the background image will automatically be refitted to the new window size.

Known Limitations
Certain types of file compression and/or byte ordering for tiff format images are not supported.

Environments

The technique used by **FieldView** for Material Rendering is called 'environment mapping'. (**See "Materials" on page 243.**) It uses images in the lighting calculation to simulate reflection. To obtain a more realistic effect, users can pick from six predefined environment images the one that better suits their geometry and application. The image below shows the effect of this environment setting on a test surface rendered as aluminum.
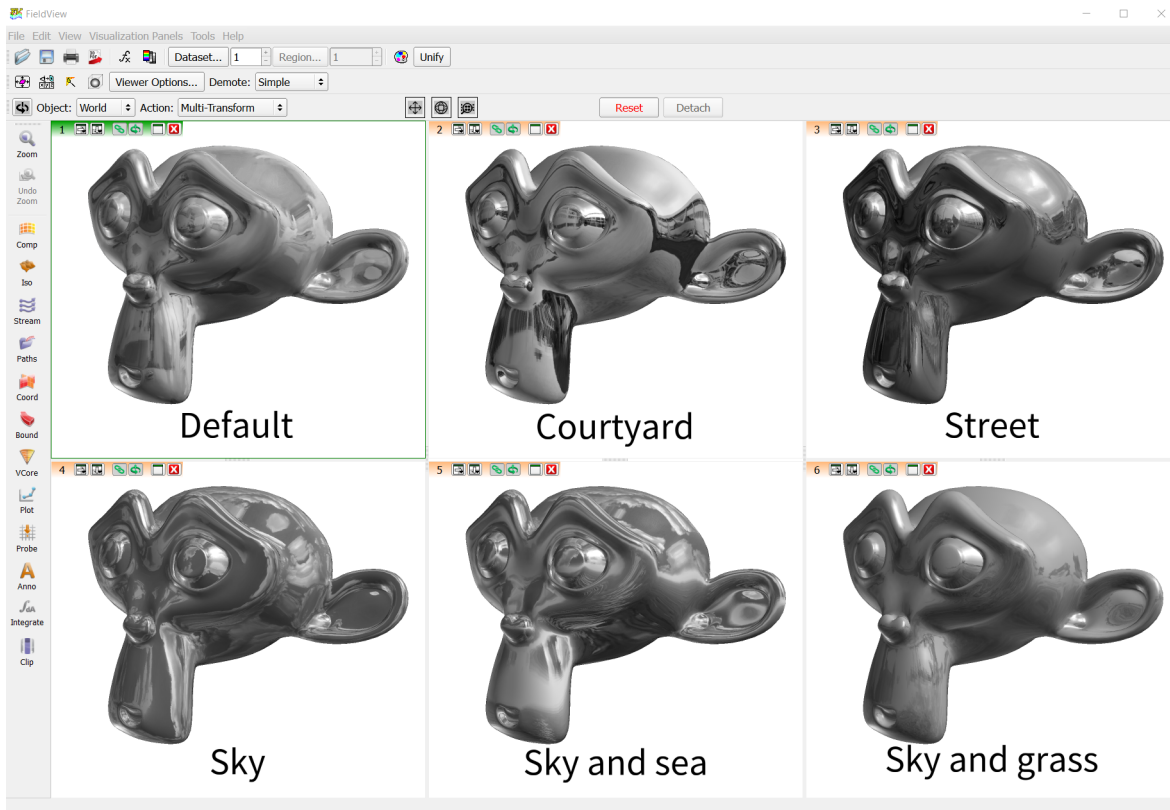
Figure 113 Effect of the environment setting on a test surface rendered as aluminum

The environment setting can be found in the Background / Environment panel, opened from the View menu.
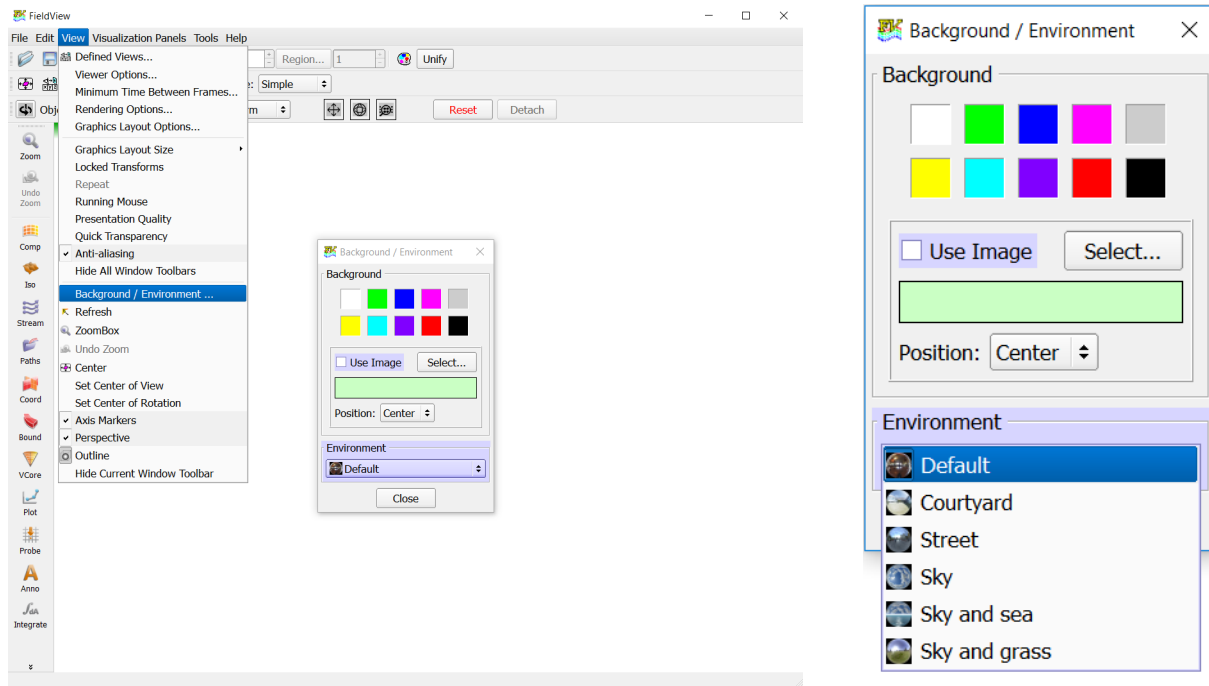
Figure 114 The Environment section in the Background / Environment panel

One environment can be set per window. The current environment for each window will be saved and restored by a Complete Restart. When **FieldView** starts, it will use by default the last environment used in window #1.

Local Scaling

A local scaling function is available for all surfaces and rakes. It is activated by simply clicking on the Local button in the Colormap Specification panel. When Local is activated, the range of the colormap spectrum changes to show the local minimum and maximum for the given scalar of the current surface or rake. The sliders for the Colormap spectrum move to the bottom and top of the GUI respectively. The calculated extremum values are based upon the visible part of a surface. Thresholding, Subsetting by Region, and Region visibility affect the reported values.

Previous settings for the Colormap range are stored. If the Local button has been turned ON then turning this button OFF will return the Colormap to the previously established settings. However, if the sliders are adjusted while the Local button is ON, these settings will be retained when the Local button is turned OFF again.

Note: If you use the UNIFY operation on a surface where the Local button is ON, the surfaces that are unified will be set to the values of the surface that was used to make the unification. For example, if Surface 1 has the Local Min/ Max settings turned on and Surface 2's Local button is OFF and you transfer Surface 1 settings to Surface 2 using UNIFY, the Surface 2 Local button will remain off and the Colormap will inherit Surface 1 settings. If you want to display the local Min/Max for Surface 2, simply turn on the Local button and Surface 1 settings will be deleted and will be replaced with the local values of Surface 2. See **Unify**, in Chapter 14.

Face Data and the Colormap Specification Panel

Face data consists of a single value per face, instead of a value for each node, with a face element made up of 3 or more nodes. Hence, each face has a single color assigned to it. The face color is constant (using Constant Shading or Faceted Shading) or has light source shading (using Smooth Shading). Each face has a scalar value, which determines its color given a specific colormap and scalar min/max values. Hence, turning on Filled Contour can change the mapping, resulting in a different colored face. The difference is greatest with few contours and least with a large number of contours.
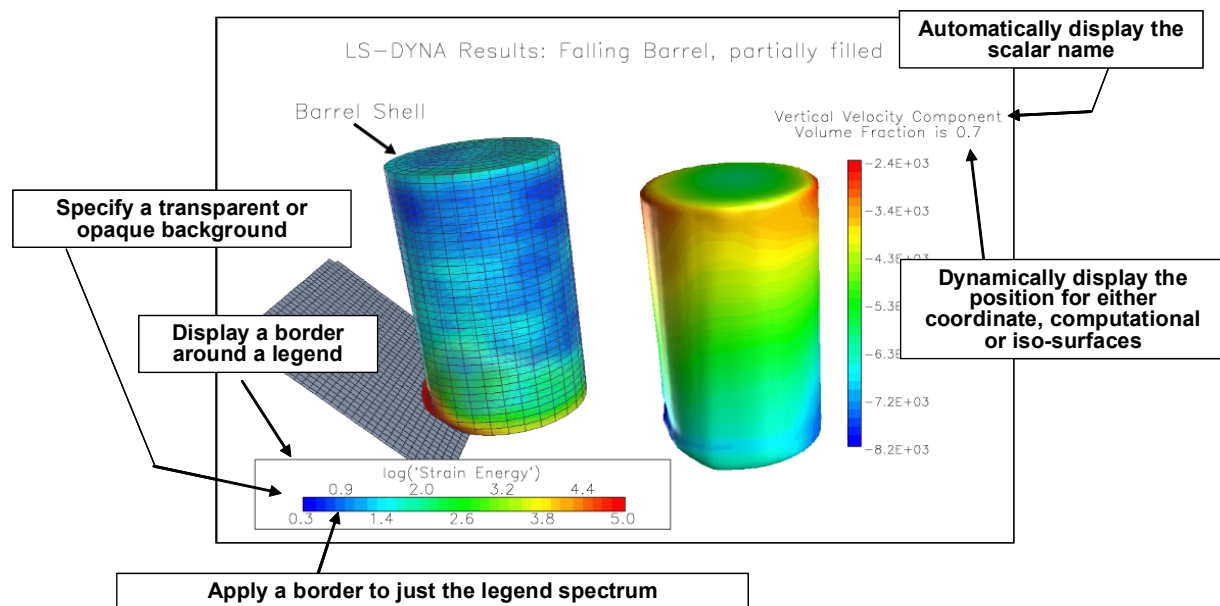
Figure 115 Illustration of Scalar Legend display features

Known Limitations

At present, there is no **FVX** support for controlling legend attributes and properties. The attributes are stored as part of surface and/or rake restarts, so it is possible to apply the legend features in **FVX** by using the `fv_script` command to call the desired restart.

There is no way to apply ordering to legends within the graphics window. Depending upon the graphical elements within the window the fixed legend order will be applied as follows, starting from the bottom layer: coordinate, boundary, comp, iso, streamline, particle path, vortex core, surface streamlines.

Interactive scaling of the legend using the right mouse (M3) in combination with the mode 'Running Mouse', can be awkward on some systems. As a work-around, we recommend switching the 'Running Mouse' off for legend re-sizing.

The starting color for legend annotation depends upon the background color of the graphics window. If you start **FieldView**, the default background color for the graphics window is black, and if you create a legend, the annotation color will be white. If you change the background color to white prior to creating any legends, then the first time you create a legend, the annotation color will be black. Because of this behavior, it is possible to loose your legend annotation in the background color by creating a legend and then subsequently changing the graphics background from black to white. We recommend that you choose your background color for the graphics window as a first step before you read data. You can set the background color to default to the color of your choice by creating a **FieldView** Preference Restart (see **Chapter 5** of the **Reference Manual**, and **See "Preference Restart" on page 12.** of the **User's Guide** for information on where they can be placed to load automatically.)

Legend/Label Control

The Legend panel (**Figure 116 Legend Panel launched from the Scalar Colormap Specification Panel**) can be launched for the current surface from the Scalar Colormap Specification panel and allows you to create Legends and Labels for Spectrums and Contours for scalar-colored objects. See also **Legend Tab Controls** for redundant access to these features from all visualization panels.

Note: All legends will always appear in front of any other graphics.

Legend attributes　　　Spectrum attributes　　　Contour attributes

Specifications for Legend Numbers

When Log Scale is used for a surface Colormap, the numerical format "10^n" is available for legend annotations (see **Powers of 10 numerical format for legends**).

Specifications for Legend Titles

Text color control
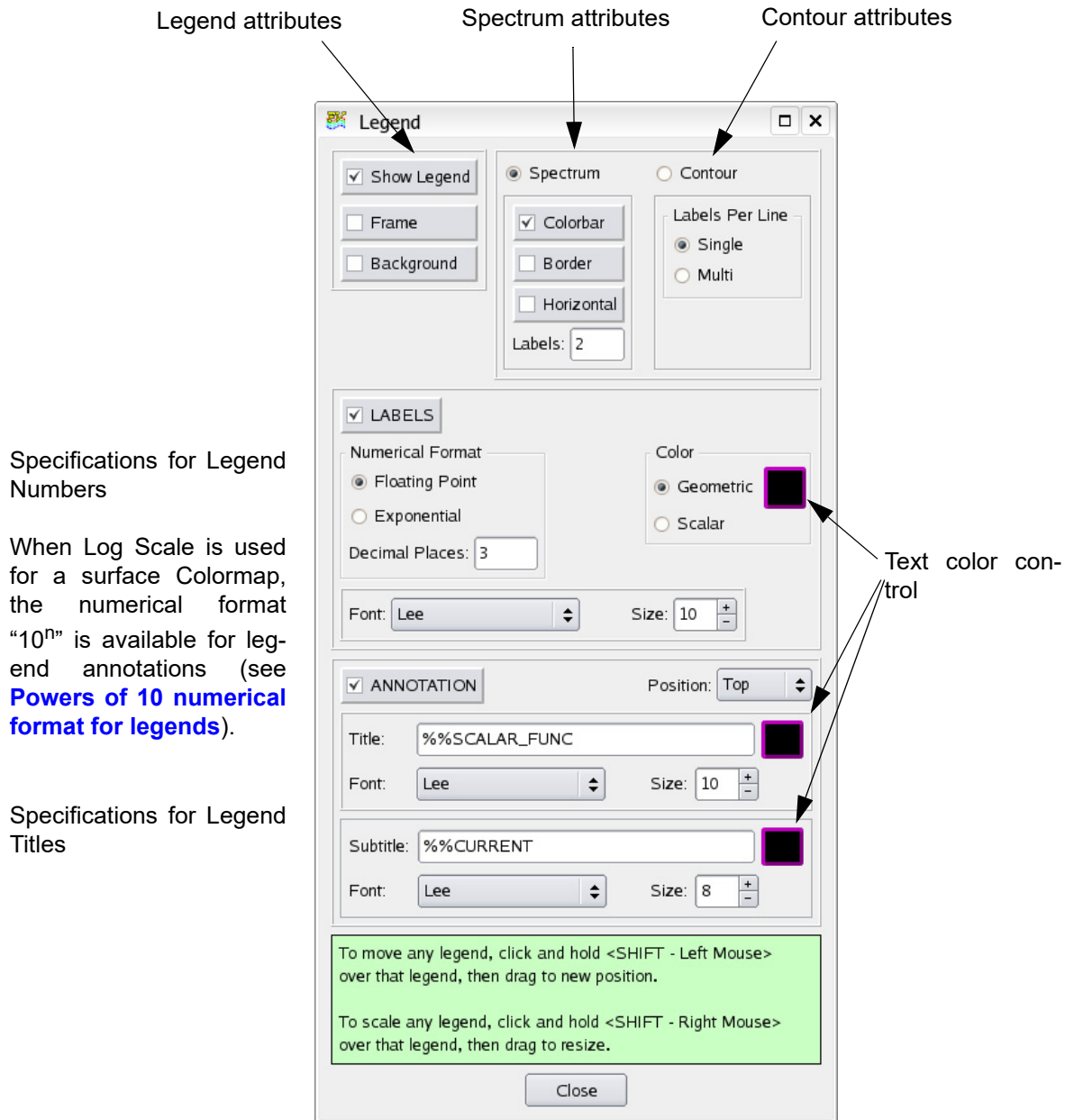
Figure 116 Legend Panel launched from the Scalar Colormap Specification Panel

Using the Legend Controls

How do I create a Legend in the graphics window?
To create a Legend that corresponds to a particular surface/rake, first, make sure that the panel that was used to create that surface/rake is current. Next, be sure that the current surface/rake is the one for which you want a legend, and that it is scalar colored. Then,

go to the Legend Tab on the current panel. Turn on SHOW LEGEND. Select a Legend Type of Spectrum or Contour (only valid if the surface is shown as contour lines or uses Contours as Attributes, see below). Legend Color is controlled by the Geometry color button on the Colormap Specification panel.

Can I add more labels to the Legend?
By default, each Legend will have two labels, one for the minimum and one for the maximum. If more labels are desired, type the number of labels you want in the Num. Labels field in the Spectrum Control area.

How do I control the size and placement of the Legend?
Placement of the legend in the graphics window is controlled using the SHIFT+Left Mouse button. A hint is provided at the bottom of the Legend tab.

Possible Problems

Why doesn't my Legend appear in the graphics window?
A legend always corresponds to the current surface or rake. If the current surface/rake is not scalar colored, or is not visible, the Legend will not appear when it is turned on. Similarly, if the Legend Type is contour and the current surface is not displayed as contour lines the Legend will not be displayed.

Transient Data Controls

If a transient (time-dependent) dataset has been read, the Transient Data Controls panel (**Figure 117 Transient Data Controls Panel**) is enabled. This panel allows for sweeping the current display through time, or specifying a specific time at which the data should be shown.

The current TIME STEP number and SOLUTION TIME are displayed here. To change a value, type into the appropriate box or move the slider bar. Next, you must press the Apply button in order update the graphics window for the new time value.

Upon pressing the Sweep button (which changes to Build or Export As... if Flipbook Build Mode or Transient Export... is on), the graphics window updates for each TIME STEP from the specified Begin to End points. Speed allows you to increase the Minimum Time Between Frames for slower playback.

The Skip factor is the number that is added to the current TIME STEP in order to see the next requested step. The Skip and Loop controls are also covered in the **Chapter 6** section on **Streaklines**. Please refer to that chapter for details of these two controls.

Use Delta Time allows you to apply a delta time to the time step numbers in the results file. That is, if the solution times in the results file are not meaningful, you can apply a delta time to the time step number such that the solution times will be calculated according to the formula: T = N * delta_time, where N is the time step value.
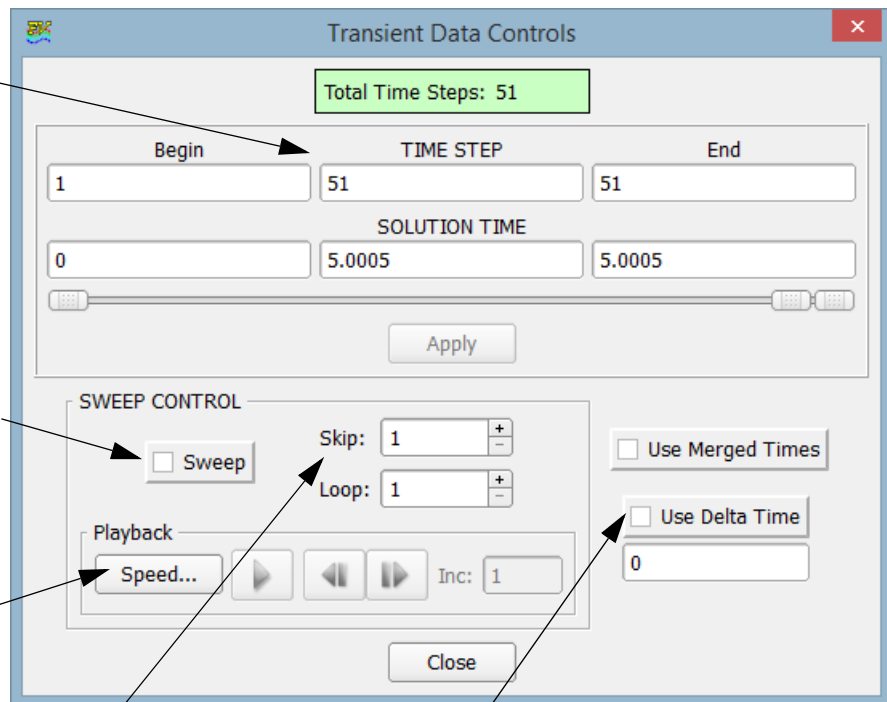
Figure 117 Transient Data Controls Panel

Skip and Loop

The Skip value affects whether all or only some of the time steps are read in while performing a transient sweep (i.e., the Sweep button is turned on). If the value is 1 (the default), every time step available is read in. If the value is 2, then every other time step available is read in, etc.

The Loop value causes **FieldView** to continue a transient sweep by repeating the series. For example, suppose you have a transient series and you have the Begin value set at time step 4 (solution time 4.0 seconds), Current is at time step 4 and End is set at time step 10 (solution time 4.6 seconds for a delta time of 0.1 seconds per time step). A Loop value of 1 (the default) means that the transient series will be swept through once. A Loop value of 2 means that the series will be swept through twice. For this example, the displayed time steps (if using transient titles) would be: 4,5,6,7,8,9,10, then 11,12,13,14,15,16,17, where the data shown at time step 11 is the same as that at time step 4 (**FieldView** assumes the data is periodic), but the solution time shown in transient titles, and used in streakline calculations is the time of time step `10 + dt` where the

delta time `dt` is computed from the time difference between the last two time steps 9 and 10.

Warning: If in the above example, the End value has been manually clipped (i.e. the transient series actually extends past time step 10 and you moved the End slider in), **FieldView** will not use the actual data from time step 11, but the data from time step 4.

Use Delta Time

By default, the Use Delta Time button is OFF and the solution times come from the original data (as read in). The type-in field takes a positive floating point number which will be used as the constant delta solution time between consecutive time steps. The default value in the Use Delta Time type-in will be 1.0.

When the Use Delta Time check box is turned ON, the value in the type-in field will be used as a constant delta time to compute new solution times. The solution times so calculated replace the solution times from the original data; the new solution times are displayed in the SOLUTION TIME text fields, will be used in streakline calculation, and will be used in annotation text using the `%%T` transient escape sequence.

When the Use Delta Time check box is turned OFF after having been turned ON, the solution times will be set back to the original solution time values.

When computing the new solution times, **FieldView** will assume that time steps start at 1 and that the solution time at time step 1 is `delta_time`. Therefore the solution time at time step number `N` would be `T = N * delta_time`. If the first time step is numbered 0, then the first solution time computed would be 0.000.

Important Note: All of the default settings for the Scalar min/max field, vector field and threshold field are set by the time step you select on input when creating entities using those quantities while situated at the time step. For example, if you read in time step 5, as shown below, and create a surface colored by pressure, the min and max pressure will be set to 0 and 100. This will cause all pressures above 100 to be the same color for all time steps. This is also true for 2D Plots. The plot scale will be determined from the current time step. Data at subsequent time steps may fall outside of this range, in which case it will not be visible in the 2D Plot window. This can be avoided by setting the plot scale to the global min/max of the transient sequence.

The delta time can be used to sync transient datasets which have no solution time, or to create streaklines on data which has no solution time.

Use Merged Times

Use Merged Times allows you to animate all time steps using a common time line for multiple datasets which provide solution time. If solution time is not provided, **FieldView** can provide this by checking the Use Delta Time option. These features simplify the task of creating animations for multiple transient cases.

The concept of the merged time line is introduced in the example below:
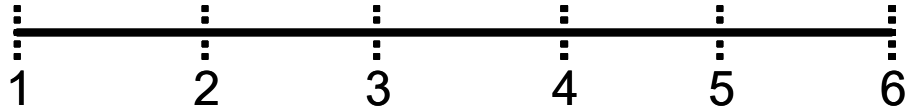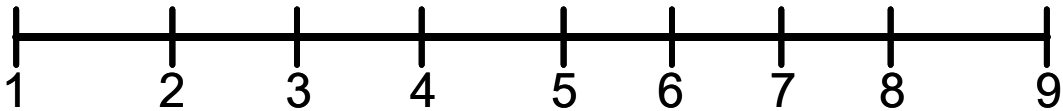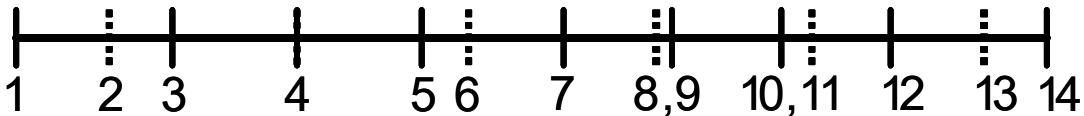
Figure 118 Merged Time Line Illustration

Here, we have loaded two transient cases into **FieldView**. There are 6 time steps in Dataset #1 and 9 time steps in Dataset #2. The solution times are represented along the horizontal line, and the time steps are shown as integer values. The solution time increases going from left to right in the illustration above. Dataset #2 starts earlier than Dataset #1. Dataset #2 also runs longer, as shown by the longer horizontal time line. To create the Merged Time Line, all times from both datasets are mapped onto a single common time line. The original solution times for the time steps within each dataset are preserved. A relative-based criterion, based on the product of the smallest time interval within a dataset and an epsilon value (1.0×10^{-3}), is used to establish whether a time step is unique within the merged time line. Once all unique solution times have been established, new numbering will be applied. To print out the dataset time steps and the merged time line time steps, you can use the Annotation (Text) escape sequence, `%%N`, as follows:

- Create a title, Dataset Time Step: `%%N1`, to show the time step for Dataset #1
- Create a title, Dataset Time Step: `%%N2`, to show the time step for Dataset #2

- Create a title, Merged Time Step: `%%N`, to show the merged time step

The same behavior applies to the solution time escape sequence, `%%T`. When no dataset number is specified, the solution time escape sequence will return the solution time for the merged time line.

Creation of the merged time line is enabled by pressing the Use Merged Times button on the Transient Controls Panel.

.

In normal operation, the Transient Data Controls panel contains time step and solution time data which is relevant to only the current dataset. This mode is used to create streaklines.
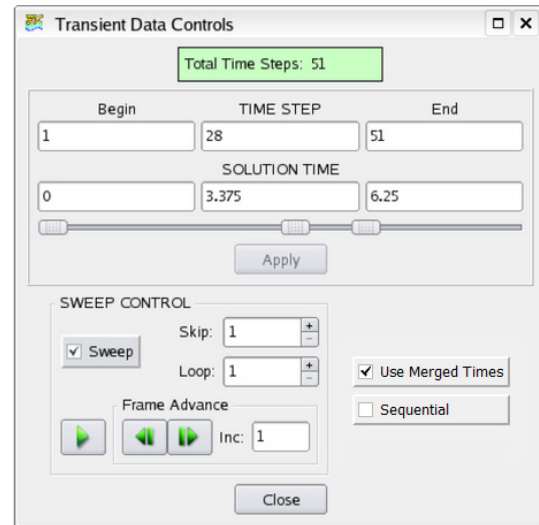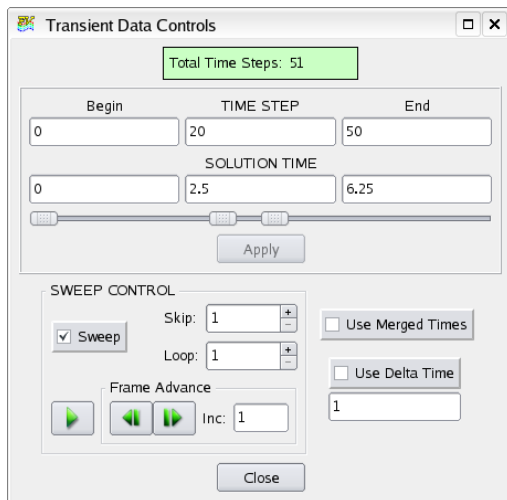
The Use Merged Times button will create a main time line for all transient datasets loaded into **FieldView**. New time steps will likely be created to represent all unique solution times.

Figure 119 Use Merged Times

As with previous versions of **FieldView**, we will not attempt to interpolate data in between time steps. In some cases, during a transient sweep through many datasets, a dataset may appear to be sitting on the same solution time for several successive steps. Another implication of having the Use Merged Times button set is that when another transient dataset is appended, this new dataset's solution times will be merged into the common time line, possibly creating additional time steps.

Restarts created with **FieldView** 12 and later will save and restore this setting. Older restarts will force this mode off. When the merged time mode is disabled, the Transient Data Controls will revert to show the time information for just the current dataset.

A script command permits the automated operation of turning this mode on or off. The command is:

```
      TIME SET MERGEDTIMES ON/OFF
```

This is similar in form and syntax to a previously existing command to set the delta-time, or common time interval, to be applied to all time steps in a series:

```
      TIME SET DELTATIME delta-time
```

When Merged Times is set to be on, either by pressing the button on the GUI, or through the use of the **FieldView** script command, the ability to specify and apply a delta-time will be disabled. In this mode, we will also disable the ability to calculate streaklines. It is recommended that streakline calculations are (pre-)computed for each dataset, and this can be done most efficiently in batch mode. All transient streakline data, for all supported transient particle path file formats, will play back correctly synchronized with their respective datasets.

Transient Sequential Visibility

This feature is designed to simplify sweeping and animating multiple transient datasets, each of which covers a different range of solution times. Please refer to the example illustrated below:
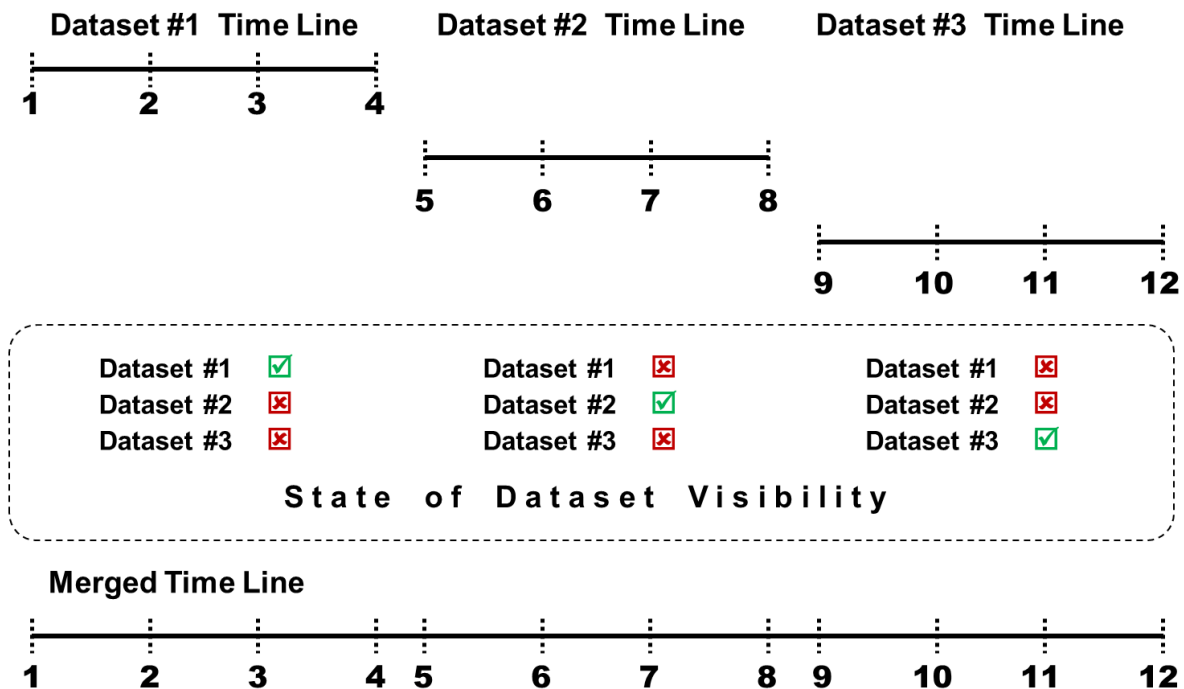
Figure 120 Transient Sequential Visibility Example

Here, we have three transient datasets in a single **FieldView** session. Each of the datasets covers a different range of solution times. Turning on the Use Merged Times button on the Transient Data Controls panel lets us sweep through all three of the datasets,

advancing each dataset according to the merged time line. If we were to create a transient animation at this point, the visibility of all three datasets will remain on during the sweep. When **Use Merged Times** is selected, a button called **Sequential** appears on the **Transient Data Controls** panel. When this button is toggled on, a dataset is visible only if the current solution time on the Transient Data Controls panel matches a solution time for that dataset. If no matching solution time exists for the dataset, the visibility for that dataset is toggled off.
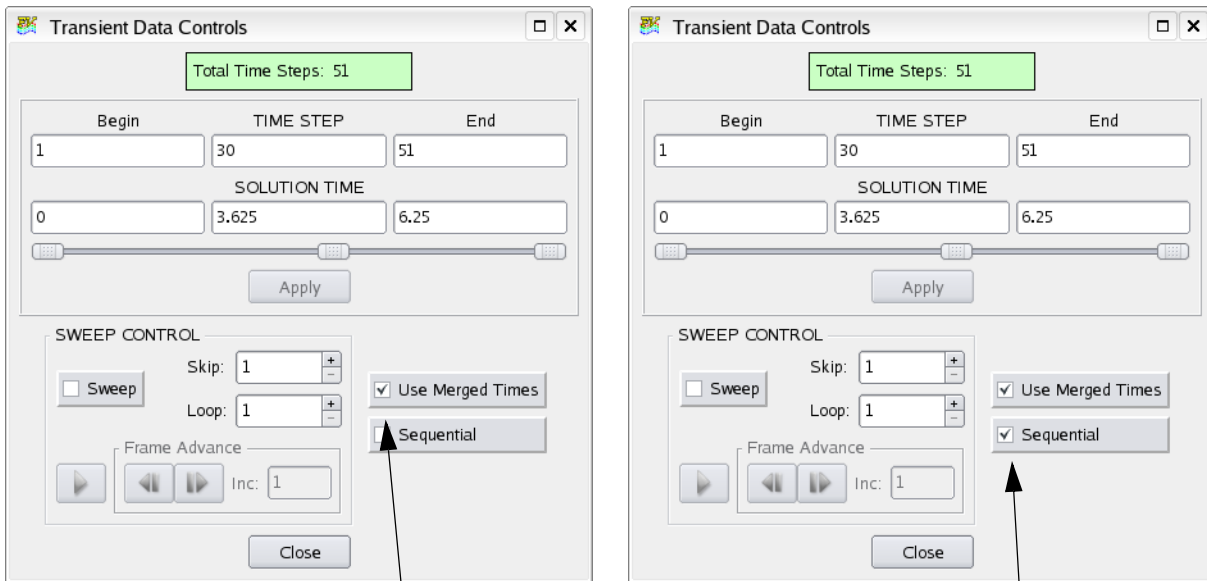
The Transient Sequential Visibility button is only available when Use Merged Times is turned on.

Turn Sequential on to enable Transient Sequential Visibility for a transient sweep.

Figure 121 Transient Sequential Visibility Controls

Transient sequential visibility can be automated using a SCRIPT command:

```
TIME SET SEQUENTIAL ON/OFF
```

Since this mode of operation is only enabled when Use Merged Times is on, a complete SCRIPT to record a transient animation may look something like this:

```
RESTART ALL multi-transient-example
TIME SET MERGEDTIMES ON
TIME SET SEQUENTIAL ON
RECORD ON my-sequential-transient
SWEEP TIME 1 1
```

`RECORD OFF`

Transient Sweep Caching is fully enabled to work with Transient Sequential Visualization.

Transient Sweep Caching

Transient Sweep caching is designed to let you advance to any time step within the cache without having to rebuild it. Additionally, all view operations such as centering, switching between locked and multi transforms, or applying dataset and surface transforms during the transient sweep cache playback are supported.

The Transient Controls Panel pictured above features a combined Play / Pause button and "Frame Advance" controls, featuring an Inc type-in field and buttons for skipping forward or backward through the cached frames.

When the "Sweep" button is ON, the transient sweep is in "cache mode". During the first sweep, the cache is being built. After the transient cache has been built, **FieldView** runs in a playback mode.   While the transient sweep cache is being initially created, you can interrupt the process at any time by pressing the Play button (it switches to Pause when this is done). Pressing the Pause button to return back to the playback mode will play back what you have cached to this point. The "Frame Forward" or "Frame Backward" buttons can be used to navigate through the cached time steps. You can also change time steps by entering a new time step or solution time in the appropriate fields, or, by simply dragging the slider to a new step. The Inc field not only controls by how many frames the "Frame Forward" or "Frame Backward" action should go, it also controls the playback of the cached frames (ie, when playback is not paused).

While the "Sweep" button is ON, most of the **FieldView** GUI, with the exception of the view controls is disabled. This is done to protect you from inadvertently changing the underlying content of the cached data, thereby forcing the cache to be recalculated.

When the "Sweep" button is turned OFF, the transient sweep is uncached, and the graphics window is updated to match the time step. The new GUI controls are also disabled when the "Sweep" button is turned OFF.

These new controls provide you with unprecedented flexibility and control over transient sweep caches.

Transient sweep caching is only done during interactive sweeps. Automated transient sweeps using SCRIPT or **FVX** commands are not cached. Also, a transient sweep cache is not built during a flipbook build, a transient XDB export or during the calculation of streamlines.

Merged Transient Sweep Behavior

It is possible to sweep multiple transient datasets through time simultaneously by turning on Use Merged Times on the Transient Data controls panel (navigate from Tools → Transient Data...).

By default, Use Merged Times is turned off. If multiple transient datasets have been loaded into **FieldView**, pressing the Sweep button on the Transient Data Controls panel will sweep the current dataset in time. This is the behavior for **FieldView** for handling transient data.

It is also possible to perform transient sweeping across multiple datasets in multiple windows. Control for this operation is accessible from the link icon, which is the same control used for linked surface sweeping. Linked surface sweeping and merged transient sweeping are exclusive operations - only one of these can be performed at a time.

The default operation for merged transient sweeping is shown at left in **Figure 122**. To sweep all the datasets in time in all windows, enable Use Merged Times from the Transient Dataset Controls panel and hit the Sweep button.
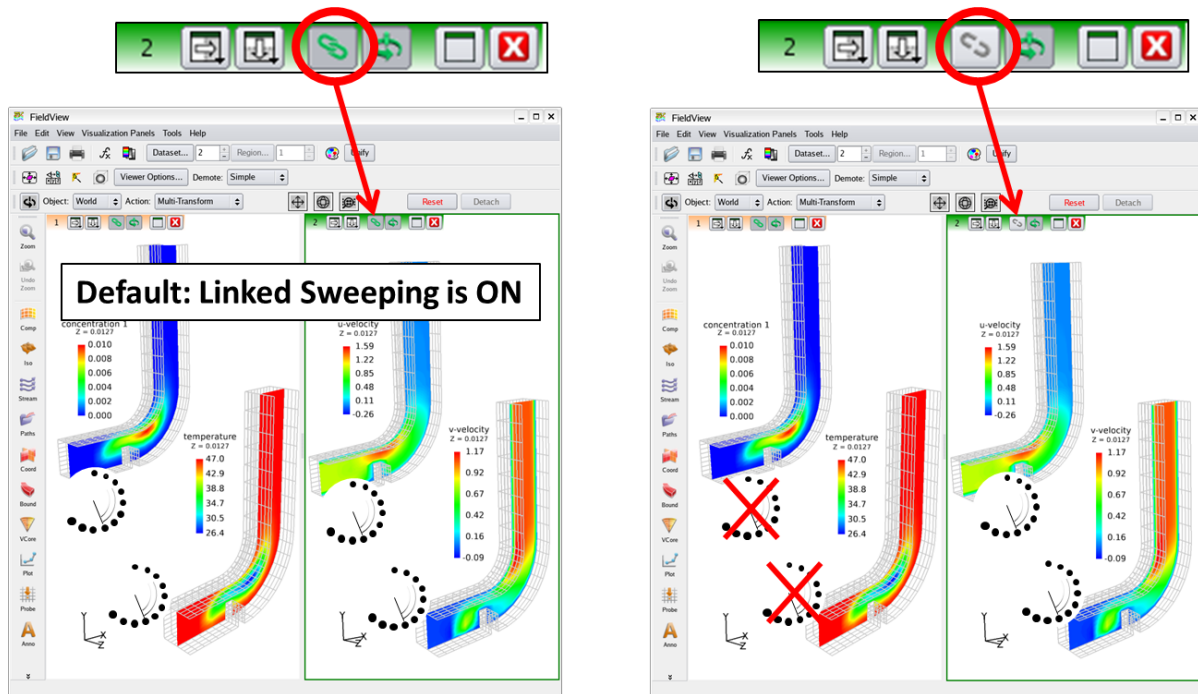
Figure 122 Merged Transient Sweep Behavior

To simultaneously sweep the transient datasets in just window #2, disable the linked sweep setting in the window toolbar by pressing the link icon. With Use Merged Times set from the Transient Data Controls panel, select any one of the datasets in window #2

and hit the Sweep button for that dataset in window #2. With window #1 and window #2 unlinked, you can sweep just the transient datasets in window #1 in the same way.

To sweep only one of the datasets in time, turn Use Merged Times off in the Transient Data Controls panel.

Frequently Asked Questions

How do I read in transient data from FIDAP, FLOW-3D®, CFX-4 or STAR-CD?
FIDAP, FLOW-3D® and STAR-CD store all of the time steps for a dataset in one file. To read in the data, simply select that file for input. You will then be shown a list of all of the time steps in the file, and be prompted to select a particular time step to work with. This is the time step that will be resident in memory after input. Thus, any surfaces you create, or variables you display, will be shown for that selected time step. All of the default settings for the Scalar min/max field, vector field and threshold field will be set by this time step if you create entities using those quantities while located at that time step. For example, if you select a time step where the pressure ranges from 0 to 100, and create a coordinate surface colored by pressure, and then use the Transient Data Controls to specify a different time step where the pressure ranges from 0 to 500, **FieldView** will use the values of 0 and 100 for displaying the scalar range on the surface. Furthermore, creating subsequent scalar colored surfaces will inherit those settings, and behave similarly.
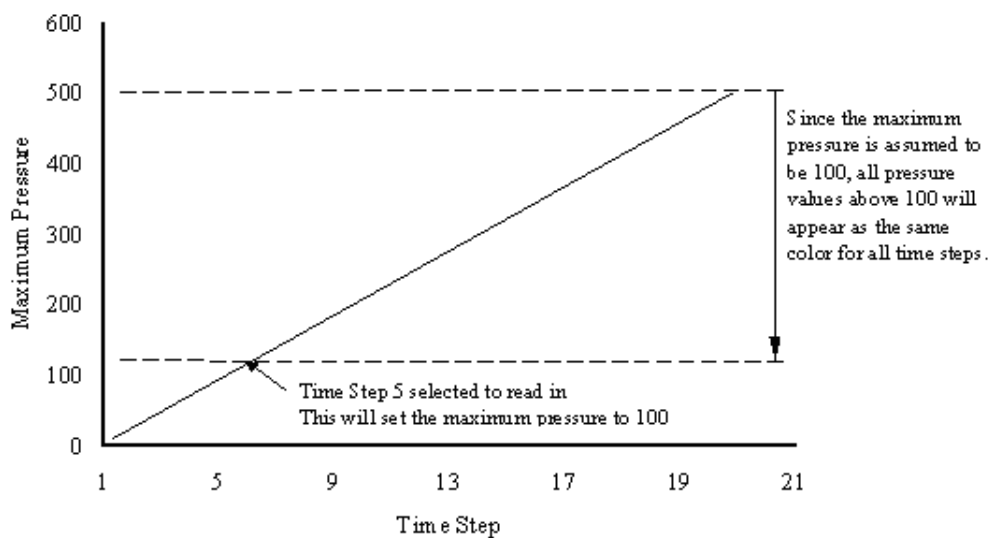
Figure 123 Transient Min/Max Determination

Default ranges specific to any given time step can be reset simply by reloading the respective register while located at that time step. You may type in the min and max you desire on the Colormap panel at any time.

How do I read in transient data that is in PLOT3D or Function files?
For PLOT3D style data (in Q and/or Function files), each time step is stored in a separate file following a naming convention described in **Transient Data** (see **Chapter 1** of the **Reference Manual** for a complete discussion of **Transient PLOT3D Data**). When **Field-View** encounters a series of files in a directory following this file naming convention, the user will be presented with the following dialog popup:
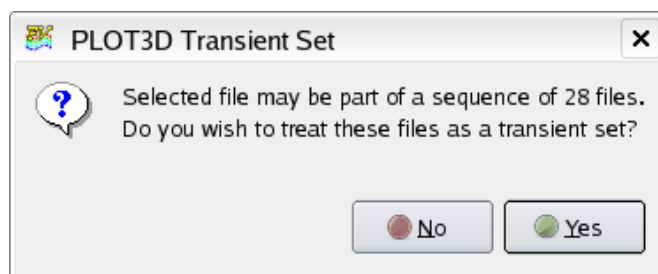
Figure 124 PLOT3D Transient Set Popup

To read in the data, select one of the group of files for input. If the files are part of a time series as shown above, you will be prompted for whether you wish to treat this data as a transient set, or whether you simply wish to read in the one file. If you select a transient set, the Transient Data Controls panel will be enabled, and the selected file will be read into memory. However, all of the default settings for the Scalar min/max field, vector field and threshold field will be set by this time step (see note above). FV-UNS (**FieldView** unstructured format) files work like PLOT3D transient data with a single file per time step.

How can I view specific time steps in my transient data?
After selecting the dataset for input, you should create the surface(s), streamlines, etc. that you wish to see by using the *current* time step you defined on input. Once you have created all of the items you wish to see change over time, select the Transient Data Controls panel from the Tools pull-down on the Main Menu Bar.

Time Step Number is an integer count of the time steps and ranges from 1 to the number of time steps read. Solution Time is a number provided by the solver (or present in the PLOT3D Q file header) that is a real number value of time during the transient simulation. Note: If you notice that the Solution Time ranges from 0 to 0, it is due to the fact that the time values in the file do not increase in each time step and are ignored.

To change time steps, simply type in the new time step number (or solution time value) into the Current type-in field on the panel, or move the middle slider bar to the correct

position. Next, press the Apply button. At this point, all items in the graphics window will be updated to now reflect the newly selected time step.

How can I animate through the time steps in my transient data?
Pressing the Sweep button will cause a series of time steps to be shown. The graphics window will be updated to show each time step from the value of Current to End and back to Current. The skip factor is used to show every Nth time step as the sweep moves from Begin to End. Please bear in mind that sweeping through a transient dataset can often take a considerable amount of time. If you would like to see a faster sweep through time, it is best to use the Flipbook Controls panel to create a movie of the time sweep. For more information, see **Chapter 6** of the **Reference Manual**.

What types of transient data are supported?
Currently, the following data types support transient data: FIDAP, FLOW-3D®, STAR-CD, and PLOT3D (including **FieldView** Function Files), **FieldView** Unstructured and CFX-4.

How can I display the Time Step Number and Solution Time in the graphics window?
If transient data has been read in, the following substitutions will be made on any title displayed in **FieldView**:

Title String	Actual Title Shown
%%N#	the Time Step number
%%T#	the Solution Time

where # is the dataset number.

For example, if your second of two datasets is transient, and you type the following:

　　　　Time Step = %%N2, which is equal to %%T2 seconds

you may see a title in the graphics window as follows:

　　　　Time Step = 3, which is equal to 10.36 seconds

These titles will, of course, update during a sweep of transient data.

Note: For single datasets, you do not have to put a number after %%N or %%T. If no value exists, a '1' is assumed.

What happens to Streamlines during a transient sweep?
When you have streamlines displayed for a given time step or even just the seeds (i.e. the Calculate button has not been pressed yet), and the Transient Data Controls panel is brought up and the Sweep button pushed, **FieldView** will calculate Streaklines instead of Streamlines. A transient sweep will calculate streaklines on all visible, seeded rakes belonging to the current dataset. No streamlines will be computed during a transient sweep for which at least one rake will compute streaklines. Only the current dataset (as specified on the Transform Controls and Dataset Controls panels) can calculate streaklines, because only the current dataset is "swept"; any other dataset remains at its current time step. Please refer to **Chapter 6** for a complete description of Streaklines.

Possible Problems

How come once I start sweeping through time, all my scalar values appear as one color?
The minimum and maximum scalar values are set after you select the single time step you wish to read in. These min and max values will be used for all time steps, as shown by the note on the first page of the Transient Data Controls section.

Integration Controls

The Integration Controls panel (**Figure 125 Integration Controls**) is used to perform numerical integration on certain visualization entities created by the user. The code provides for the integration of a single scalar function (stored in **FieldView**'s Scalar Function Register) over one or more surfaces in the current dataset. Any combination of surfaces may be used in the calculation, including Computational Surfaces, Coordinate Surfaces, Iso-Surfaces and Boundary Surfaces. In addition a "sweep" function is provided that calculates a sequence of integrals, by integrating at each step of a sweep operation.

Displays the name of the scalar function that will be integrated over the surface(s).

By default the integral will be calculated over all surfaces in the current dataset. If Current Surface is selected, then the integral will be calculated at the surface's current position. If Sweep Current Surface is selected, the current surface (Computational, Iso, or Coordinate) will be swept once from its minimum to maximum value, with the integral calculated at each sweep step. The results of these calculations will be placed in the Sweep Integration Output File. If Partial Surface is selected, the integration will be performed over selected part of Coordinate, or Iso-surface.

This area is used to select the name of the output file when Sweep Current Surface is selected.   If a file which already exists is selected, a choice will be given to overwrite, append, or cancel the integration.

Calculates the integral. If Sweep Current Surface is selected, the results will be placed in the specified output file, otherwise the results of the integration will be output here.
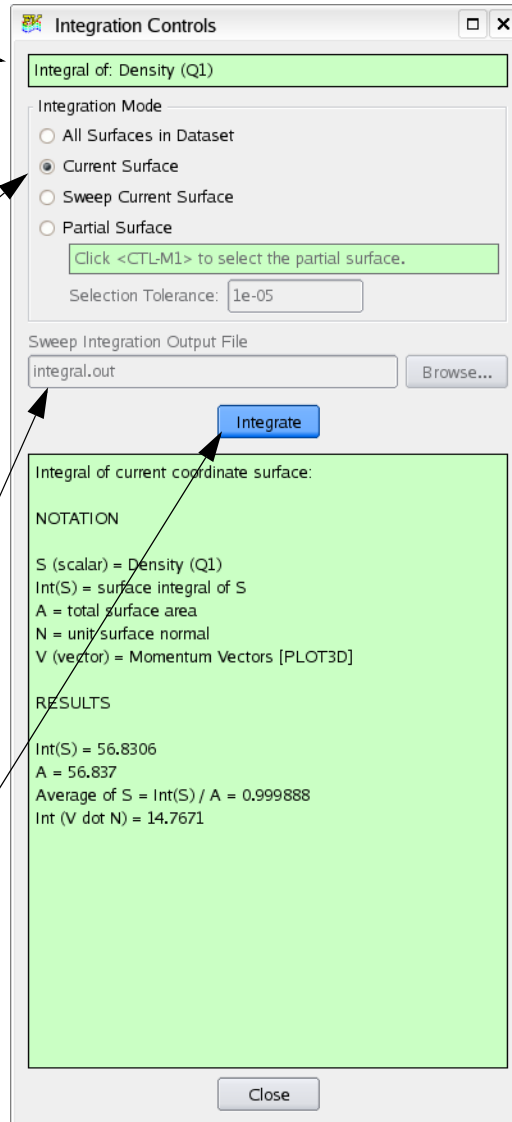
Figure 125 Integration Controls

The integrated result returned is for the visible portion of the surface.

Note: While Dynamic Clipping and Duplication are both active, all visible parts of duplicated data are included when obtaining integrations, but Int *(S * Nx)*, Int *(S * Ny)*, Int *(S * Nz)* and *Int (V dot N)* results will reflect values for the additional area in its original position with the original normal orientation. For this reason, the integration result will not be zero in the direction of mirroring. FV prints *WARNING: Integral includes all copies*

Existing surfaces in the current dataset may be omitted from inclusion in the integration calculation by turning the Visibility off for those surfaces. However, making a surface fully Transparent does not have the same effect. Transparent surfaces are included in integrations. With regions, it is possible to change the visible portion of a surface by subsetting using regions, or by simply toggling the visibility of individual regions off.

Output files are only generated for Sweep Current Surface integration results. It is possible to cut and paste integration results directly from the results section of the integration controls panel. When cylindrical coordinates are specified using an FVREG file (Region definition), Sweep Integration is performed in the R, T, or Z direction. This is carried through to the output file as well. See **Chapter 3** of the **Reference Manual** for more information..

If the mode is All Surfaces, Current Surface, or Partial Surface, the results of the integration will be output in this section. If Sweep Current Surface is selected, the results will be placed in the specified output file.

If no Surface Normals are present, then the integration results will be limited to the surface integral, Int(S), the surface area, A, and the average, Int(S)/A.

For results when Surface Normals are present, see **Figure 127 Integration Result Panel with Surface Normals**.

```
Integral of current coordinate surface:

NOTATION

S (scalar) = Density (Q1)
Int(S) = surface integral of S
A = total surface area
N = unit surface normal
V (vector) = Momentum Vectors [PLOT3D]

RESULTS

Int(S) = 56.8306
A = 56.837
Average of S = Int(S) / A = 0.999888
Int (V dot N) = 14.7671
```
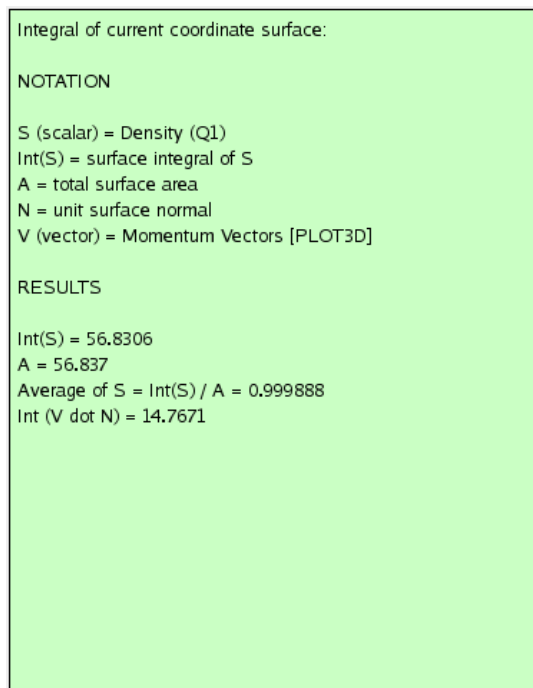
Figure 126 Integration Result Panel without Surface Normals

If the mode is All Surfaces, Current Surface, or Partial Surface, the results of the integration will be output in this section. If Sweep Current Surface is selected, the results will be placed in the specified output file. If Surface Normals are present, either Boundary, or Computational, or Coordinate, or Iso- Surface panel is current and the Integration Mode is Current Surface, or Partial Surface, then the integration results will contain additional results.

```
Integral of current boundary surface:

NOTATION

S (scalar) = pressure
Int(S) = surface integral of S
A = total surface area
N = unit surface normal
Nx = x component of N
V (vector) = velocity

RESULTS

Int(S) = 0.0651615
A = 0.00564944
Average of S = Int(S) / A = 11.5341
Int (S * Nx) = 0.000610711
Int (S * Ny) = -0.0319559
Int (S * Nz) = 0.000711149
Int (V dot N) = 0
```
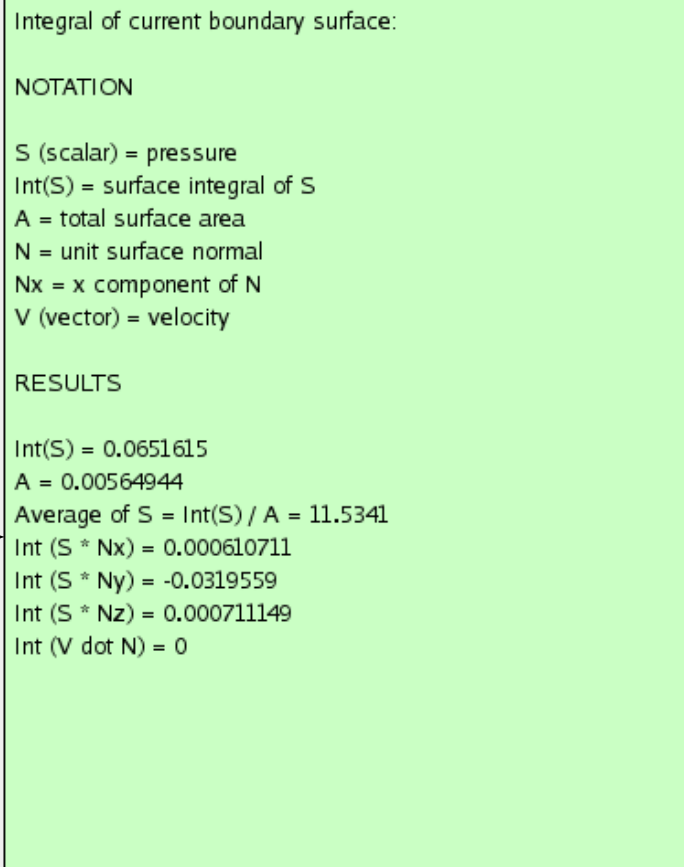
Figure 127 Integration Result Panel with Surface Normals

Surface Normals and the Integration Result Panel

When surface normals are not available, integrating over the surface will yield the integral of a current scalar function over the surface, the surface area and the ratio of these two numbers:

Int(S)	= the total integral (over the surface) of S
Area, A	= the area of the surface
Int(S)/A	= the area weighted average (over the surface) of S

where
  S   = scalar quantity

Surface Normals are available for boundary surfaces where the surface normals are explicitly specified (in a boundary file for PLOT3D format, in the file itself for FV-UNS-combined format, in the grid file for FV-UNS-split format). Surface Normals are implicitly

available for computational, coordinate and iso-surfaces. For all these surfaces additional integration results will appear:

Int(S*Nx)	= integral (over the surface) of S in the x-component direction
Int(S*Ny)	= integral (over the surface) of S in the y-component direction
Int(S*Nz)	= integral (over the surface) of S in the z-component direction
Int(V dot N)	= integral (over the surface) of V in the direction normal to the surface, or
	= volume flow (if current vector is velocity), or
	= mass flow (if current vector is momentum)

...where

S	= scalar quantity
V	= vector quantity
N	= surface normal vector
Nx	= x-component of surface normal, or
	= dot product of surface normal and the unit x vector
Ny	= y-component of surface normal, or
	= dot product of surface normal and the unit y vector
Nz	= z-component of surface normal, or
	= dot product of surface normal and the unit z vector

For coordinate surfaces, the x/y/z component integrals are redundant, and so the output is limited to:

Int(S)	= the total integral (over the surface) of S
Area, A	= the area of the surface
Int(S)/A	= the area weighted average (over the surface) of S
Int(V dot N)	= integral (over the surface) of V in the direction normal to the surface

 "Volume" (vertex-based nodal) data can be used for scalar or vector quantities together with face data in the integrals. The face normal will be multiplied by the "average" value of the nodal quantity. For nodal scalars, the average for the face is calculated as Int(S)/Area, just as it is for the surface as a whole. For nodal vectors, a simple average of the cross products at the nodes is used.

Note: The additional integration results are only available if the Integration Mode is Current Surface, or Partial Surface.

Note: Boundary Surface normals are provided in the data by the solver (or if specified in *.fvbnd* files for Structured Cases.) If FieldView detects that these our out of compliance for *FieldView Unstructured* cases, they will not be used, and FieldView will print the following to the terminal:

```
WARNING: boundary clockness flag in FIELDVIEW Unstructured
file was set, but boundary faces had inconsistent
clockness. Therefore, the boundary face normals will
not be used. Boundary surface integrals with normals
will not be available.
```

Notes regarding Boundary Surface Integrations
Since it is possible for results to exist for some but not all of the boundary faces of a boundary surface being integrated, all faces without face data will be skipped during integration. This applies if the current scalar is a face data function or if the current vector is a face data function. When this occurs, a warning will be issued in the Integration Results pop-up:

```
Warning: Surface results are incomplete. Missing data
skipped.
```

If surface normal direction information only exist for some, but not all of the boundary surface being integrated, then the normal-based integrals will be skipped and the following message will be issued:
```
Note: Normals are incomplete. No normal-based integrals.
```

If surface results are present, but surface normal direction information does not exist for the boundary surface, then normal-based integrals are skipped and the following message will be issued:
```
Note: Normals are absent. No normal-based integrals.
```

Partial Integration

The Partial Integration mode works with only coordinate and iso-surfaces. In some cases, particularly in flows through complex manifolds (**Figure 128 Partial Integration**), it is desirable to obtain an integrated result on only a part of either a coordinate or iso-surface. Partial integration will return a result for a part of a surface, which is selected by specifying a point. This point specification can be done graphically or explicitly using the **FVX** command, `integrate_partial_surface`. A partial surface is calculated within **FieldView** by collecting all of the polygons on that surface that are connected by at least one node. In order to determine whether nodes are actually to be connected on this surface, a Selection Tolerance (node tolerance) parameter is used. The default of 1.0E-05 should be sufficient for most cases, but it can be modified on a case by case basis.

In the graphical mode of operation, the user needs to pick the portion of the surface to be integrated using the CTRL-left mouse (or CTRL-M1) method. Once a surface has been identified, the underlying mesh is highlighted to indicate what area will actually be integrated. If the graphical selection of the surface is incorrect, the user can make a different selection before the integration calculation actually takes place. In cases where the desired partial surface is very close to other sections of the surface, Selection Tolerance (node tolerance) may be adjusted to refine the partial surface. Two pieces of the surface are assumed to be connected to each other if they have nodes that are closer to each other than the Selection Tolerance.

Pressing the Integrate button on the Integration Controls panel produces the integration results. Coordinates of the selected point are shown in the Integration Results section, in addition to integration results. Note that selection of a surface for partial integration will make that surface the current one.

Thresholding can be also applied to a surface that has sections which will subsequently be selected for integration. The surface highlight will always show the area which is to be integrated. In the example below, integrated results can be obtained for each of the three sections of the manifold.
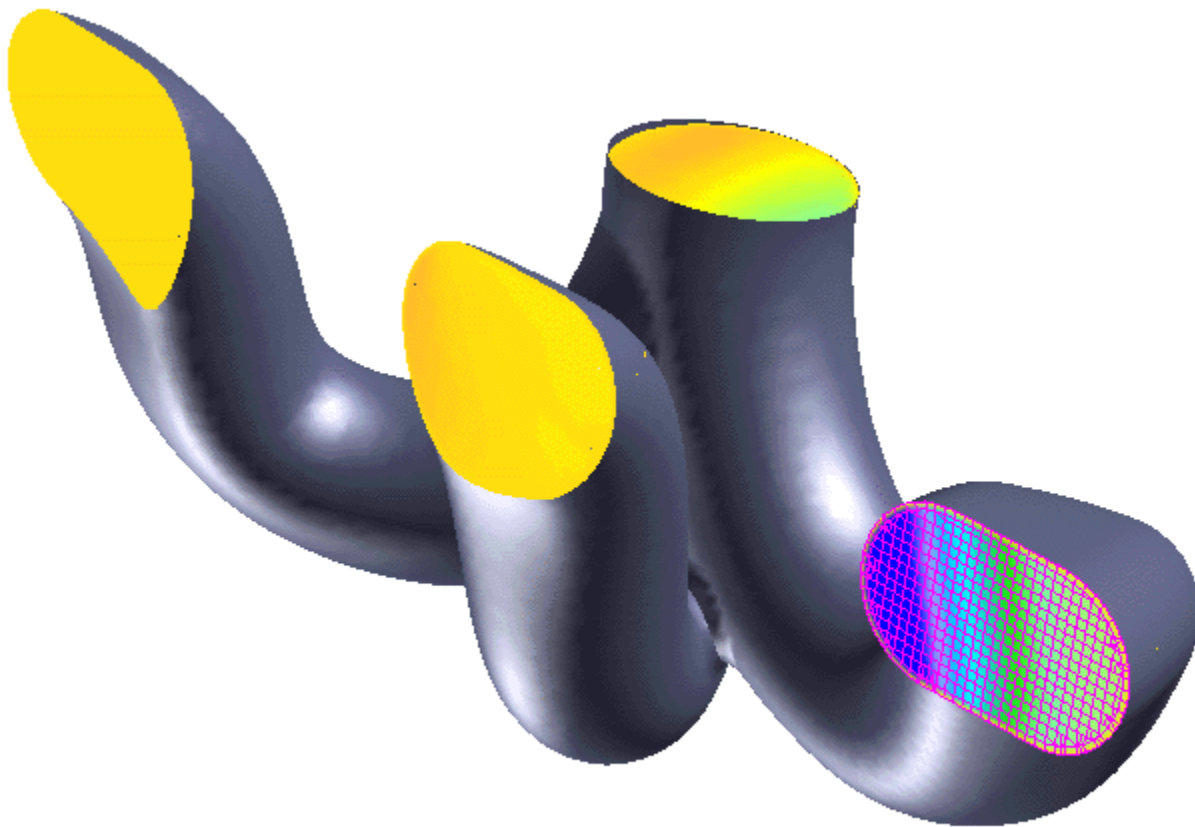
Figure 128 Partial Integration

See **Chapter 4** of this manual for details on `integrate_partial_surface` **FVX** command.

Frequently Asked Questions

How do I calculate the integral of a function over a single surface or group of surfaces that are displayed on the screen?

If it is not already present, bring up the Integration Controls Panel. Press the Integrate button and the surface integral of the current Scalar Function will be calculated over all valid surfaces displayed in the current dataset. Valid surface types are: Computational Surfaces, Iso-Surfaces, Coordinate Surfaces and Boundary Surfaces. However, Boundary Surfaces with a Display Type of Outlines will not be used during the integration calculation. The name of the current Scalar Function used for the integration will be displayed near the top of the panel. A pop-up will appear that displays the value of the integral as well as the area of the surface(s) and the average function value over the surface(s).

Where is the output of the integral calculations displayed?
If the integration is calculated over all surfaces, the results of the integration will be displayed in a pop-up window. If the integration was calculated for a sweeping surface, the results will be placed in the specified output file. Note that you do not have to dismiss the pop-up after each integrate step; the value in the displayed pop-up will be updated. If you wish, you can move the pop-up to another, more convenient screen location and leave it there for all of your integration work.

Will streamlines, particle paths, point markers, Vortex Cores / Surface Flows or titles affect the integration result?
No, they are ignored by **FieldView** during integration.

What if I have two surfaces displayed, each with a different scalar?
Remember that **FieldView** integrates the current Scalar Function over all surfaces in the current dataset. When Integrate is pressed the Current Scalar will be computed for all surfaces for the purpose of integration. Each surface will then be reset to the function it held prior to the integration operation.

Can I use Thresholding or Range Subsetting to affect integration?
Yes. Whatever you have set for each surface will be used during the integration process. Note that setting the Increment greater than 1 may cause a loss of accuracy, as a more coarse polygon representation will be used. If this is the case, a warning pop-up will be issued prior to calculating the integral. At this point you may either continue with the integration, or cancel.

What is the Integration method being used?
The integration is driven by the list of surfaces currently displayed by **FieldView**. These surfaces are created and modified using the standard Create/Delete and modify operations on the Visualization Panels (Restart Files may also be used). When processing the list of surfaces for integration, each surface is broken down into a series of polygons. The numerical integral is computed as follows:

```
Set Integral Value = 0
FOR each Valid Surface[1,2]
      Decompose into Polygons[3]
      FOR each Polygon
```

Notes on the Integration method
1. A Valid Surface is a Computational, Coordinate, Iso- or Boundary Surface that has been created and whose Visibility is ON.
2. In the default mode ("All Surfaces in Dataset"), all valid surfaces in the current dataset are subject to integration. Otherwise, only the current valid surface is considered - all others are ignored.
3. Surfaces are decomposed into triangles and quadrilaterals.
4. In the case of triangles, the integral is computed by taking the average of the function's value at the nodes and multiplying by the area. In the case of quadrilaterals, a Gaussian technique is used to increase accuracy.

Note that the actual Surface Type (Geometric, Scalar, Vector), Display Type (Mesh, Surface, Contour, etc.) and Line Type are ignored during the integration process. Also, no modification is made to the graphic image by the integration process.

The integration is always performed using the Current Scalar Function in the Scalar Function register. If there is more than one valid surface in the display, a single Scalar Function is used for integration across all of the valid surfaces: the Current Scalar is recomputed for each of the valid surfaces prior to integration.

How can I calculate the integral as I sweep a surface?
First, make sure that the surface that you want to sweep is the current surface and that its Visibility is set ON and Highlight and Sweep are turned OFF (you can use Highlight to determine whether or not the desired surface is the current surface). Next, you need to set the bounds and number of steps for the sweep operation from the Visualization Panel that created the current surface. An example is provided below for iso-surface sweeping (**Figure 129 Iso-surface Sweep example**).
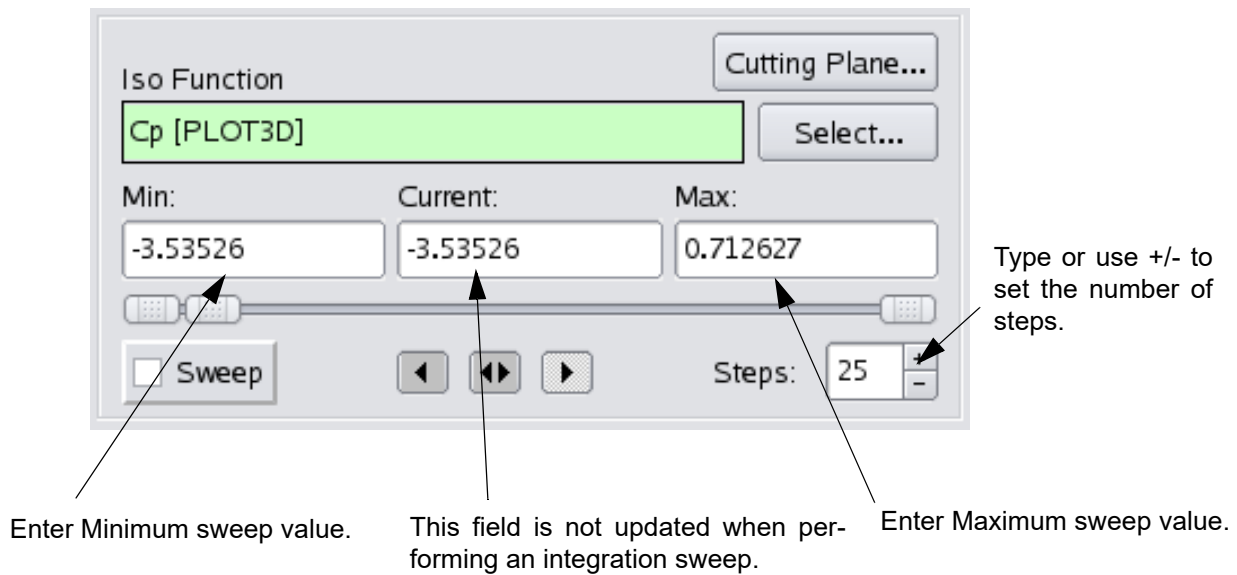
Figure 129 Iso-surface Sweep example

Since the output will go to a file, you need to use the Sweep Integration Output File section of the Integration Controls panel to select an existing file or to name a new one. Finally, press the Integrate button to perform the calculation. There is no feedback during the computation, except for the **FieldView** "Working" message that may appear.

An ASCII file will be created to record the results from each sweep step. Each time the Integrate push-button is pressed, a header is written to the output file. Then for each sweep step, a line is added to the file containing two real numbers: the sweep step value and the results of the integration at that step.

When the Integrate button is pressed, **FieldView** looks at the file name and directory path on the Integration Controls panel and attempts to open the file. If the file already exists, you are given the option to Append to the Existing file, or Overwrite it.

The file format is:

1. A Header, consisting of 2 lines

    The first line will identify the surface type being swept:
        Computational Surface: I, J or K
        Coordinate Surface: X, Y or Z
        Iso-surface: Name of the Iso-Surface Function (e.g., Temperature)

    The second line will identify the Scalar Function being integrated:

Integral of Fname (where Fname is the Function name, e.g., Pressure)

2. Data, one line per sweep step

Each line will contain two floating point values, the Sweep Step and the Integral.

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ Coordinate Surface labels (and values) become RTZ (Radius, Theta, Z) labels and values. The direction of the sweep will then be in R, T, or Z space. See **Chapter 3** for more information.

I have data in multiple datasets and I want to perform sweep integration of an Iso-surface through all datasets at the same time. How do I do it?
At first glance, this seems to be a problem, since Iso-surfaces (and in fact all **FieldView** surfaces) are limited to a single dataset. Since only the current surface can be swept, only the portion of the iso-surface that is in the current dataset will be swept. However, there are two ways to accomplish the goal.

The first way involves restructuring your data outside of **FieldView**. If you are dealing with a multi-block problem and it is split into multiple datasets, combine the results into a single dataset, with each block or region in a separate grid. Then, iso-surfaces and coordinate surfaces will be evaluated in the entire volume and sweeping will work just fine.

The second approach does not require changes to the data files, but requires a separate sweep operation for each dataset. You can use a Restart file to create the multi-part iso-surface, but you will have to manually cycle through and integrate each dataset. The integral values for all of the datasets will have to be summed up outside of **FieldView** to get the total. You might want to use Append mode on the output file to place the data for all datasets together in the same file.

What happens if some of my surfaces are displayed as Geometric or Vector type or Transparency is not equal to 0?
Integration is not affected by Surface Type (Geometric, Scalar, Vector), Display Type (Mesh, Surface, Contour, etc.), Line Type or Transparency. If the surface has been created and not removed from the screen, it will be used for integration. Similarly, Integration will be performed even if the Dataset or Region has its Visibility turned off. An exception is that existing Surfaces are skipped if Visibility is turned off.

Possible Problems
The integration result is displayed as something other than a number (++, --, ?, +Inf, -Inf, NaN, etc.). What does this mean?
During the integration step, **FieldView** came up with a result for the integration value that was an invalid floating point number. This can occur if the Scalar Function being integrated has one or more invalid values (e.g. Infinity).

```
Integration Confirmation: There are surfaces which have a subset
increment > 1. The integration result will be less accurate. You
must press Confirm to continue with the integration.
```
When a subset increment is set to a value greater than one, a coarse representation of
the surface is passed to the integrator. This leads to a less accurate result. If this is not
acceptable, press Cancel. Then find out which surface or surfaces are set this way and
return the increment to one.

```
Integration Confirmation: About to calculate the integral of
<function>.  Not all surfaces are colored by <function>. The cal-
culation will take extra time. Press confirm to continue.
```
This message is displayed when **FieldView** is about to re-calculate the current Scalar
Function for some of the displayed surfaces because they were calculated with scalar
functions different from what is currently loaded in the Scalar register, which is being
used as the basis for the integration. Note: This will not affect the display of any of the
surfaces, but will take extra time to calculate. If this is what you want to have happen,
press Confirm. If it is not, press Cancel.

Error Conditions

```
Integration Error: Sweep Current/Current Surface: No current sur-
face to sweep.
Integration Error: Sweep Current/Current Surface: No current sur-
face to integrate.
Integration Error: Sweep Current/Current Surface: Not available
for Boundary Surfaces.
```
You attempted to Integrate in "Sweep Current Surface" or "Current Surface" Mode at a
time when there was no current Computational, Coordinate, or Iso-surface available. If a
valid surface already exists, you must bring up its panel, and make the valid surface cur-
rent.

```
Integration Error: Sweep Current Surface: An output file must be
specified
```
A valid file name has not been entered into the File Selector on the Integration Controls
panel prior to pressing the Integrate button. Enter or select a file and press Integrate
again.

```
Integration Error: Sweep Current Surface: Sweep integration not
available for this panel.
```
You attempted to do a Sweep when the current surface was not a Computational, Coor-
dinate or Iso-Surface. These are the only valid surface types for Sweep Integration. This
could appear if the current surface was the Titles Panel, for instance.

```
Integration Error: Sweep Current Surface: Current surface has
visibility turned off. Toggle it on.
```

The selected current surface for sweeping has Visibility set to OFF. You must turn it on in order to do the integration.

```
Integration Error: All  surfaces: There are no valid surfaces to
integrate.
```
If you have already created any surface in **FieldView**, check the visibility to ensure that it is on. Otherwise, you will need to create one or more valid surfaces before performing the integration.

```
Integration Error: Current surface being highlighted (swept).
Highlighting (Sweeping) must be turned off.
```
Integration cannot be performed while either Highlight or Sweep are on for any of the valid surfaces. This must be turned off for all displayed surfaces prior to pressing Integrate.

Important Note: In previous versions of **FieldView** the faces of interior boundaries would be included twice for integration calculations. This should no longer be the case. FV-UNS files in ASCII format (for all versions), or in binary/ unformatted version 2.2 and earlier, would allow the vertices of quadrilateral faces to be given in arbitrary order. The faces would then be matched up to the element or elements they were attached to and the vertices were taken from the element - not the face - for final boundary faces in **FieldView**. Now that **FieldView** separates faces from elements, files that previously read without errors can now result in degenerate boundary faces. Typically they will resemble a bow-tie when displayed with Mesh. Switching the order of the last two vertices for any degenerate quad faces is all that is required to fix them.

Flipbook Controls

Flipbooks are recorded movies that may be saved to disk, and replayed at any time without requiring the data files. These flipbooks can be used to record the sweeping of surfaces, streamlines, rotation of the model, or a visualization of transient data, or a keyframe animation. They can be created interactively or from a script. For a complete description of animation capabilities within **FieldView**, see **Chapter 6** of the **Reference Manual**.

Note: See the Tools, Graphics Layout Size sub-menu (**Figure 264**) for a way of easily setting the graphics window sizes for use during Flipbook and Keyframe animation creation.

Note: It is possible to create flipbook animations even if the graphics window is partially covered by another application, or, if the graphics window has been minimized.

When Flipbook Build Mode is on, all Visualization Panels that contain a Sweep/Animate button will have that button replaced by a button that says Build. When the Build button on any panel is pressed, the surface (or streamlines) will be swept one time from the current setting, with all of the frames from the sweep or animation cycle saved in a flipbook.

Once a flipbook has been created, turning on the Play / Pause button will continue to play the flipbook until the button is turned to Pause. Note: All other panels will be disabled while a flipbook is playing.
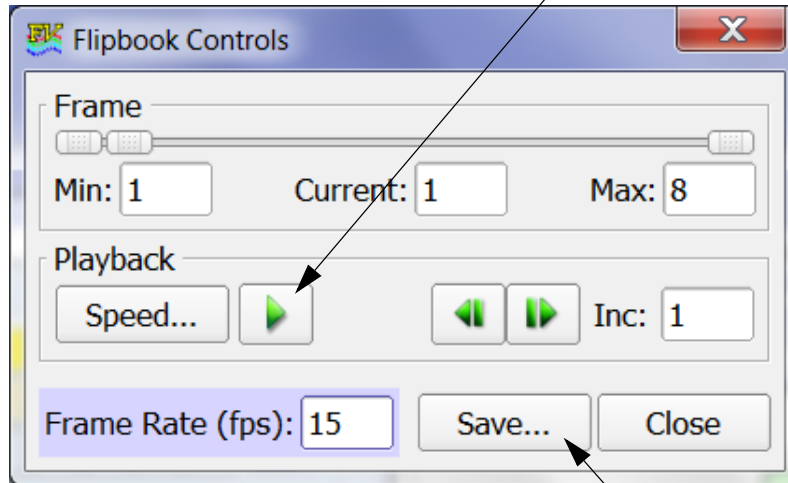
Once the flipbook is built, the Playback may be too fast; to slow it down, click the Speed button and increase the Minimum Time Between Frames. Frame Rate (fps) applies only when saving a flipbook in MP4 format; the default, 15 (frames per second), may be adjusted to any integer from 1 to 60.

Pressing the Save button will bring up a browser (see **Figure 131 Flipbook File Save Panel** below) giving you format options (see **Output Formats** in the **Reference Manual** for details) and allowing you to choose a directory in which to save the file. The frames from Min to Max will be saved.

Figure 130 Flipbook Controls Panel
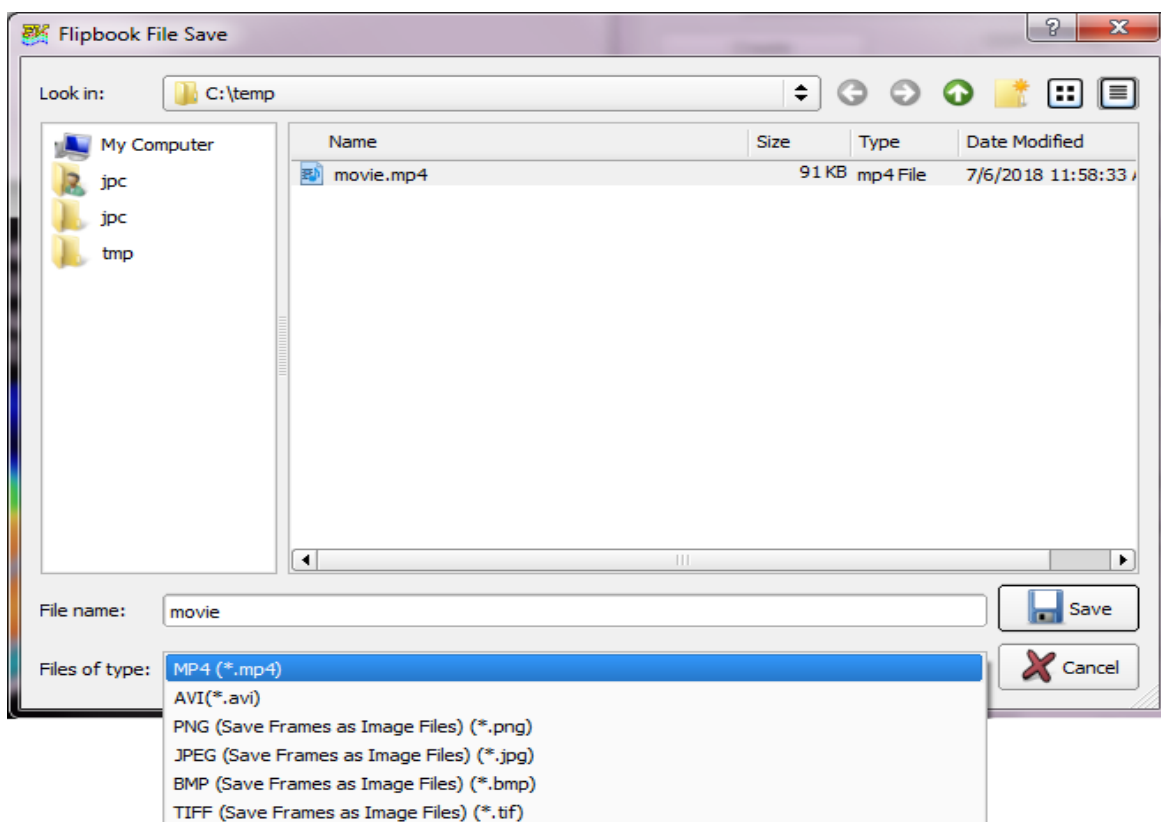
Figure 131 Flipbook File Save Panel

Using the Flipbook Controls Panel

Why use Flipbooks?

Flipbooks are easy ways to save out a sequence of images and play them back later as a movie. In this way, the movie may be played at any time without requiring the original data to be present. Flipbooks are portable and can be played on a different platform than the one from which they are created.

Flipbooks can be used to show a sequence of images quickly, that would otherwise take a long period of time to generate. A good example of this would be a large, transient model that may take several seconds to generate each frame. By saving each frame into a flipbook, the sequence can be played in near real time.

How do I create Flipbooks?

The Flipbook Controls panel is used to create flipbooks of a single sweep (or animate, in the case of streamlines or particle paths). This allows sweeping of cutting planes, iso-surfaces or even time, in transient datasets. More complex flipbooks can be generated using the **FieldView** Scripting language (see **Chapter 5** of the **Reference Manual** for

more information), or by creating a Keyframe Animation (see **Chapter 6** of the **Reference Manual**).

Turn on Flipbook Build Mode from the Tools pull-down menu and dismiss the Flipbook Size Warning panel that comes up. This will cause the Sweep (or Animate) button on the visualization or Transient Control panel to be replaced by a button that says Build. Pressing the Build button on a visualization or Transient Control panel will cause the surface, streamlines or time to be swept one cycle from its current position. Each frame of the sweep will be saved to a flipbook as it is drawn. The Build can be aborted at any time by clicking Stop on the Building Flipbook panel. Once the build is complete, the sweep will be turned off and the Flipbook Controls panel (**Figure 130**) will come up. In order to save the flipbook, press the Save button. After entering in a filename and directory in which to place the flipbook file, **FieldView** will save the flipbook by playing it once from start to finish. If you interrupt this play, you will interrupt the saving process.

Note: The Tools, Graphics Layout Size sub-menu (**Figure 264**) provides a way of easily setting the graphics window sizes for use during Flipbook and Keyframe animation creation. The list offers several standard image sizes. Choosing one of these predefined sizes will cause the graphics window to be resized.

Please Note: When creating a flipbook, it is very important not to resize or move the graphics window, place any other windows over it, or touch any of the controls on the visualization panels. If you wish to make the graphics window a different size, you should make the change before creating the flipbooks.

How do I play a Flipbook?
Once a flipbook has been created, you may play it any number of times by pressing the Play button on the Flipbook Controls panel. While the flipbook is playing, all other controls on the panel will be disabled until the Play button is turned off.

If you wish to build another flipbook, you must press Close. Pressing Close will bring up a pop-up warning that the current flipbook has not been saved. Once you Close without saving, the flipbook must be built again from the beginning. If you wish to use the flipbook again, save it in a disk file.

How do I play a Flipbook that has been saved to disk?
If you have created and saved a flipbook to disk, you may play the flipbook without starting FieldView by using your default movie player on Windows, or other movie players such as VLC media player, or ImageMagick's *animate* command which is available with many Linux installations. Note that there is no capability within FieldView to read in a flipbook that has been saved.

How can I control the speed of the playback?
If the flipbook playback speed is too fast, you may slow it down by using the Frame Rate controls on the Flipbook Controls panel. This control will allow you to add a maximum amount of time (up to 1 second) in between the display of each frame. Note this will not

affect the speed of playback outside of **FieldView**. If you output MP4 format animations, you can embed frame-rate using the setting *Frame Rate (fps):* option on the Flipbook Controls panel. For other movie formats, your movie player may offer such control.

What formats does the Flipbook support?
Pressing the Save button on the Flipbook Controls panel will cause the Flipbook to be output to disk in one of several formats, the recommended format being .mp4, which allows frame rate to be embedded in the file.

Why can't I adjust a visualization panel while the Flipbook panel is up?
Once a Flipbook has been created, all other panels will become disabled. This is due to the fact that the graphics window is now being used to display the flipbook. In order to re-activate the other panels, you must delete the current flipbook from memory.

What effect does the graphics window size have on flipbook performance?
Each flipbook frame requires storage. The amount of storage required within **FieldView** depends on the size of each image and the number of frames. Smaller flipbook images therefore have the following benefits: they play faster and allow you to create longer sequences within a given amount of swap space. To create smaller flipbook images, resize the graphics window before starting the flipbook build.

2D Plot Controls

The 2D Plot Controls panel is used to make two-dimensional plots (graphs) of the current scalar function on a straight line in X, Y, Z space, or along a curve on a particular surface for the current dataset.

Press Create on the Plot tab of the 2D Plot Controls panel to create a 2D Plot.

Switch to the Paths tab to be able to create a path on a 2D plot.

Use PLOT SELECTION on the 2D Plot Controls panel to navigate between 2D Plots. The maximum number of plots is 9.

Parameters associated with Annotation, Left, Right and Horizontal Axes, Sampling and Border control the appearance of each specific 2D Plot, and are applied on a per 2D Plot basis.

Press Create on the Paths tab to create a path within a 2D Plot.

Show Path will display the permanent markers in the graphics window showing the points used to create the plot.

Show Plot will display or embed the 2D Plot in the graphics window.

When Right Axis is pressed, the variable loaded into the Thresholding register will be used as the right-hand axis of the 2D plot.

Pressing the Create button allows creation of Plot Paths of the type: Line Path, Open Curve Path, Closed Curve Path, XYZ Closed Curve Plot Path, and XYZ Curve Plot Path on the current dataset only. The maximum number of Plot Paths per 2D Plot is 10.

The Path Style option menu can be used to select how the paths in the plot should be displayed. Options are: Lines, Lines and Symbols, and Symbols.

Horizontal Axis/Plotting Direction is applied to all Plot Paths within a specific 2D Plot. This will change whether the plot is shown as a function of distance, or X, Y or Z. Note that for line plots, selecting X, Y, or Z will constrain the location of the endpoints.

Import is used to import the 2D plotting information from an external file including Point Probe output and 2D Plot Export. The Export button writes out 2D plot information for only the current plot path to an external file.

Figure 132 2D Plot Controls and 2D Plot Paths Tab

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the Horizontal Axis/Plotting Direction XYZ buttons become RTZ (Radius, Theta, Z) buttons. See **Chapter 3** of the Reference Manual and **Chapter 12** for more information.

FVX File Execute and Convert Script Panels

The **FVX** File Execute panel is used to open and execute **FieldView** Extension language program files. To bring up the **FVX** File Execute panel, go to the Tools menu and select Execute **FVX** File... This panel exhibits the features found in all of the file selector panels encountered in **FieldView**. Files of type `fvx` are listed by default. To execute the **FVX** program, select and open it. Program execution begins immediately. **FVX** does not require file extensions to be `.fvx`.

The Convert Script panel is used to convert **FieldView** script files to **FVX** files. To bring up the Convert Script panel, go to the Tools menu and select Convert Script to **FVX**... This panel exhibits the features found in all of the file selector panels encountered in **FieldView**. Files of type `scr` are listed by default. To convert the **FieldView** script to Extension language, select and open the specific file. The Convert Script panel has the same layout and functionality as the **FVX** File Execute panel.

Displays the current Files available. File read action can be accomplished by double clicking on filenames or clicking once and using the OK button.

This area displays the current Directories available. It also allows the user to traverse up and down the current directory structure.

To specify the file name, type directly into the entry box, followed by a carriage return.
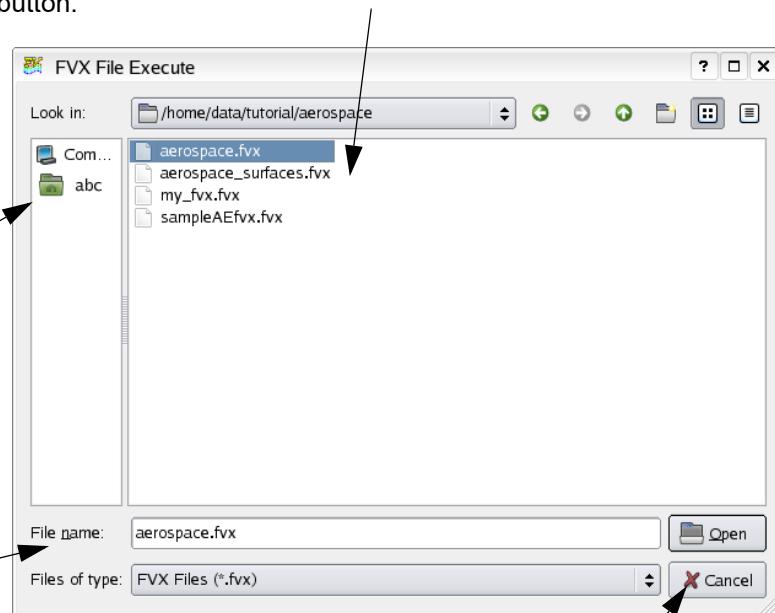
Press this button to Exit the file selection panel. No other action will be taken.

Figure 133 FVX File Execute Panel

Extension language File Debugger Panel

The Extension language File Debugger Panel is the `console window` from which **FieldView** was launched. It is the means by which **FieldView** Extension language (**FVX**) programs provide text feedback. Any **FVX** program that is executed can be monitored from this panel. This panel is the standard input/output interface for **FVX** programs. For example, the output from the `print` command is displayed through this panel. If an error is present in an **FVX** program, this panel provides a prompt to facilitate program debugging. **Figure 134 FVX Debugger Window** shows a panel as it appears when **FieldView** is started. See **Chapter 4** of the **Reference Manual** for detailed information on the debugger.
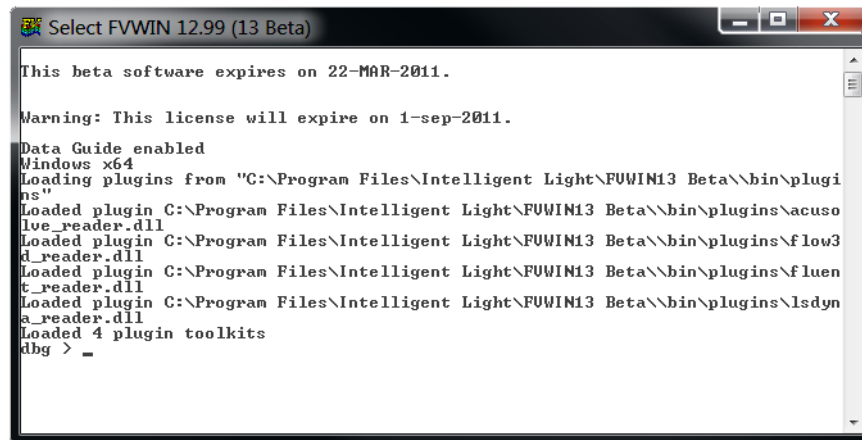
Figure 134 FVX Debugger Window

Chapter 4

Computational Surface

4

This panel is used to visualize data on the computational mesh. A computational grid is a set of points in IJK space, with XYZ position values defined for each point. A Computational Surface is a plane in IJK space (that is, a surface of constant I or J or K in a single grid) which may be curved in XYZ space. This panel is only available for structured grids. For buttons and features not mentioned below, please refer to **Chapter 3**. See (**Figure 135**) below for a discussion about the Computational Surface Panel.

Exporting: Computational surfaces can be exported using the Tools/Export command. For information on the format, please see **Appendix I** of the **Reference Manual**.

Computational Surface Panel

The Surface Plot is used to create 2D plots on the current surface. See the section on Surface Plot for more information.

The Surface Probe is used to probe individual I, J, and/or K points on the current surface. See the section on Surface Probe for more information.

Shows the locations and provides labeling of the minimum and/or maximum values of a scalar on the given surface. The Options button will produce a panel for further control of this feature, see **Scalar min/max Sub-panel**..

Surface Selection defines the current value for the Computational Surface. After pressing Create, I, J, or K must be chosen to define the surface. For the dimension that is chosen, the location of the surface can be controlled by either moving the slider or typing into the Current field. The Min, Max and Inc fields will control the bounds of the animation for the surface.

For the dimensions that are not chosen, the Min and Max can be adjusted to clip the surface in either direction. Modifying the Inc field will change the coarseness of the display. For example, setting the Inc field to 2 will show every other grid point for the direction chosen. This can be used to increase performance and decrease the level of detail of the surface. The Current field has no effect for the non-selected dimensions.

The Grid Selection controls the grid in which the current surface belongs. To change the grid click on the arrow keys or type the grid number in the menu.
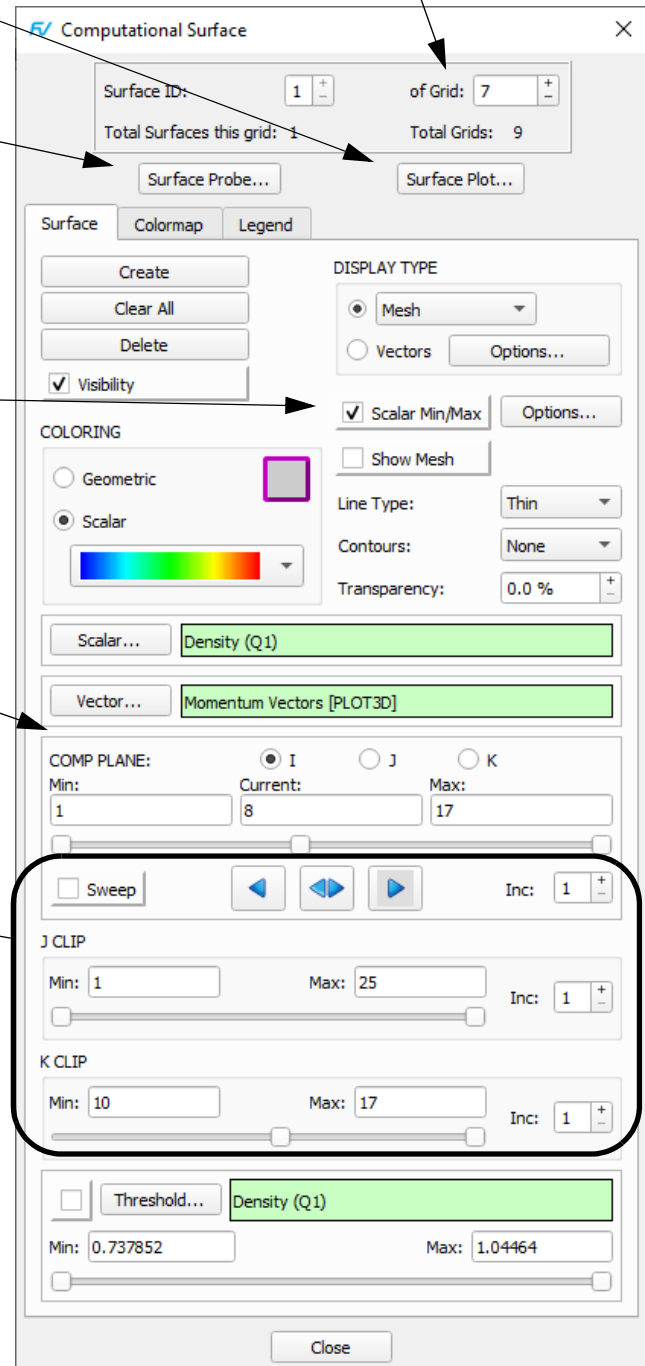
Figure 135 Computational Surface Panel

Frequently Asked Questions

How do I create a Computational Surface?

Bring up the Computational Surface Panel. In order to create a surface, press the Create button. The first time it is pressed, an I surface will be created, with its value at half the distance between the minimum and maximum I. (Note: If the dataset is 2D, the direction that only has a single plane will be created by default). If an I surface is not desired, simply press the button marked J or K to change the surface. If a second surface is desired, again first press the Create button. Now, a copy of the previous surface will be created. To change this surface, either move the appropriate slider bar to a different value, or choose a different type of surface to display.

How can I animate the surface?

To animate the current surface, press the button marked Sweep in the Sweep Control area. By default, the animation will move forward and start again at the minimum after it reaches the maximum. You may also specify the animation to be backward or to bounce between the two by pressing on the arrow keys in the Sweep Control area. Pressing on these arrow keys will not only change the direction of the animation, but will also move the plane one step in that direction.

To increase the speed on the animation, increase the number in the Inc. area for the current surface. If the Inc. field is set to 2, the animation will show every other plane between the specified minimum and maximum.

What are my display options for the surface?

A computational surface may be colored by a Geometric or Scalar color. The DISPLAY TYPE pop-up menu allows it to be shown as a constant, faceted or smooth shaded surface, a mesh, contour lines, vertices or shaded vertices. An alternative setting displays Vectors and has a button to control Options. Mesh and contour lines may be shown in conjunction with any display type including Vectors.

How do I use Multiple grids?

If you have read-in a Multiple grid file, you must first select which grid you want the surface on before pressing the Create button. Surfaces *of this type* will not go across multiple grids unless an I, J, or K Iso-Surface is used. See **Chapter 5** for more information.

How can I clip the current surface to only show a particular area?

The Computational Surface that is created will show the full bounds in the two non-selected dimensions. For example, an I surface shows all J and K points for the I value selected. If you would like to clip the plane in one of the other directions, you simply move the minimum or maximum slider (or type in to the appropriate field) for the dimension desired.

How can I make the grid more coarse?

If for the current surface, there are too many grid points being shown on the plane, you can subset the surface. This is accomplished by incrementing the Inc. fields for the two non-selected dimensions. This will cause the plane to be shown using a subset of the grids points. For example, if you have an I plane, and you increment the Inc. field for the J dimension to 2, the I plane will be shown using every other J grid point.

How do I use this panel with more than one dataset?
Each dataset has its own surfaces defined. They are kept separate from other datasets. That is, if you have two datasets in memory, and each of them has a computational surface created, then both computational surfaces would be numbered 1. You can only switch between the two computational surfaces by switching datasets. You can do this on the Dataset Controls panel or by "quick-picking" on one or the other computational surfaces.

Surface Probe Data Window

Note: Function values which are floating point NaN's (not-a-number) are displayed as Nan in the FN VALUE field of the Surface Probe Data section of the Surface Probe Panel.

Using the Surface Probe

What is the Surface Probe?
The Surface Probe panel (Figure 136) allows you to request information about specific points on the current computational surface. The information is displayed in a text box within the panel. In addition, you can request that the location of the point be shown in the graphics window by turning on Display Point.

How do I probe a specific point?
To probe a specific I, J, or K point, move the respective sliders to the appropriate position, or type the desired value into the value field on the Surface Probe panel. The I, J, K, X, Y, Z and Scalar value of the probed point will be displayed on this panel.

How does this differ from the Point Probe?
The Surface Probe can be used to probe a specific I, J, K grid point for structured grids only. The Point Probe can be used for both structured and unstructured grids, and probes intermediate points in X, Y, Z space, not just the vertices.

The location of the point to be probed may be changed by typing a new Value, by clicking on the left and right arrow buttons; or by moving the slider.

The Display Point displays the location of the selected point in the graphics window. When in Presentation Rendering, the point will be light source shaded.

The Surface Probe Data displays the I, J, K, IBlank, X, Y, Z, values and the current scalar function and value. The GRID number will be the grid's number for the current dataset. There may be other GRIDS numbered 1 which belong to other datasets in memory.
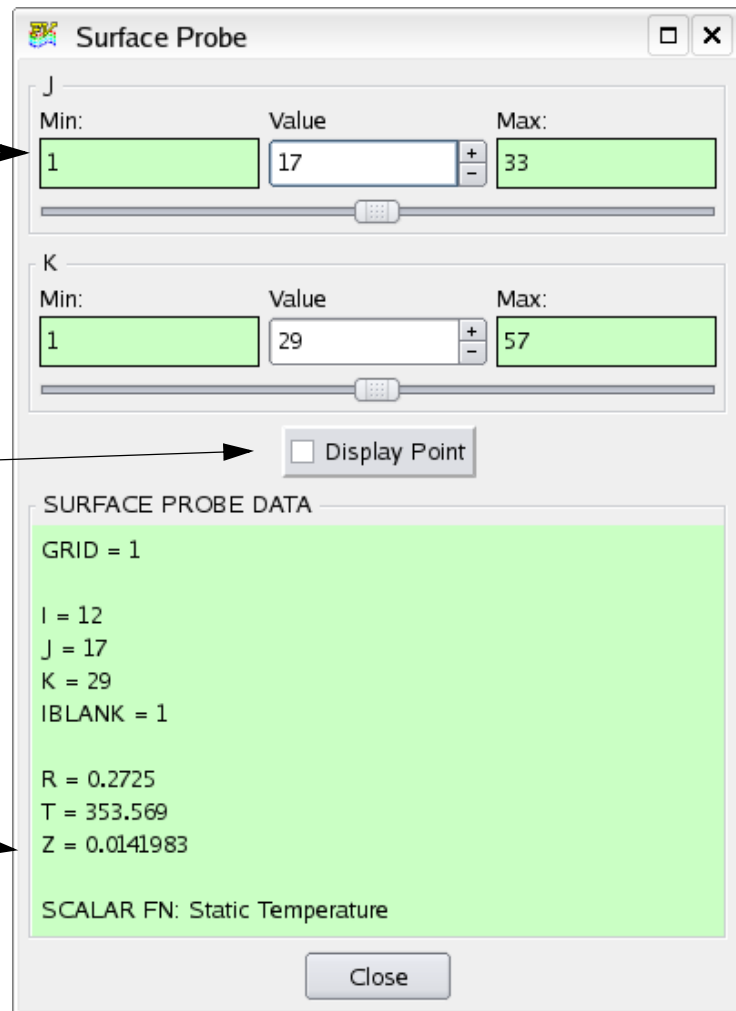
Figure 136   Surface Probe Panel

Surface Plot

The Surface Plot panel (**Figure 137**) is used to make two dimensional plots (graphs) of the current scalar function on a particular Computational Surface. To create two dimensional plots on surfaces other than Computational Surfaces, the 2-D Plots panel should be used.

FieldView Surface Plots are triggered from the Computational Surface panel and replace the Computational Surface panel in the Visualization panel area. The Surface Plot button is unavailable (disabled) in the Computational Surface panel if there is no current computational surface. If no results data has been loaded for the current grid the Surface Plot panel is unavailable and an error message will appear.

Surface Plot Controls Panel Definition Tab

The Slice area is used to designate which axis on the current computational surface will be used to create the Surface plot. The direction used to define the current surface may not be selected here.

This area controls the location of the plot for Single plots, and the minimum, maximum and number of lines drawn for Carpet plots.

The Annmotation tab controls the name and the min and max of the axis.

The Title and Subtitle text that will appear above the plot may be typed into these locations.

Show Path turns on a series of markers in the graphics window, showing the location of the plotted points.

Show Plot will display or embed the 2D Plot in the graphics window

The Plot Type may be either Single, meaning one cut through the surface, or Carpet, meaning a series of cuts.

The Horizontal Axis of the plot may be either default (I, J, or K), or X, Y, or Z depending of this selection.

The Plot Style controls whether the plot is displayed as a line, as symbols or both.

To Zoom in on an area in the surface plot press and hold M1 while in the Line Plot window. To zoom back out, press the Unzoom button.

The Export button will write out all Surface Plot information to an external file specified by the user in the Surface Plot File panel. This Export is independent of the Tools…Export pull-down menu option which is intended for current surface exporting only. See **Appendix I** of this **Reference Manual** for Export formats.

Figure 137   Surface Plot Controls Panel

Embedded Surface Plot

To display or embed the surface plot within the graphics window, use the Display Plot button on the Surface Plot Controls Panel, in the GRAPHICS WINDOW section.

By default, the plot will have a transparent background. The Surface Plot sits above all other graphical elements within the graphics window. If you want the plot background to cover everything else underneath it, turn on the Background button.

The color of the plot lines will be the same as the geometric color for the Computational Surface that the Surface Plot is based on. To change the color of the lines in the plot, you will need to change the geometric color of the surface. At this time, there is no explicit control over line color for the embedded surface plot.

There is a limit of one embedded plot per dataset. If a 2D plot has been created, and has already been placed within the graphics window, you will receive the message shown in the figure below.

Figure 138   Surface Plot Display Embedded Plot Warning

To overcome this limitation, a dataset can be re-read into **FieldView**, and additional 2D or Surface plots can be generated for each dataset. Since **FieldView** maps to appended datasets which are already read into memory, when they are recognized as identical, little additional memory usage penalty is incurred by using this work-around.

The embedded plot is designed to be in synchronization with the Surface Plot window. For instance, if you zoom into the Surface Plot window, the embedded plot within the graphics window will update at the same ime. If the dataset is transient, and you are sweeping through time, the embedded plot will update for each time step as the data changes.

Embedded plots can be scaled and translated using the Transform Controls by setting the object from the normal default Object: World to Object: Plot. The left mouse button is used to translate the embedded plot, and the right mouse button is used to control the plot size within the graphics window. There are limits to the location of the plot within the screen, and on the minimum and maximum plot size.

Surface Plot Controls Annotation Tab

The Horiz Label is used to change the default label for the horizontal axis.

The Horiz Min & Max controls the minimum and maximum values to be plotted horizontally.

The Vert Label is used to change the label for the vertical axis from the current scalar function name, which is the default.

The Vert Min & Max controls the minimum and maximum values to be plotted vertically.

The Setup controls whether the Date, Plane and Grid Information, or Legend are displayed in the Surface Plot Window.

Figure 139   Surface Plot Parameters Panel

Using the Surface Plot Panel

How do I create a Plot?
When the Surface Plot panel is invoked, a default plot will appear. Since this plot is on the current Computational Surface (of the specified dimension), you must supply the direction to slice this surface in order to create the plot. This dimension must, of course, be different from the dimension used to create the surface. The third, not selected dimension, will be the default horizontal axis. For example, if you have an I plane, and you choose to slice it in J, the horizontal axis of the plot will be K.

How can I control where I am plotting?
To control the specific slice where you wish to plot, either move the slider on the Surface Plot panel, or type a specific slice into the slice type-in field. For carpet plots, you can control the minimum and maximum slice, as well as the number of slices by modifying the min, max and slice type in fields, or moving the sliders.

How can I see where I am plotting?

The plots you create will follow lines of I, J or K. In order to see which grid line the current plot is following, press the Display Points button.

How do I control the axis of my plot?
The default axis is determined based on the slicing axis as defined above. However, if you wish instead to have the horizontal axis be X, Y or Z, select the desired button from the Horizontal Axis option menu. The Axis will be defined with default minimum, maximum and names. To change any of these values, type the desired entry into the Surface Plot Parameters panel.

What is the difference between a surface plot and a line plot?
A Surface Plot is defined along lines of I, J, and K and is invoked from the Computational Surface panel. A line plot is a straight line in X, Y, Z space that can be defined using the 2-D Plots panel (see **Chapter 12** for more information.)

How can I zoom in on a particular area of my plot?
By moving the cursor into the Plot Display window, and pressing and holding M1, you will create a rubber band box that can be used to define a specific area of your plot that you wish to zoom in on. This is similar to the Zoom box on the Transform Controls panel. When you zoom in, the axis will automatically adjust to show the new range of values. To reset back to the default view, press the Unzoom button on the Surface Plot panel.

What is the difference between a Single and Carpet Plot?
A single plot shows the scalar function values along a single slice through the current Computational Surface. A carpet plot shows multiple slices through the current Computational Surface, with the spacing between slices determined by the Inc.: value in the Slicing Area.
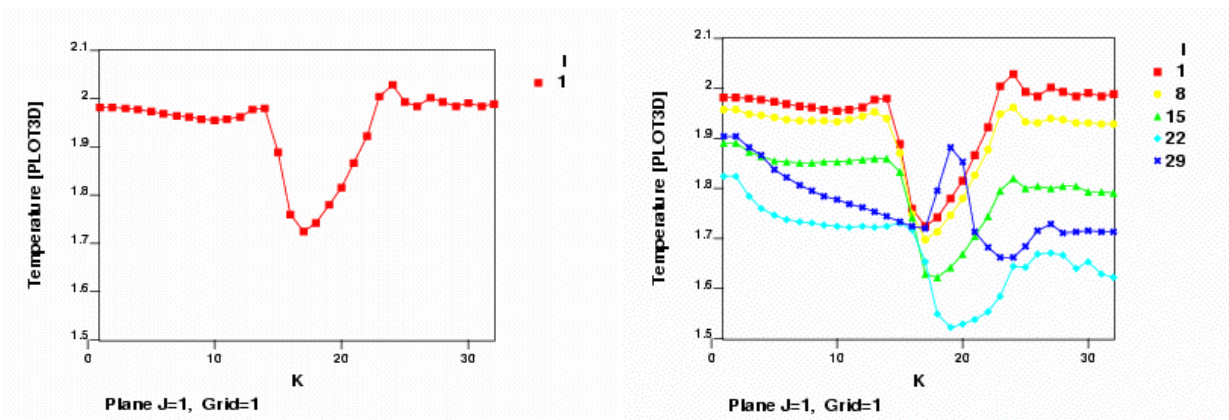
Figure 140   Single Plot vs. Carpet Examples

Chapter 5

Iso-Surface

5

An Iso-surface is defined as a region of constant function value. This value can be based on any of the Geometric or Scalar Functions defined for your dataset. You can select the iso-function scalar directly, or you can load the Iso-Surface Function Register on the Function Specification panel. Once the iso-function and the iso-value are specified, **FieldView** will examine the scalar quantities in the field and display the Iso-surface. You may also specify a range of values to sweep the Iso-Surface through. The Iso-Surface itself can be mapped with colors based on the current Scalar Function. For example, the Iso-surface may represent all areas in the solution space with a temperature equal to 500 degrees, while the colors on the Iso-Surface show the various pressures across that surface. For buttons and features not mentioned below, please refer to **Chapter 3**.

This panel can (**Figure 141**) also be used to create Arbitrary Cutting Planes. These cutting planes can be defined by either selecting 3 points in the graphics window, or a single point and a normal direction. The plane can also be swept along its normal vector in order to view a series of cuts. Since an Arbitrary Cutting Plane is a type of Iso-surface, it can be viewed in all the same ways as an Iso-surface.

If the Iso-Surface Function is constant on a grid (minimum = maximum), then no surface will be displayed on that grid.

Exporting: Iso-Surfaces can be exported using the Tools/Export command. For information on the format, please see **Appendix I** of the **Reference Manual**.

Iso-Surface Panel

This button will bring up the Cutting Plane panel, for defining arbitrary cutting planes through the current dataset. Once created, these planes are treated like any other Iso-Surface with regard to sweeping, display types, etc.

This area controls the value of the current Iso-Surface. The minimum, maximum and current values can be controlled by typing in exact values, or moving one of the three sliders. The Min and Max field are used to control the bounds of the animation.

Subset is used to "coarsen up" the data to improve performance, or to skip grids or regions when calculating this surface. The Modify... button will bring up the Iso-Surface Subset Parameters panel. Any increment greater than one will skip grid points when creating the iso-surface. For example, an Inc of 2 means that every other grid point will be used. This is only available for structured grids.
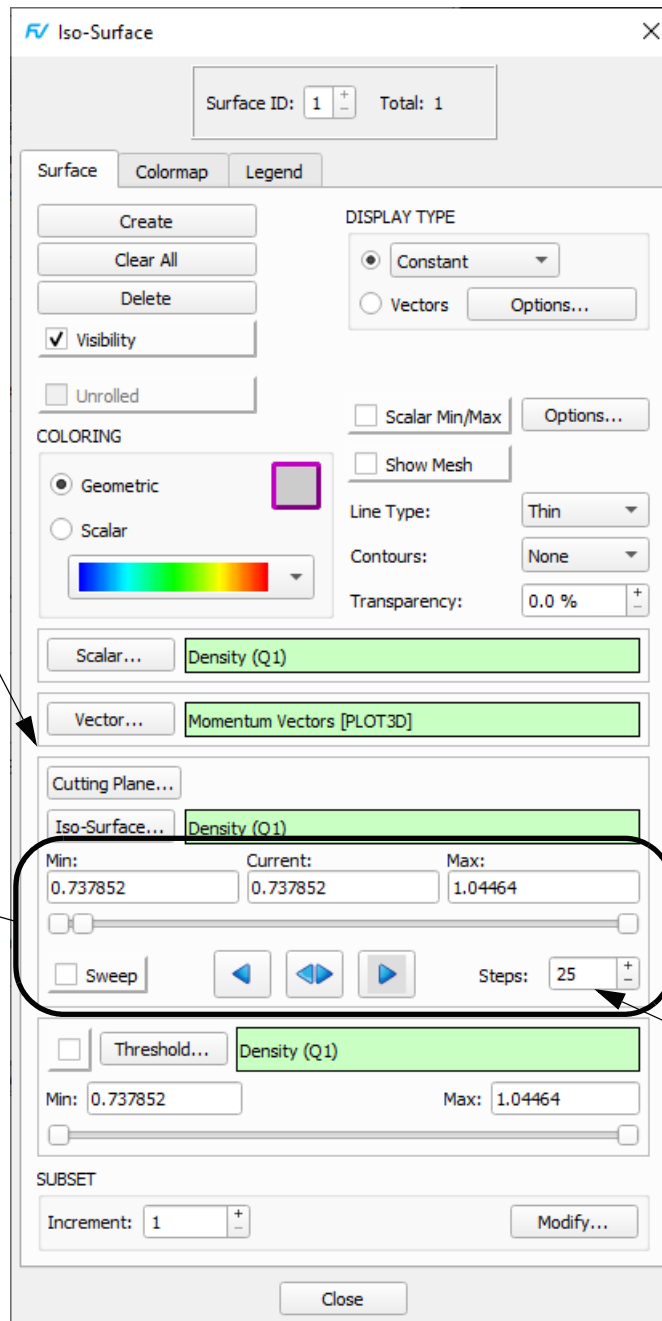
Transient Data note: When an iso-surface is swept in time, if there is a point in time when the iso value for the surface is not within the iso-function range, **FieldView** will turn off visibility for that surface until a time step is reached when the iso value is within range.

The Steps indicate how many different iso-surfaces, between the specified minimum and maximum, should be shown during a sweep operation.

Figure 141   Iso-Surface Panel

Shock Surfaces

Shock Surfaces are one type of Feature Extraction utility in **FieldView**. This surface is a type of Iso-Surface, and is described in **Chapter 11**. The reader should refer to that chapter for full details. The Shock Finding feature has certain functional requirements for use.

You can create a Shock Surface for the current dataset by using the Iso-Surface panel. This is done by loading the Iso Function register with one of the 'special' Shock Finder functions. These functions only exist if certain variables (and variable names) exist for the current dataset. They are indicated in the Function Selection panel with a specific naming convention. All of these functions begin with the words [Shock Finder] followed by a description label for the specific type available. The type available depends on what solver results variables exist and also whether the dataset is steady state or transient. Because of this, not all datasets will have these functions available.

The Shock Finder function is a true Iso-function. However, the only viable value for this function is 1. Therefore, the Iso-Surface panel will show the Iso Function slider as having a Min, Value and Max all equal to 1. None of these three fields will accept any value other than 1.

Cutting Plane Panel

The Plane Definition Method is used be define a plane by selecting 3 points, or a point and a normal. Pressing the Parallel to Screen button will create a cutting plane whose normal vector is pointed directly out of the screen.

Note: The Normal Vector option is disabled if Dataset Scaling is in use.

Depending upon the selection above, either 2 or 3 points must be specified to define the plane. These points may either be selected interactively in the graphics window or typed into these fields.

If the points are typed into the above fields, the Calculate button *must* be pressed to update the cutting plane.

Figure 142 Cutting Plane Panel

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ Cutting Plane Points become RTZ (Radius, Theta, Z) points. See **Chapter 3** for more information.

Note: For cutting planes defined by a Normal Vector, Point 1 is the start of the vector and Point 2 is the end of the vector (Point 3 is not used). For cutting planes defined by 3 Points in a Plane, all three points are used. The points are drawn in the graphics window with markers that are colored as follows:

3 Points in a Plane			**Normal Vector**	
Point #	Color	Meaning	Color	Meaning
1	Green	1st Point	Green	Normal Start, Plane Location
2	Yellow	2nd Point	Red	Normal End
3	Red	3rd Point	Not Used	

Note: The Normal Vector cutting plane option is not available if the user has activated Dataset Scaling.

Displaying Cell Volumes

The cell volumes of an Iso-Surface (often referred to as a Crinkle surface) are easily displayed in **FieldView** (**Figure 143**). Crinkle surfaces allow users to visualize surfaces by showing the complete elements (or cells) that intersect either a coordinate or iso-surface. They permit an examination of mesh quality along with CFD simulation results on an element basis. This view capability is an essential prerequisite for extending **FieldView** functionality to display things like cell metrics, and ultimately, cell-based data. Boundary surfaces and computational surfaces are both defined at element faces, so these surfaces are not considered for this feature.

Crinkle is a DISPLAY TYPE option in addition to Constant, Faceted, Smooth, Mesh, Contours, Vertices and Shaded Vertices. Surface attributes such as Show Mesh and Show Contours are also supported. The illustration below shows the Crinkle display type for two iso surfaces created by the plane tool, and the smooth display type for two additional iso-surfaces.
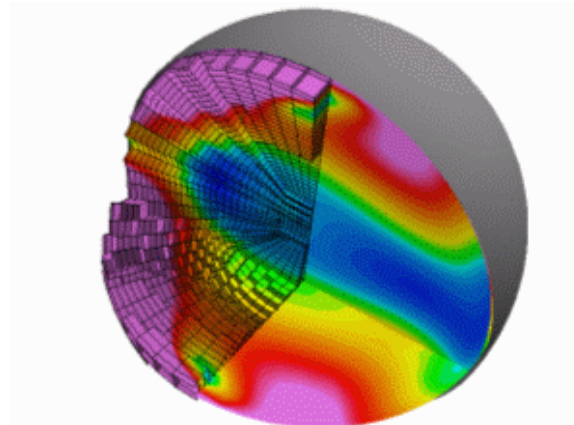
Figure 143 Crinkle Surface

Displaying Crinkle Surfaces

First create a surface inside of the Iso-Surface panel and then choose Crinkle in the Display Type menu.
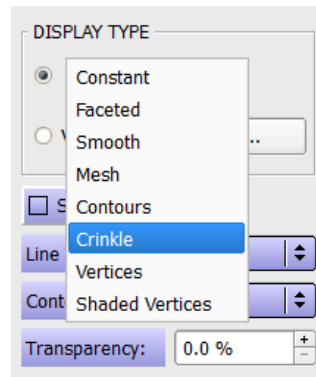
Figure 144   Crinkle Display Type

The Crinkle display type is not available if the surface is Coordinate 'R' or 'Theta' (cylindrical dataset).

Frequently Asked Questions

How do I create an Iso-surface?

To create an Iso-Surface, press the Create button on the Iso-Surface panel. If you have not previously defined an Iso-Surface function, you will be prompted for whether you would like to define a function, create a cutting plane or cancel.
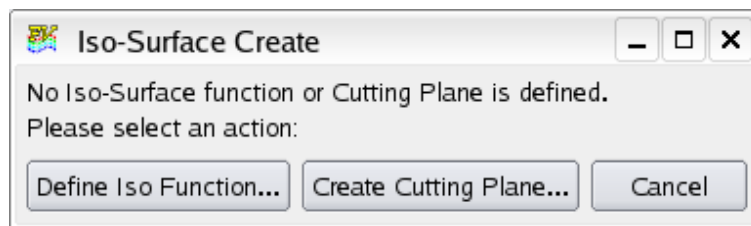
Figure 145 Iso-Surface Create Panel

You should select Define Iso Function. This is accomplished by loading a function into the Iso-surface register on the Function Specification panel (which will be brought up automatically.)

Once you have selected the function, an Iso-surface equal to the minimum value of the Iso-surface function will be created. To change the value of the Iso-surface, you may either move the middle slide bar or type into the field marked value.

How do I create an arbitrary cutting plane?

If you have not previously created an Iso-Surface, press the Create button. You will be prompted for whether you would like to define a function, create a cutting plane or cancel. You should select the cutting plane option, which will bring up the Cutting Plane panel.

There are two ways to define a cutting plane: by selecting 3 points to define the plane, or by selecting 2 points which represent the plane position and the normal direction. Select the Plane Definition Method you wish to use. Next, you must specify the 2 or 3 points. This can either be done interactively in the graphics window (by moving the cursor over the point, holding down the Control key and pressing the left mouse button (CTRL-M1), or by typing the points in on the Cutting Plane panel, and pressing the Calculate button.

> Note: Picking a location with CTRL-M1 will only work within the current dataset. If you try to pick at a location occupied by a dataset other than the current dataset, **FieldView** will "beep".

Once the points have been specified, the plane will appear in the graphics window. You may move one or more of the points interactively with CTRL-M1, or by typing new points in on the panel. Remember to press Calculate to update the plane location.

Once a cutting plane has been created, it is treated like a function in that it has a minimum, maximum and current value. The initial *current* value will be zero and indicates the position of the plane at its creation. The minimum and maximum values give the range of the position of the cutting plane while still remaining within the bounds of the dataset. The positive direction will be in the direction that the normal is pointing. For instance, changing the *current* iso surface slider by `+n` units will relocate the cutting plane surface `n` units in the positive direction along a line normal to the surface. In addition, the cutting plane can be "swept" like any other surface in **FieldView**. This is explained below.

How do I animate my surface?
To animate the current surface, press the button marked Sweep in the Sweep Control area. By default, the animation will move forward and start again at the minimum when it reaches the end. You may also specify the animation to be backward or to bounce between the two by pressing on the arrow keys in the Sweep Control area. The number of steps the animation will take between the specified minimum and maximum is controlled by the Steps field in the Sweep Control area. Pressing on these arrow keys will not only change the direction of the animation, but will also move the plane one step in that direction. The animation can be used for both Iso-surfaces and arbitrary cutting planes. Cutting planes animate along a line perpendicular to the defined cutting plane.

What does subsetting an Iso-surface do (structured grids only)?
Subsetting is used to "coarsen up" the data to improve performance, or to skip grids or regions when calculating this surface. The parameters button will bring up the subset parameters sub-panel. Any increment greater than one will skip grid points when creating the iso-surface. For example, an Inc of 2 means that every other grid point will be used. This is only available for structured grids. For more information, see **Chapter 3**.

Possible Problems

I pressed Create and loaded the Iso-surface register, but no picture appeared in the graphics window.
The initial value for the Iso-surface function is the minimum value for the entire flow field. This is usually a tiny region, which results in no visible iso-surface. Try moving the Iso-surface Value slider or typing in a value which is somewhere in between the minimum and the maximum. Or, you can press Sweep.

Unrolled Cylindrical Iso-surfaces

This feature lets you unroll an isosurface created from one of the three built-in cylindrical iso-functions, namely "Rcyl: (X^2+Y^2)^.5", "(X^2+Z^2)^.5" or "(Y^2+Z^2)^.5". To unroll an existing isosurface (based on one of the three cylindrical functions), simply press the unrolled button on the isosurface panel in **Figure 141**.

Pressing the unrolled button (after an isosurface has been created) will unroll the isosurface around its axis, and transform it flat to the screen. The initial orientation of this surface will have the R*theta axis aligned in the horizontal direction, and the cylinder major axis aligned in the vertical direction. In **Figure 146** below, a cylindrical surface for a single blade passage, which has been rotationally duplicated 6 times, for a compressor is shown at left. In the same figure, the unrolled version of this surface and its initial orientation is shown at right.
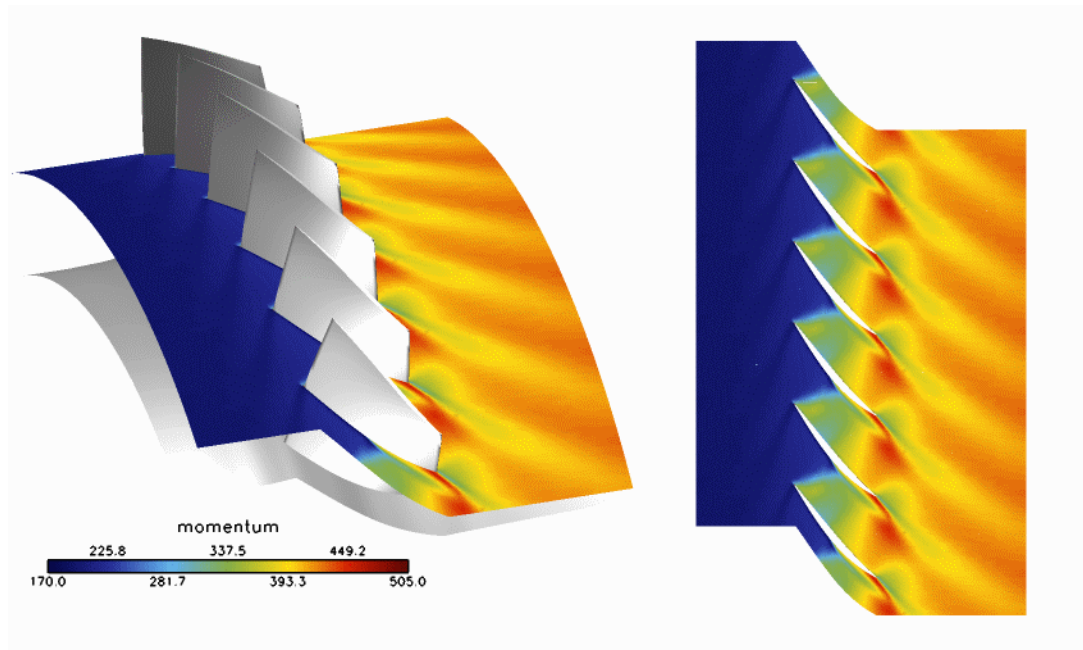
Figure 146 An unrolled Iso-Surface

Note that full support for rotational duplication, at either the dataset or region level, along the machine axis direction is fully supported.

The isosurface, once unrolled, will be permanently detached from all other transform operations. This lets you move the rest of the surfaces in your dataset without moving the unrolled isosurface. Transform controls for unrolled surfaces have been modified at the Surface Transform hierarchical level to permit the following operations:

Figure 147 Surface Transforms for unrolled Iso-Surfaces

Left Mouse	Translates the unrolled surface in the plane of the screen
Middle Mouse	Rotates the unrolled surface in the plane of the screen
Right Mouse	Permits independant horizontal and vertical scaling of the surface

The display properties of the isosurface (such as contours, show mesh, thresholding and transparency) will be carried over from their settings prior to being unrolled. Also, once a surface has been unrolled, its display properties can be modified as desired.

Another illustration of an unrolled surface, created from a cylindrical surface positioned within the scroll section for a compressor is shown in **Figure 148**. Unrolling surfaces which span nearly 360 degrees
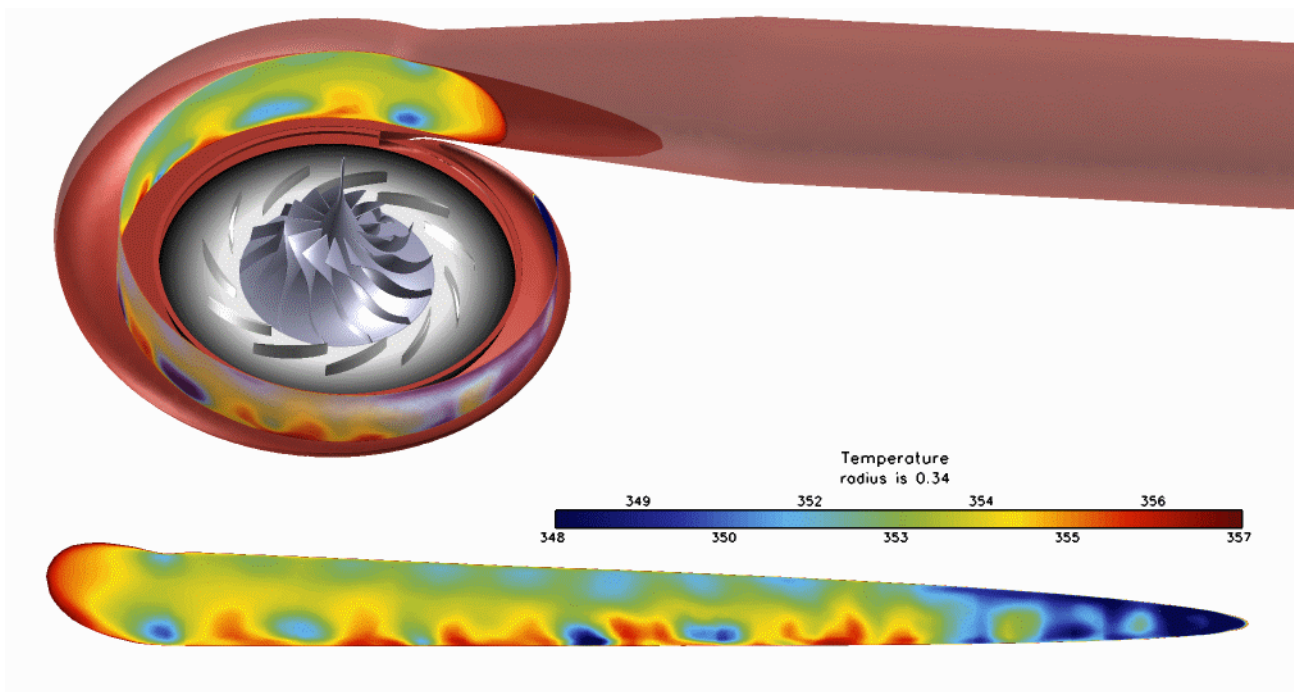
Figure 148 Illustration of an unrolled surface in a compressor

offers the benefit of being able to see changes in scalars over the span of the surface more easily. In the case above, the temperature change on the tip of the cylindrical surface at the start of the volute is much easier to see on the unrolled representation of this surface.

In some cases, there may be a gap in the unrolled surface. This can happen for two reasons. First, there may actually be a gap in the surface prior to it being unrolled. Second, a gap may be created due to the positioning of the seam where the surface is unrolled from. Currently in **FieldView**, this seam will always be located at a theta value of zero. This can be controlled to some degree however by creating (or modifying) a **FieldView** region file; a typical **FieldView** region file may look like:

```
FVREG 2

DATASET_COORD_TYPE        cylindrical
MACHINE_AXIS                     Z
ROTATION_ORIENTATION    CW

! Set origin to center of impeller
ORIGIN    0 0 0

! Set machine axis perpendicular to plane of impeller rotation
MACHINE_AXIS_VECTOR  0 0 1
```

```
! Zero Theta Plane definition (change to move the seam location)
ZERO_THETA_VECTOR 0 1 0
```

By setting the `ZERO_THETA_VECTOR` to be aligned with the desired location of the seam, the location of the gap in the unrolled surface may be controlled. In the example illustrated in **Figure 149** below, we show the overhead view of the compressor and an unrolled surface, matching the cylindrical surface from the overhead view.
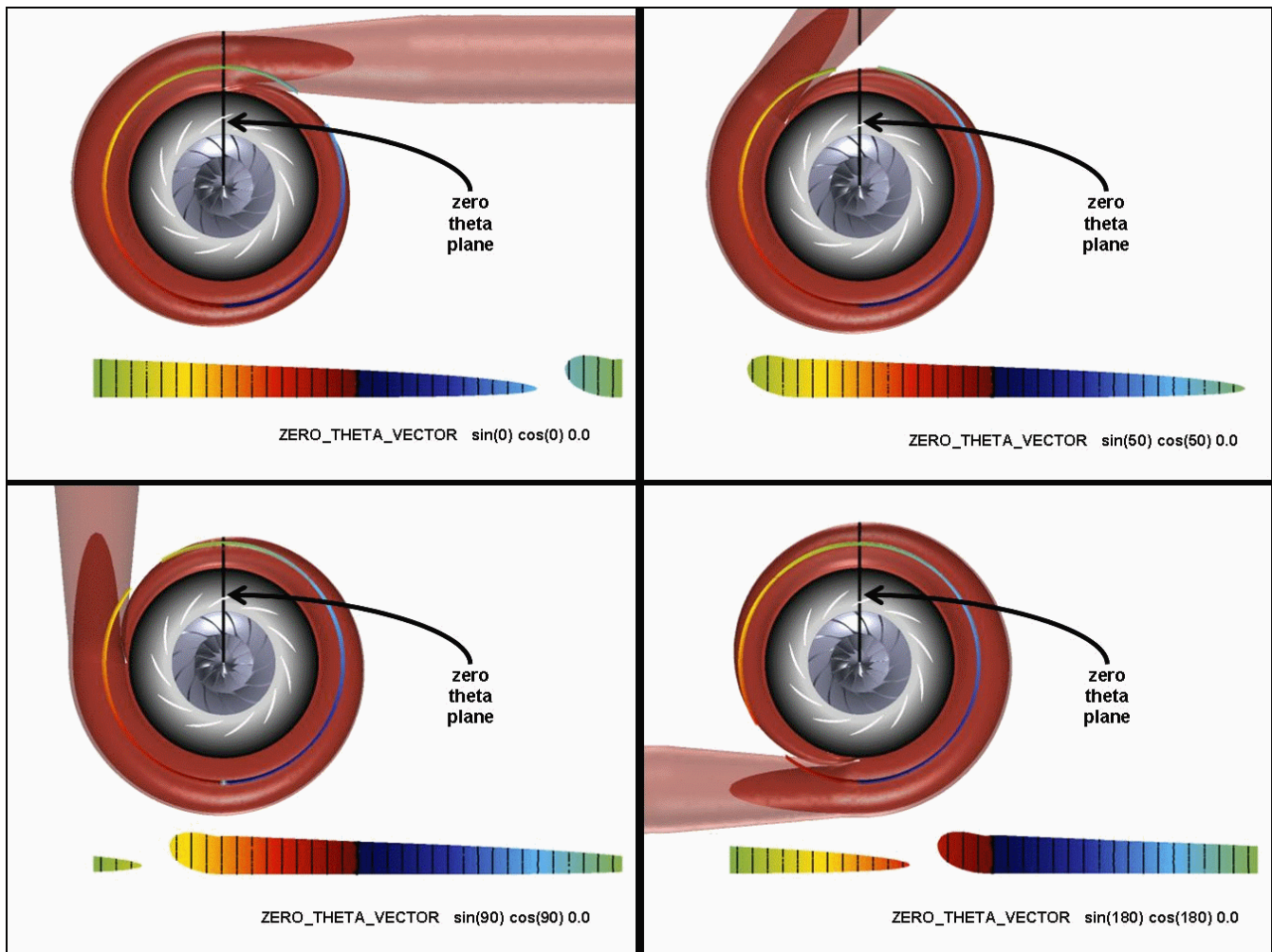
Figure 149 Using a region file to change the seam location for an unrolled surface

Each surface is unrolled in a counterclockwise direction based on the original cylindrical depiction, starting at the zero theta plane. The unrolled surfaces are colored by theta - blue corresponds to -PI and red corresponds to +PI. These cases differ according to the region file that was used; the specification of the `ZERO_THETA_VECTOR` is shown for each case. The difference cases show how the location of the seam and the resulting gap can be manipulated.

Unrolled isosurfaces can be swept simply by pressing the Sweep button on the isosurface panel. By turning Flipbook build mode on, animations of swept, unrolled surfaces can easily be created.

Vectors are displayed by projecting them onto the unrolled surface. A simple case to show how this is done is illustrated in **Figure 150** below. A vector of uniform length is created to point at a 45 degree angle in the XZ plane. For the cylindrical isosurface (which depicts one half of a cylinder), illustrated at left, this vector is seen to be pointing in the same direction for all points on the surface. In the unrolled representation of this same surface, at right, the direction of this vector will change depending on where it is located. Vectors at the axial midpoint of this surface are seen to point at a 45 degree angle. At the left and right edges of this unrolled surface, vectors are aligned with the edges.
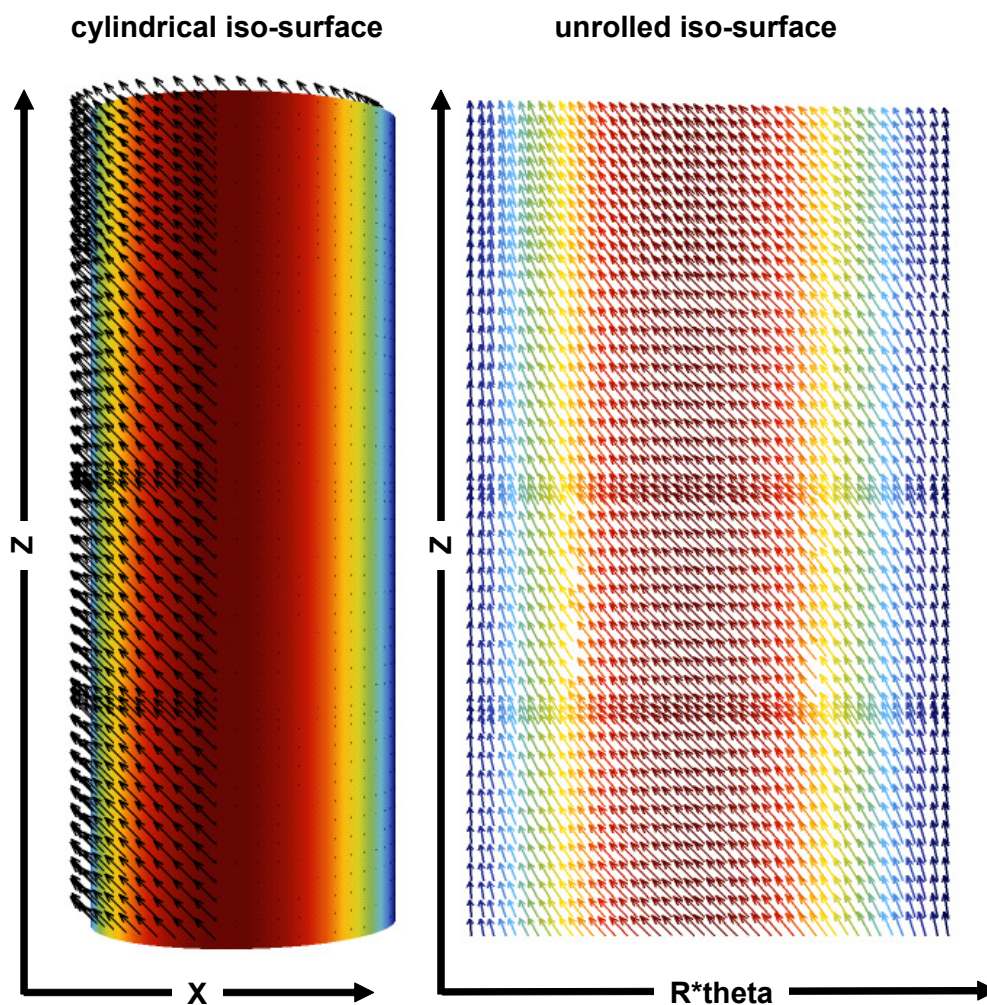
Figure 150 Vectors (45 degrees in XYplane) on an unrolled surface

Point Probe is supported for unrolled surfaces. The actual point probe location marker will be shown within the context of the original dataset and not on the unrolled surface itself. This matches what **FieldView** currently does when you use Point Probe on transformed surfaces.

Limited support for 2D plotting on the unrolled surface is also possible. It is possible to create a line plot, selecting two end points. However, keep in mind that the line plot created by selecting two points on an unrolled surface will actually be constructed through the dataset volume, and will not be constrained to lie on the unrolled surface, as shown in **Figure 151**.

cylindrical iso-surface   **unrolled iso-surface**
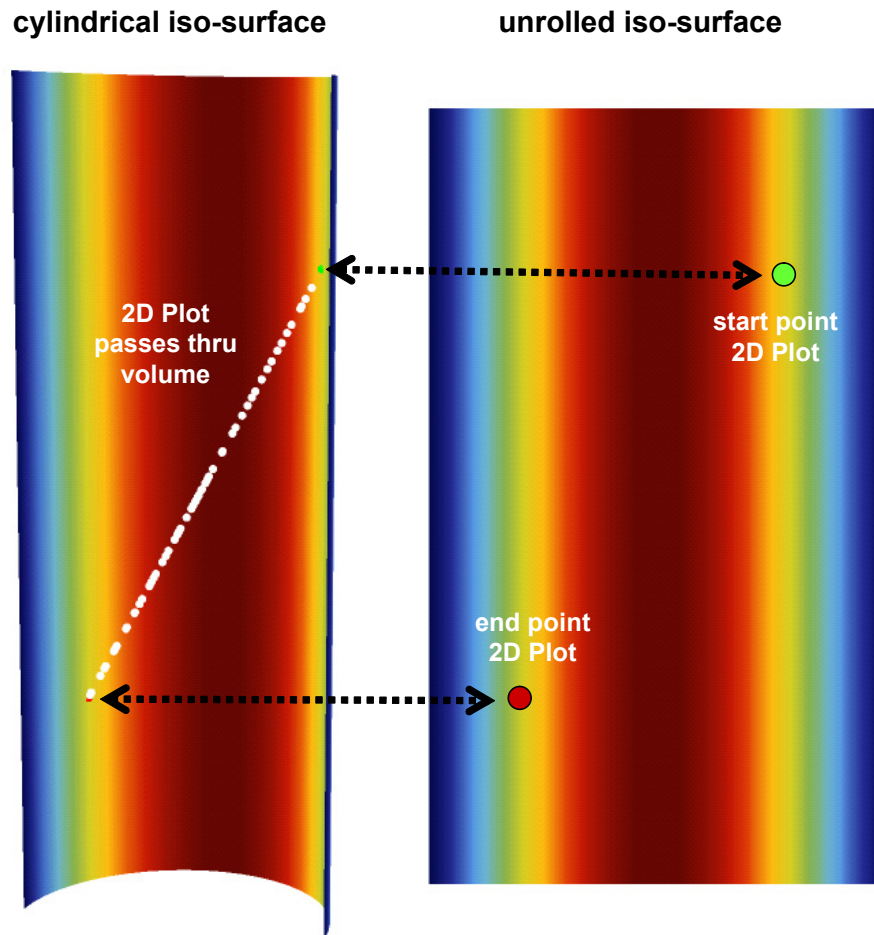
Figure 151 2D line plot creation by selecting on an unrolled surface

Unrolled isosurfaces are fully supported by **FieldView** RESTARTS. In addition, **FVX** can be used to create an unrolled surface, by explicitly setting the unrolled state of a surface. An example using **FVX** to create an unrolled surface is:

```
unrolled_surface = create_iso( {dataset = 1,
    iso_func = "Rcyl: (X^2+Y^2)^.5",
    scalar_func = "density",
    iso_value = {current = 1},
    show_mesh = 'off',
    unrolled = 'on',
```

```
    }   )
```

In the case that you attempt to unroll an isosurface using **FVX** whose isofunction is not one of the three cylindrical built-in functions, you will receive an error message, and the surface will not be created.

Additional Comments and Limitations

Unrolled surfaces are reconstructed from their original representations by applying a numerical transform, mapping XYZ coordinates to an R-theta representation. Consequently, integrations calculated on unrolled surfaces may differ slightly from those obtained from their parent representations. For datasets with coarse meshes, these differences may be significant.

Partial surface integration is currently not supported for unrolled isosurfaces.

Unrolled surfaces cannot be exported, to either the standard ASCII text export formats, or to the XDB file formats (compatible with the **FieldView** Advanced Technology Viewer).

The crinkle surface display type is not enabled for unrolled surfaces.

Display properties such as scalar or vector display, or transparency for instance, can be unified across unrolled and normal isosurfaces. However, it is not possible to unify the unrolled surface attribute. This is consistent with not being able to unify the isofunction across a series of isosurfaces.

Chapter 6

Streamlines

6

The Streamlines panel (**Figure 154**) is used to calculate and display streamlines, vortex lines, and other paths that can be computed by integrating through a vector field. A set of seeds must be placed in the flow field. Once the seeds are created, the Streamlines can be calculated. The integration will be performed in the vector field defined by the current Vector Function Register, as shown on the Function Specification panel. To calculate true streamlines, the Vector Function Register should be loaded with Velocity.

In transient data, Streaklines (which go across time steps) are usually more useful than streamlines (which only use the current time step). If transient data with Streamlines is present in **FieldView** and you press the Sweep button in the Transient Data Controls panel, **FieldView** will calculate Streaklines. See **Streaklines** later in this chapter for full information.

Exporting: Streamlines can be exported using the Tools/Export command. For information on the format, please see **Appendix I** of the **Reference Manual**.

Note: After a PLOT3D Q file is read, the default Vector Function register is Momentum Vectors [PLOT3D]. This is incorrect for streamlines unless the flow in incompressible. Otherwise, you should change the Vector function to Velocity Vectors [PLOT3D].

Rake Definition

A Rake is defined as a set of seed points and streamlines. Any number of rakes, each with up to 2000 seed points, can be created in a visualization.

A rake acts as a grouping mechanism, so that it is easy to change the color or other visual attribute for an entire set of streamlines, in a single operation.

Streamlines can be drawn using one of several Display Types.

Complete	Default type, shows entire path	Not an animated type.
Filament	Streamline divided into segments, with segment length determined using equal time intervals based on longest streamline or length fraction. The number of filaments is controlled by divisions (divs).	Looped animation, length is 10 frames.
Filament & Arrows	Streamline divided into segments, same as Filament display type. Arrows are placed at the end of filament segments to show flow direction. The number of filaments is controlled by divisions (divs). Arrow size can be independently controlled.	Looped animation, length is 10 frames.
Filament & Spheres	Streamline divided into segments, same as Filament display type. Spheres are placed at the end of filament segments to show flow direction. The number of filaments is controlled by divisions (divs). Sphere size can be independently controlled.	Looped animation, length is 10 frames.
Growing	Animates streamline from seed point with length successively increasing. Step interval is determined using equal time segments based on longest streamline or length fraction.	Animation length is equal to number of divisions (divs).
Spheres & Lines	Spheres are added to the Complete display type. Sphere positions are calculated similar to the growing display type. Spheres animate by moving along the complete streamline.	Animation length is equal to number of divisions (divs).
Spheres	A single sphere is displayed for each streamline. The position of the sphere along the streamline trajectory is controlled using the Animation settings.	Animation length is equal to number of divisions (divs).
Dots	A single dot is displayed for each streamline. The position of the dot along the streamline trajectory is controlled using the Animation settings.	Animation length is equal to number of divisions (divs).

Lines of Spheres	Many spheres are used to display the streamline. The spacing between the spheres is determined using equal time intervals based on longest streamline or length fraction. The number of spheres is controlled by divisions (divs). Sphere size can be independently controlled.	Looped animation, length is 10 frames
Lines of Dots	Many small dots are used to display the streamline. The spacing between the dots is determined using equal time intervals based on longest streamline or length fraction. The number of dots is controlled by divisions (divs). Dot size cannot be adjusted.	Looped animation, length is 10 frames
Ribbons	A flat continuous ribbon is used to display the streamline. The ribbon twists to show the local rotation of the flow. Ribbon width can be independently adjusted.	Not an animated type.

Presentation Rendering

This feature will change the way that rakes will look in the graphics window. The following characteristics of streamlines will be affected:

- Shaded cylinder streamlines will be shown for streamline types: Complete, Filament, Growing, and Spheres & Lines. The size of the cylinders can be set by the Thin/Medium/Thick options.
- Spheres (for Spheres & Lines, Spheres, and Lines of Spheres) will be shown as shaded spheres. The size of the spheres will be determined by their existing display options (e.g. Scalar Sphere Sizing, etc.).
- The sphere in streamline display type Spheres & Lines will be the same color as the line
- The dots in the streamline type called Lines of Dots will not be shaded.
- Ribbons will be light source shaded.
- Streamline seeds will also be light source shaded.
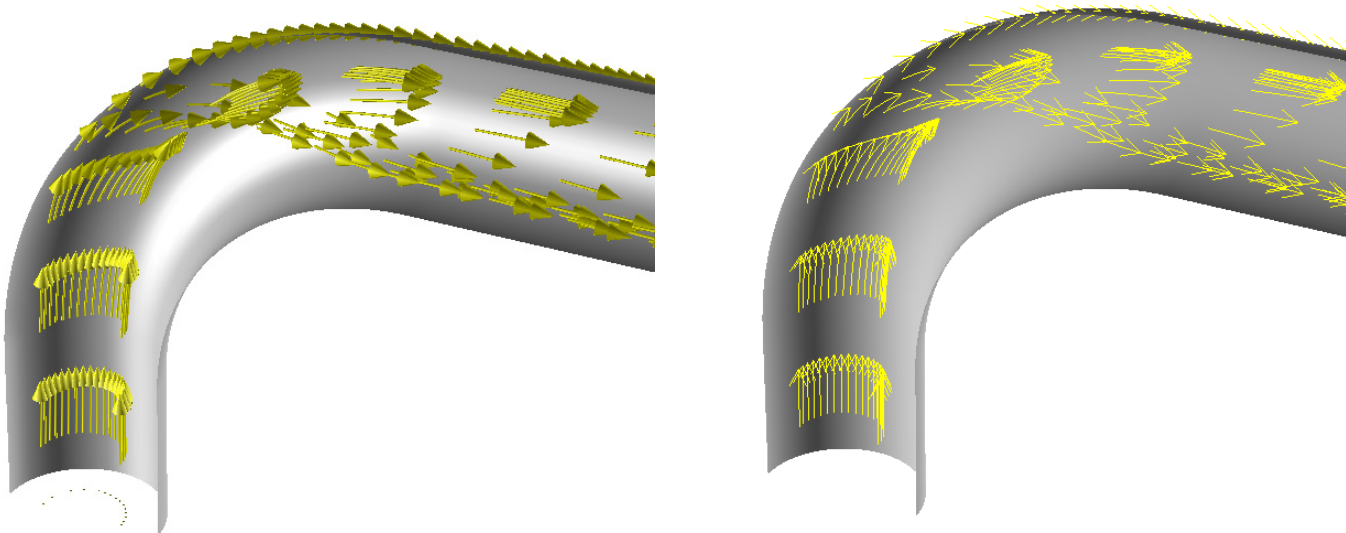
Figure 152 Filament & Arrows with (L) & without (R) Presentation Rendering
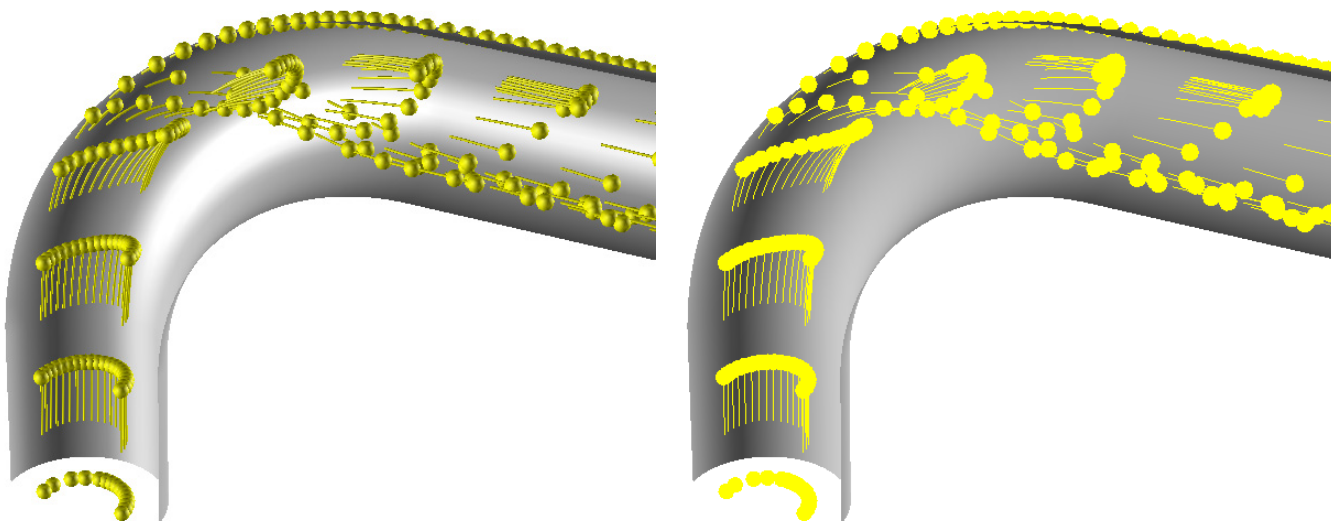
Figure 153 Filament & Spheres with (L) & without (R) Presentation Rendering

Streamlines Panel

Subset allows skipping of grids or regions when calculating streamlines. For structured grids only, each streamline rake can be subsetted in the I, J, K directions. This can be used to eliminate the streamlines from certain areas of the model. Note that IJK Clipping and Grid INCREMENT settings (but not grid/region visibility) will affect display of the Auxiliary Seed Plane.

The possible streamlines Display Types are: Complete, Filament, Filament & Arrows, Filament & Spheres, Growing, Spheres, Spheres & Lines, Spheres, Dots, Lines of Spheres, Lines of Dots and Ribbons. Complete and Ribbons do not animate. If the display type is set to Ribbons, an additional control will be displayed to control the width of the ribbon. For any display type with Spheres, two additional controls are available to scale the spheres, or to set their size based upon the current scalar variable.

Sets the Transparency of the streamlines. A value of 0.0 is fully opaque, while a value of 1.0 is fully transparent. Valid values are 0.0, 0.125, 0.25, 0.375,…,1.0.

Streamlines may be animated by turning on the Animate button. The number of steps the streamlines will be cycled through, as well as the direction, can also be controlled.

With Seed Coordinates the user can seed at I, J, K Integer, Real locations, or at X, Y, Z coordinates. Warning: Changing this selection will delete any seeds on the current rake. After pressing Create, this selection should be made.

Once the seeds have been created, pressing the Calculate button will compute and draw the streamlines. Using the Streamline Parameters sub-panel, the streamlines may be restricted to a surface, the step size and direction of the calculation can be controlled, and the streamline limited on duration.

An Auxiliary Seed Plane may be positioned anywhere in I, J, K or X, Y, Z space as an extra area to seed particles. The plane may be shown as a mesh, a scalar colored mesh, a surface or a scalar colored surface.
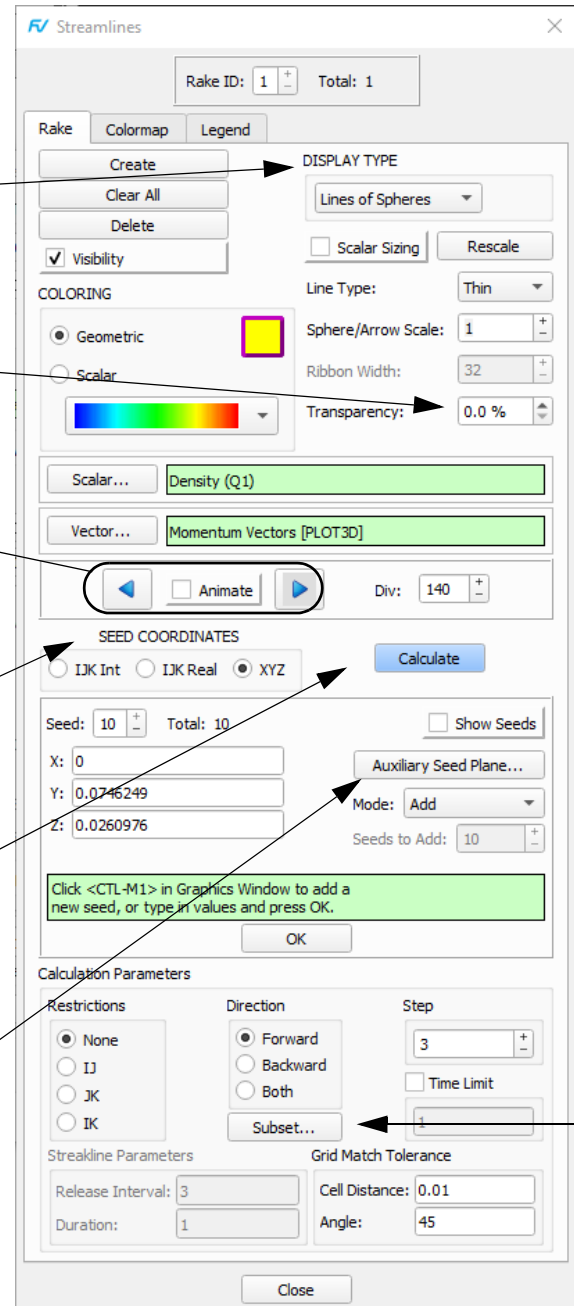
Figure 154 Streamlines Panel
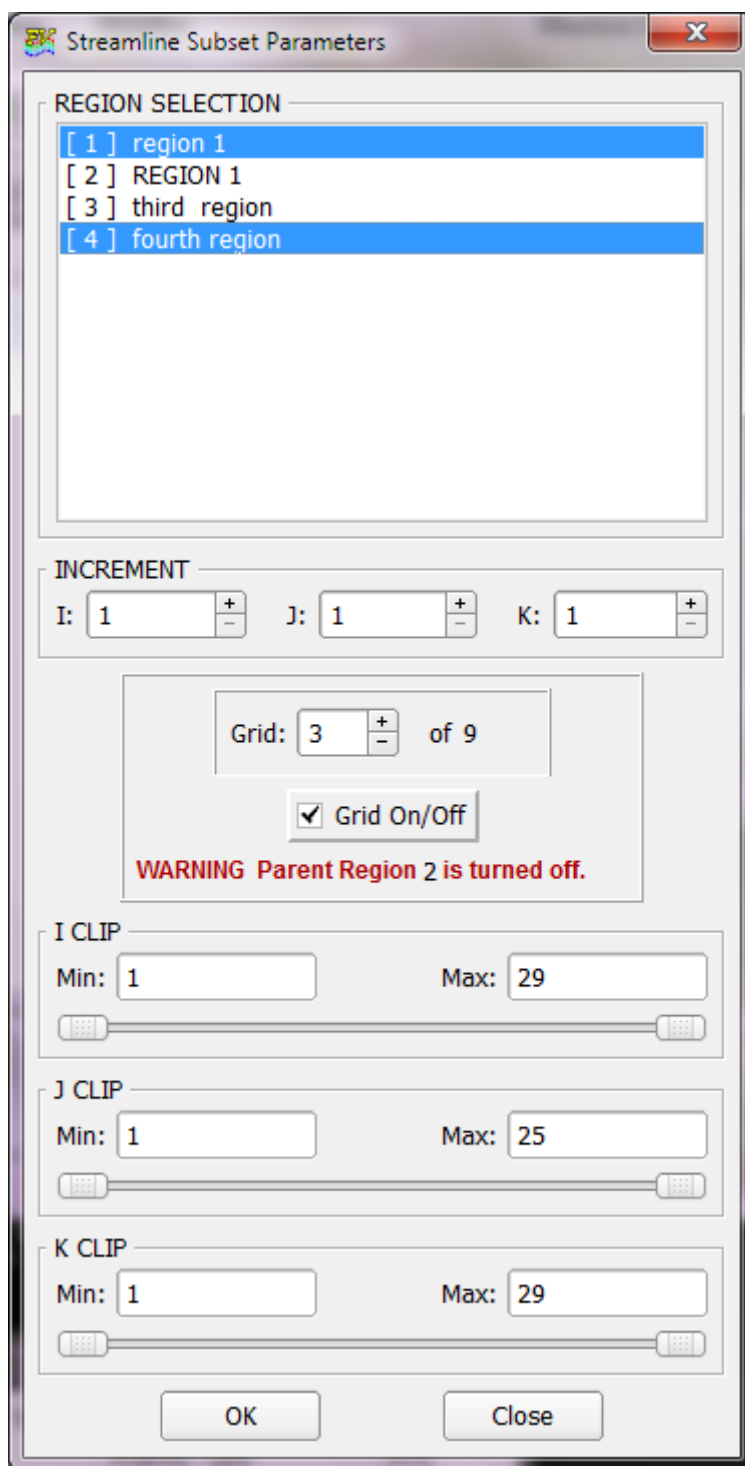
Figure 155 Streamline Subset Parameters Panel

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ SEED COORDINATES become RTZ (Radius, Theta, Z) coordinates. Furthermore, when RTZ seed coordinates are used, the AUXILIARY SEED PLANE directions become RTZ as well. See **Chapter 3** for more information. However, only Cartesian (absolute or relative) velocities should be used as the Vector Function for streamline and streakline calculations. Cylindrical velocities ([cyl] and [cyl][rel]) should *not* be used. For buttons and features not mentioned above, please refer to **Chapter 3**.

The Mode menu controls the behavior of the Seeding Area. The possible modes are:
Add: for adding particles
Delete: for deleting particles
Move/Edit: for changing the position of particles already created
Add Between: for creating a line of particles
Seed a Surface: for creating a group of particles on a particular surface.

This area is used to select the current seed. You may type or use the arrow buttons to scroll through the seeds.

When the mode is Add, this area is used for typing in the coordinates of a particle to seed. When the mode is Move/Edit, the coordinates of the current particle may be changed by editing these fields. Otherwise, these fields will show the coordinates of the current seed.

When Show Seeds is on, the seeds will be shown as spheres in the graphics window, with the current seed colored red and all others colored green.

When Select Point is on, pressing M1 while the cursor is in the graphics window will add a seed, delete a seed or select the seed at the current location, depending on the Mode. When Select Point is off (the default) holding down the Control Key while pressing M1 will perform the same operation.

This display field will prompt the user for actions required depending on the Mode.

The Apply/OK button must be pressed for any changes made in the type-in fields to take effect. If the mode is Delete, this button will delete the current seed.

Figure 156   Seeding Controls

Calculation Parameters Panel

Streamlines may be integrated either downstream (forward), upstream (backward) or in both directions. The default is forward.

Integration Restrictions are useful in calculating stream-lines that lie on the surface of flow boundaries. This is analogous to the path taken by a viscous substance (such as a drop of oil) along a surface of an object in the flow field. For IJK-Real seeds, surface restriction is to the nearest integer value. The streamlines may be restricted to either an I, J or K surface for structured grids only.

Integration Step is used to specify the precision of integration within a cell. For example, if this value is set to ten, then each integration step covers a distance of at most one-tenth of a cell's largest dimension. This causes the streamline calculation time to increase, but gives more accurate results for rapidly curving streamlines.

Time Limit, if turned on, will limit the length of streamlines. If the Direction is Both, 1/2 (half) the specified Time Limit will be Forward and 1/2 will be Backward (for example, a Time Limit of 10 will be 5 Forward and 5 Backward). The units are the same as for the current Vector Register. Minimum value must be greater than 0.0. There is no maximum. Complete streamlines will be produced when Time Limit is equal to the Duration of longest streamline value.

When sweeping transient data, these options will controll how often particles are released (Release Interval) and for how many consecutive time steps they will be released (Duration.)

This section's controls support grid jumping under cases of Surface Restricted streamlines. The defaults and ranges of these parameters are:
- Cell Distance default: 0.01 min: 0.0 max: 0.5 (fraction of a cell)
- Angle default: 45 min: 0.0 max: 90

Cell distance should be strictly less than 0.5. These settings apply to the current rake, not per streamline or across all rakes.

Figure 157 Calculation Parameters Panel

Using the Streamlines Panel

What are Streamlines?

Streamlines are the paths of massless particles released at specific points in the dataset. The locations of the particles are determined by integrating through the current vector function loaded in the Function Registers using a second order Runge-Kutta method, with an adaptive step size (based on

several criteria). This allows for a much more accurate solution than a standard second order Runge-Kutta. The seed particles and their associated streamlines are grouped into Rakes by **FieldView**.

How do I create streamlines?

In order to create streamlines, you must first create a rake by pressing the Create Rake button on the Streamline Panel. Next, the seed locations must be specified either interactively or by typing in specific values. Once the particles have been seeded, pressing Calculate will create the streamlines. For structured grids, particles will be seeded on I, J, and K integer locations, by default. You may choose to seed on I, J, or K real values or in X, Y, or Z coordinates by choosing a different option in the Seed Coordinates area. The Seed Coordinates selection is for an individual rake. If the option is changed after seeds have been placed, all seeds in the current rake will be lost, although you will first receive a warning asking you to confirm your action. For unstructured grids, particles may only be seeded in X, Y, Z coordinates.

How do I seed particles in the graphics window?

To seed particles in the graphics window, simply move the mouse to position the cursor in the graphics window over the location where you wish to seed the particles, hold down the Control key and press M1 (CTRL-M1). If a point outside of the current dataset is picked, **FieldView** will "beep". The I, J, K or X, Y, Z coordinates (depending on the selected Seed Coordinates) of the seed will be displayed in the Seeding area of the Streamlines panel. If you turn the Select Points button on (default is off) you will not need to hold down the Control key to seed the particles; however, the Transform Controls will be disabled until Select Point is turned off.

How do I seed particles at specific locations?

In order to seed particles at a specific location, the Seeding Mode must be Add. Next, make sure that the Seed Coordinates area is set to the desired selection. Then, type in the desired I, J, K or X, Y, Z coordinate in the Seeding area and press the Apply/OK button. The Total Seeds number will be incremented by one when the seed is added, as the seeds are numbered consecutively within each rake.

How do I move the location of a seed ?

To move a seed location, change the Seeding Mode to Move/Edit. At this point, the location of the current seed may be changed. The coordinates of the current seed will be displayed in the Seeding area, and, if Show Seeds is on, the current seed will be colored red. The current seed can be moved interactively just as if you were creating a new seed except the current seed will be moved to the new location. Alternatively, you may edit the coordinates of the current seed in the Seeding area and press the Apply/OK button to move the seed. To make a different seed the current seed, press the Seed increment or decrement arrows in the Seeding area. As the arrows are pressed, the Seed number will be changed, and the coordinates of the Seed will be displayed in the Seeding area. If the Show Seeds button is on, the current seed will be colored red, while all other seeds will be green.

How can I delete an unwanted seed?

To delete an unwanted seed, begin by changing the Seeding Mode to Delete. The current seed can then be deleted by pressing the Apply/OK button. To make a different seed the current seed, press the Seed increment or decrement arrows in the Seeding area. To delete seeds interactively, move the mouse to position the cursor in the graphics window over the seed you wish to delete and press CTRL-M1.

How can I create a line of streamline seed points?

To create a line of particles, the two end points of the line must first be defined using Add mode. Next, the seeding mode should be changed to Add Between. Then you must select the two end points and define the number of seeds wanted on the line. You will be prompted for these actions in the Seeding Area. The end points are chosen interactively by moving the cursor into the graphics window over each of the desired points and pressing CTRL-M1. Once the end points of the line are selected, set the number of streamline seeds desired by setting the Btw number in the Setup Area and pressing the Apply/OK button.

How can I create a group of streamline seeds on a particular surface?

To create a series of particles on a surface, first set the seeding mode to Seed a Surface. Then you must select which surface you wish to seed upon. This is accomplished by moving the cursor into the graphics window over the correct surface. If the Select Point button is off, you will need to press CTRL-M1. If it is on, you only need to press M1. Next, you will be prompted for the number of particles that you wish to seed on that surface. This is set by adjusting the Btw number in the Setup Area, and pressing OK. At that point the particles will be placed on the surface. The number of particles that will be created is equal to the lesser of the Btw number, and the number of polygons that make up the surface. Press Calculate to create the streamline paths.

Note that the particles will be created across the entire surface that was selected in the graphics window. Thus if you create a surface that includes an inlet as well as part of a wall, but you wish to seed only on the inlet, you should trim the surface (possibly using subsetting or thresholding) such that only the inlet portion is visible, before using Seed a Surface mode.

> Note: Seed a Surface mode is now available when seeding in IJK Int mode. When Add Surface mode is selected for IJK Int mode, the Btw: prompt is replaced by an Inc: prompt, with the default changing from 10 to 3. This Inc field allows the user to select whether they will seed on every grid point (Inc=1) every other grid point (Inc=2) etc.

If a Computational surface is picked, the surface has its own Inc: values for IJ or IK or JK. These will be internally multiplied by the seeding Inc: to determine that actual IJK grid point skipping. In other words, if there is an I=5 surface with Jinc=3 and Kinc=5, and this surface is seeded with a seed Inc of 2, then the surface is drawn as if it had Jinc=6 and Kinc=10, applying the current thresholding. The resulting grid points are then made into IJK Int seed positions.

For structured boundary surfaces, the Inc will operate as if an Seed a Surface (IJK Int) were done on each computational surface individually.

> Note: If the number of seeds to be created exceeds the maximum number of seeds remaining on the current rake, then no seeds will be created on the surface, and the following error message will be printed:
> ```
> Cannot add that many seeds - would exceed
> maximum number of streamlines on a rake.
> ```

What is the Auxiliary Seed Plane?

The Auxiliary Seed Plane can be used to seed particles on a surface other than the surfaces currently displayed. In this way, you can seed particles on a particular I, J, K plane (for structured grids) or X, Y, Z plane (for unstructured grids), without creating the plane itself. The Auxiliary Seed Plane can be positioned anywhere in space using the slider in the Auxiliary Seed plane area, and the buttons to select the coordinate desired. The exact value of the plane location may also be typed into the Value type-in field above the main slider. The plane may be displayed as a Mesh, Scalar Colored Mesh, Surface or Scalar Colored Surface by changing the Display Type (Mesh is the default). To turn on the Auxiliary Seed Plane, press the button on the upper left hand side of the Auxiliary Seed Plane area. By default, the Auxiliary Seed Plane will be turned off, and will be an I plane (for structured) or an X plane (for unstructured) when turned on.

How do I see the locations of the seeds in the graphics window?
If a permanent marker is desired to show the location of the seeds in the graphics window, turn on the Show Seeds button in the Seeding Area. All of the seeds will be shown as green spheres, except for the current seed, which will be red. These markers will remain in the graphics window until Show Seeds is turned off or the rake is deleted.

How do I Move/Rotate the model once the Streamline panel is up?
When Select Point is on (default is off), you will be able to seed, move, or delete particles interactively in the graphics window without holding down the Control Key on the keyboard.   However, the transform controls will be disabled. In order to use the transform controls to move or rotate the model, turn Select Point off.

Why do my streamlines only go downstream?
By default, streamline integration will only go downstream. Using the integration direction option, on the Calculation Parameters sub-panel, you may select whether the streamlines should be calculated upstream, downstream or in both directions (showing the full streamline).

Can I control the accuracy of the Streamline?
Adjusting the Integration Step size on the Calculation Parameters sub-panel will change the accuracy of the streamline. This value is the actual precision of the integration within each cell. For example, a value of 3 (the default) will cause each integration to cover a distance of at most 1/3 of a cell's maximum dimension. When this value is larger, the integration time is longer, but the streamlines may get more accurate.

What is the duration message printed in the background window mean?
When the streamlines are calculated, a message will be printed in the window where you started **FieldView**. This message will display the duration of the streamline animation cycle. In other words, this message will show the amount of time (in whatever units the current vector is in) that it will take the longest streamline to animate from start to finish. This is also the time it takes a massless particle to travel from beginning to end of the longest streamline. This message is printed for the current rake and will give the rake number as well as the duration of the longest streamline. The units are those of the currently loaded vector register.

What controls how my streamlines go across multiple grids?

According to the PLOT3D standard, in order to allow streamlines to go across multiple grids, the IBlank values must be set appropriately on each grid (see the **Appendix B** of the **Reference Manual**). If, however, you do not have the IBlank information in your data (or it is not set correctly), **FieldView** will still make your streamlines go across grids. Whenever a streamline reaches the end of a grid, it will jump to the next grid automatically if they are adjoining. For chimera or overset grids, where the grid to jump into may be ambiguous, you will have to set the IBlanks correctly to ensure that the streamlines will jump into the correct grid. Otherwise, if there is more than one possible grid to jump into, **Field-View** will pick the grid with the lowest number that does not have IBLANK equal to zero at the jump location.

If you do not want the streamlines to go across grids automatically, you will need to set the environment variable:

```
FV_NO_GRID_JUMP=1
```

This can either be done by hand before **FieldView** is started, or it may be added to the **FieldView** startup script `fv`. If this flag is set, the streamlines will only go across multiple grids if the IBlanks are set correctly.

Why are my seeds not being placed on the grid number that I want when I seed on overlapping grids?
FieldView gets the (X,Y,Z) position of the picked location in the graphics window. It then determines which grid this XYZ point lies in. For the case of overlapping grids, **FieldView** traverses the grids from lowest number to highest number. The first match is used as the grid to seed on. Therefore, the lowest number grid is always the one "found" by **FieldView**, even if seeding on a visible Computational surface. In order to seed on the higher numbered grid, either the seed location can be edited and the desired grid number entered manually (change SEEDING Mode: to Move/Edit, change the Grid field and press the Apply/OK button), or the grids need to be renumbered (outside of **FieldView**) so that the desired grid has a lower grid number.

How can I control the size of my streamline spheres?
For any streamline type that contains spheres, you may control the size of the sphere by adjusting the Scale field directly beneath the Display Type. In addition, by turning on the Scalar Sphere Sizing button, the size of the spheres will be based upon the current scalar variable.

Length Fraction Controls

Filament segment lengths are based upon the residence time of the longest streamline in a given rake, or set of streamlines. However, it is often the case that a small number of streamlines may be caught in a recirculation zone, and as a result, their residence times may be much longer than the other streamlines in the rake. Since the filament length uses this longest time, the filament display applied to the much shorter streamlines may produce a picture or animation in which streamlines appear to be missing.
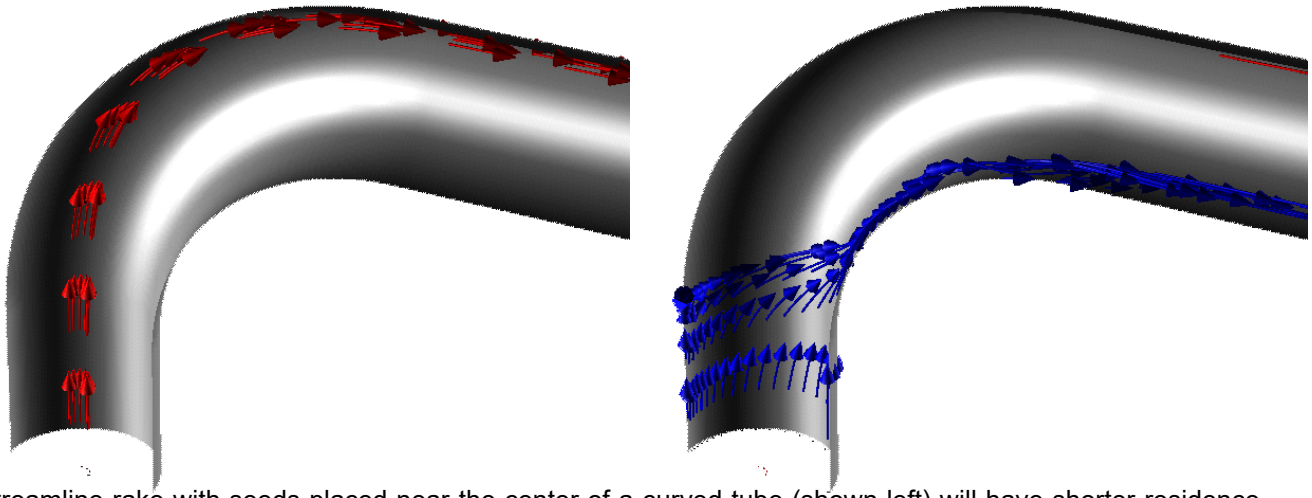
A streamline rake with seeds placed near the center of a curved tube (shown left) will have shorter residence times than a streamline rake with seeds placed near the wall of the curved tube (shown right).

Figure 158   Streamlines with differing residence times

As an example, consider the two streamline rakes previously illustrated for flow in a curved tube. On the left, streamlines are introduced at the center of the tube - these streamlines will have the shortest residence time. On the right, streamlines are introduced very near the wall of the tube - these stream-lines will have the longest residence time. When the filament length is calculated for the longer streamlines, the faster moving streamlines in the center appear to be gone. This is because the fila-ment segment length is too short to properly display these streamlines.

To resolve this problem, an additional control parameter, length fraction, has been introduced. This parameter ranges from 0 to 1, and it corresponds to the cumulative fraction of all streamline rakes, spanning all datasets, for the distribution of residence times.   A default value of 0.9 has been chosen and should work well to eliminate streamlines with uncharacteristically long times. A histogram for a typical streamline rake is illustrated in the figure below. At the default length fraction, the residence time used to calculate the filament length is based on the residence time for 90% of all of the stream-lines calculated. In the example illustrated, the residence time drops from 1.0 to a value of approxi-mately 0.37.

Important Note: This change affects all streamline and particle path display types, except for Complete and Ribbon. Therefore, streamline and particle path RESTARTS are likely to produce images which will be different from those calculated with earlier versions of **Field-View**. If the illustration for streamlines or particle paths generated with earlier **FieldView** versions is desired, an environment variable can be used to override the length fraction default setting:

```
FV_USE_LONGEST_PATH = 1
```

It is possible to refine the appearance of streamline and particle paths further by directly controlling the value of length fraction using the **FVX** utility called `length_fraction.fvx`. In the following illustration, changing the length-fraction to a value of 0.5 would reduce the residence time down to a value of 0.22.

Figure 159 Streamline Time Distribution Histogram
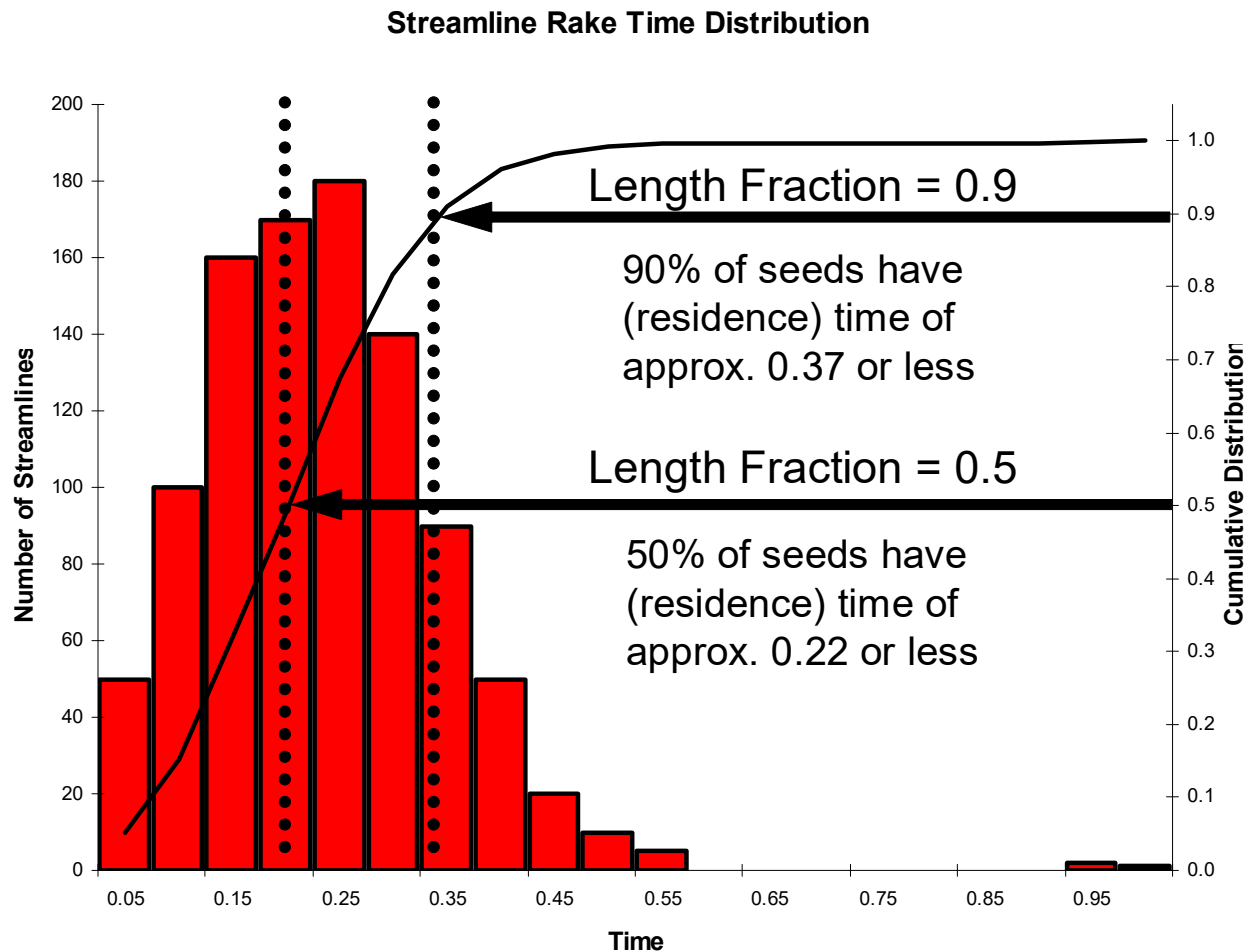
As a practical illustration, the curved tube flow example is shown again in the following figure. At left, a length fraction of 1.0 set using the environment variable, `FV_USE_LONGEST_PATH`, is applied. At right, the same streamline rakes are shown using a length fraction equal to 0.6, which is set using the **FVX** utility, `length_fraction.fvx`. The filament segment length for the Filament & Arrows display type is well balanced for the difference in residence times between the two rakes.
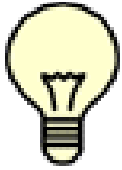
Important Note: The **FVX** utility, `length_fraction.fvx` is located on the **FieldView** DVD in the `/fvx_and_restarts` subdirectory. To use this utility, first copy it to common or shared directory. Next, call the **FVX** utility from the tools menu, or the command line using the `-fvx` switch before you generate any streamline rakes. The slider on this panel will let you actively control the length fraction.
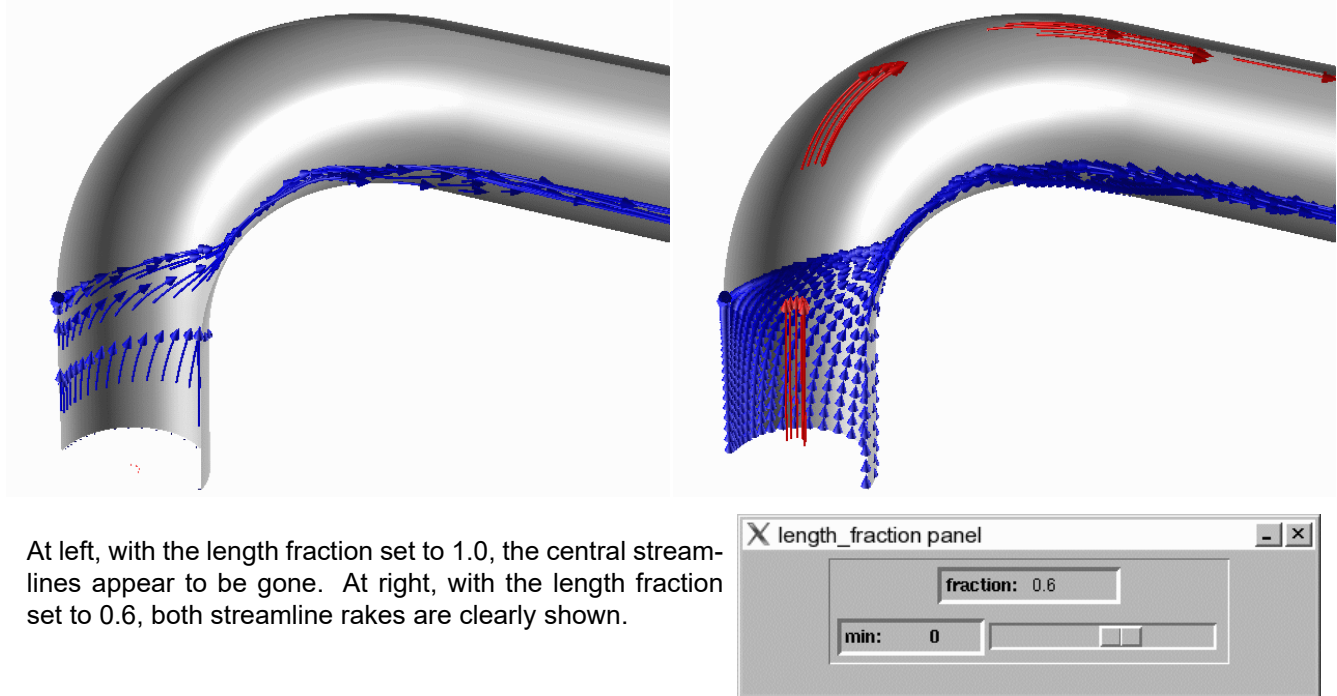
At left, with the length fraction set to 1.0, the central stream-lines appear to be gone. At right, with the length fraction set to 0.6, both streamline rakes are clearly shown.

Figure 160   Example showing the FVX length fraction feature

The divs setting will also have an impact on the filament segment length. In the following figure the streamlines produced for a length factor of 0.6 and divs equal to 25 are shown at left. By increasing the divs value from 25 to 50, we can see shorter segments in the illustration at right. Depending upon the problem of interest, the length fraction and divs setting can be tuned to optimize the display of either the streamlines or particle paths for any problem.
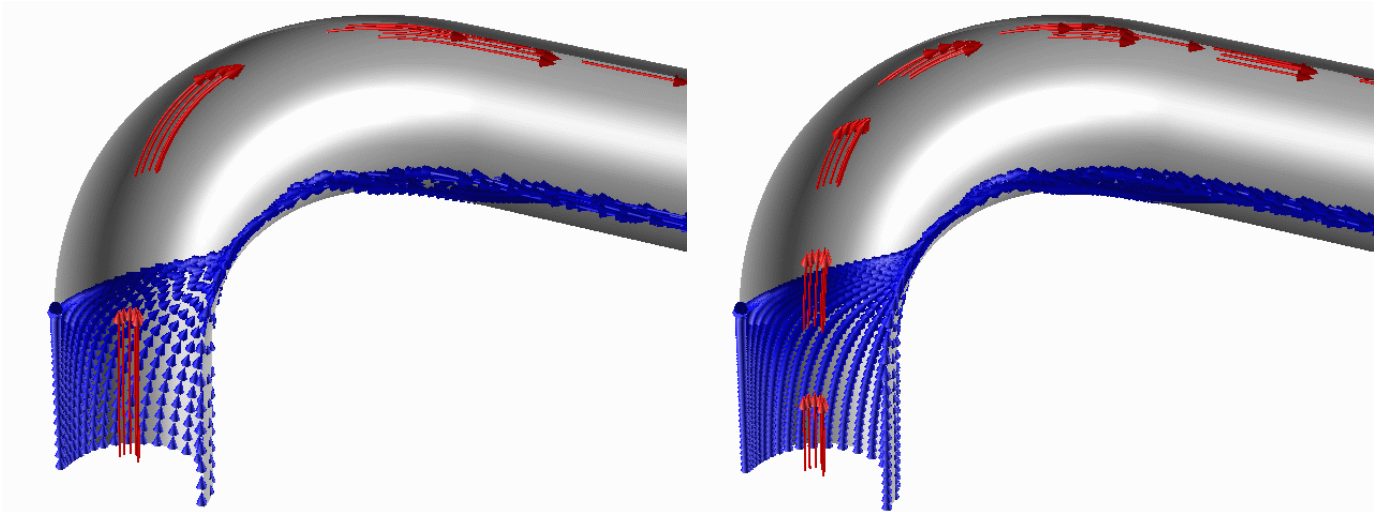
Changing the divs from a setting of 25 (at left) to a setting of 50 at right decreases the filament segment length.

Figure 161   Example showing the effect of a Streamline Div change

Auxiliary Seed Planes and Grid/Region Subsetting

Auxiliary Seed Planes consist of a single grid (for structured datasets) but can span several grids for unstructured datasets (X, Y, or Z plane). If an Auxiliary Seed Plane is created for a structured dataset (I, J, or K plane), and the grid (or region) that contains the Auxiliary Seed Plane is subsetted out, then a plane will be drawn, and seeds can be placed on the plane, but no streamlines will be calculated, since the grid (region) has been subsetted off. This has been done to avoid the confusion that might result in the creation of an Auxiliary Seed Plane, but no plane being shown due to subsetting.

If selected grids (or region) of a multi-grid unstructured data have be subsetted off, then any Auxiliary Seed Plane containing those grids will have portions turned off. If all of the grids that an Auxiliary Seed Plane contains are turned off, then no plane will be drawn.

Periodic Streamlines

For data with periodic boundaries, streamline calculation can be enabled to pass "through" periodic boundaries, and immediately re-enter the dataset on the "opposite" side. No special checks are performed to see if the boundary is truly periodic.

If DUPLICATION is set to Rotate or Translate on the Dataset Controls panel, **FieldView** will assume periodic behavior and attempt continuation of any streamlines that leave the dataset. Setting the number of copies to 1 allows periodic streamlines without any visible mirroring. The sweep angle used for rotation and the delta used for translation must be set to the correct value.

The implementation of this actually results in multiple streamlines from a given seed. Each time the streamline leaves the dataset, and periodicity brings it back in the opposite side, a new streamline is

created internally. A side effect of this is that when you delete a seed, only the first streamline "piece" is immediately removed from the graphics window. The other "pieces" that resulted from re-entry stay on-screen until a re-calculation is performed.

Note: The Ribbon display type will be changed to Complete. Another side effect is that the continuation 'pieces' may exceed the maximum number of streamlines, in which case they will be incomplete and a pop-up warning will be displayed.

Warning: If you type in an incorrect angle for rotation or delta for translation on the Dataset Controls panel, the resulting streamlines will be incorrect as well. If you set the number of copies to 1, there is no obvious way to see that the angle is wrong.

Streaklines

In order to properly display the integration of massless particles through a transient dataset, Streak-lines must be calculated. Since the position of the seed is a function of the time step, the only display types that make physical sense are Spheres, Dots and Polyspheres. Complete streaklines make no physical sense. Since each seed reaches a new position on the integration path at each time step, and only one time step is displayed in the graphics window at a time, a Complete path would be incorrect. Therefore, the DISPLAY TYPE on the Streaklines panel is forced to Spheres and disabled during the streakline calculation.

When you have streamlines displayed for a given time step or even just the seeds (i.e. the Calculate button has not been pressed yet), and the Transient Data Controls panel is brought up and the Sweep button pushed, **FieldView** will calculate Streaklines instead of Streamlines. A transient sweep will calculate streaklines on all visible, seeded rakes belonging to the current dataset. No streamlines will be computed during a transient sweep for which at least one rake will compute streaklines. Only the current dataset (as specified on the Transform Controls and Dataset Controls panels) can calculate streaklines, because only the current dataset is "swept"; any other dataset remains at its current time step.

Saving Streaklines

Important Note: Streaklines are not saved in **FieldView** memory. Hence, once a transient sweep is performed, the streakline information is lost. In order to review/browse the streaklines, they must be recalculated. Saving the streaklines to an `*.fvp` file saves the streakline computation time, which, for a large number of seeds, can be a time consuming task. In order to avoid this time consuming behavior, the option to save the streakline information as a **FieldView** Particle Path file is presented. Note that whether they are saved into an `*.fvp` file or not, the transient data needs to be swept through for you to animate the streaklines.
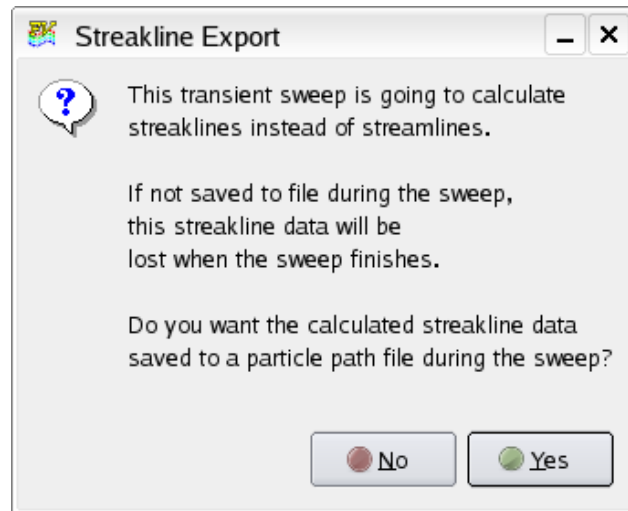
Figure 162   Streakline Export

The default Particle Path format for saving streaklines is binary **FieldView** PARTICLE SET (see **Appendix L** of the **Reference Manual**). The PARTICLE SET format creates one particle path export for each timestep. The information exported to the particle path file includes the scalar values of the particles. The export will contain dataset scalar values at each particle position for the currently loaded scalar function, even if the streamlines are geometric colored. However, dataset scalar values will not be exported if more than one rake is set to scalar coloring, and the scalar functions are not the same across all rakes. In this case, the GUI will issue a warning:

```
**** WARNING ****
The Scalar function is not the same for
all rakes. Scalar values cannot be
exported. Only "Time" will be exported.
```

If streakline data is going to be exported to a particle path file, either interactively or via the script commands (see **Chapter 5** in the **Reference Manual**), a message will be printed to the console when you are running **FieldView** (using `/usr/tmp/streak.fvp` as an example filename):

```
Streakline data will be exported to /usr/tmp/streak.fvp
```

This message is especially useful if the script command does not explicitly specify an export file name, in which case a default filename is used.

Transient Binary File Formats

The transient particle path format containing all of the particle path information for all of the time steps in a single file is best described as a STREAKLINE binary format. For cases with many time steps and

large numbers of particles per timestep, this single file can become quite large, on the order of GB in size. This format is also path-oriented - the history for each individual particle is threaded throughout the file.

To overcome the performance drawbacks associated with handling particle path data stored using the STREAKLINE format, we have introduced a binary PARTICLE SET file format. For some or all of the timesteps in a transient series, there will be a corresponding binary file containing a set of particle path information for that timestep. Association between the dataset and the particle path data is inferred from the filenames which follow the standard transient file naming conventions used by **FieldView**.

For example, if the dataset has timesteps named as follows:

```
box_injection_results0001.fvuns
box_injection_results0002.fvuns
box_injection_results0003.fvuns
...
```

Then a corresponding particle set could have the following names:

```
injection_rake_001.fvp
injection_rake_002.fvp
injection_rake_003.fvp
...
```

Data with this format is automatically recognized when read into **FieldView**.   As a result, there are no changes to the `read_particle_paths` **FVX** command. Also, RESTARTS will fully support the PARTICLE SET format.

Display type options for the PARTICLE SET format are limited to Spheres, Dots or Polyspheres. Display type options for STREAKLINE particle path data are Spheres, Dots, Polyspheres or Growing. Since both of these formats apply to transient data, animation is controlled by doing a transient sweep.

Streakline Export

Formerly, when a transient sweep was performed on a dataset with one or more streamline rakes, streakline data was saved using the STREAKLINE particle path data format.   Recognizing the considerable performance improvements that the PARTICLE SET format has to offer, we changed our default streakline export to use this format.

The PARTICLE SET format creates one particle path export for each timestep. When creating streaklines, you will need to specify a base name. The time step number will be automatically embedded using the base name, following our transient file naming conventions. Specifically, the time step number will be inserted before the first ".". If the root filename has a number in front of the first ".", we will insert a "_" character to maintain our transient file naming convention.   We will check to make sure that an exported PARTICLE SET does not already exist. If one is found, you have the option to cancel the streakline calculation or overwrite the existing transient series.

The PARTICLE SET format does not contain the "Time" scalar variable which is present in the STREAKLINE format. This scalar is considered to be redundant since the "Time" is constant for each timestep, and association between the exported streakline data and the dataset is done by matching the timestep number in the file names. The "Emission Time" scalar is stored in the PARTICLE SET format.

Since the PARTICLE SET format is now the default, an environment variable lets you use the STREAKLINE export format. Set the following variable to any value:

```
FV_SINGLE_FILE_STREAKLINES
```

Please note that streaklines created using the PARTICLE SET format cannot be read into versions of **FieldView** prior to Release 14.

Performance Optimization

An optimization dramatically speeds up the display of either streamlines or particle path data using either of the 'spheres' or 'dots' display types. In **Figure 163**, a benchmark case depicts transient particle path data derived from 64,000 emitters. Particles are released at every time step. After 100 timesteps, there are 6.4M particles being displayed for every timestep. The time needed to perform a transient sweep over the complete transient range (200 timesteps total) was measured to be over 400X faster compared to previous versions of **FieldView**.
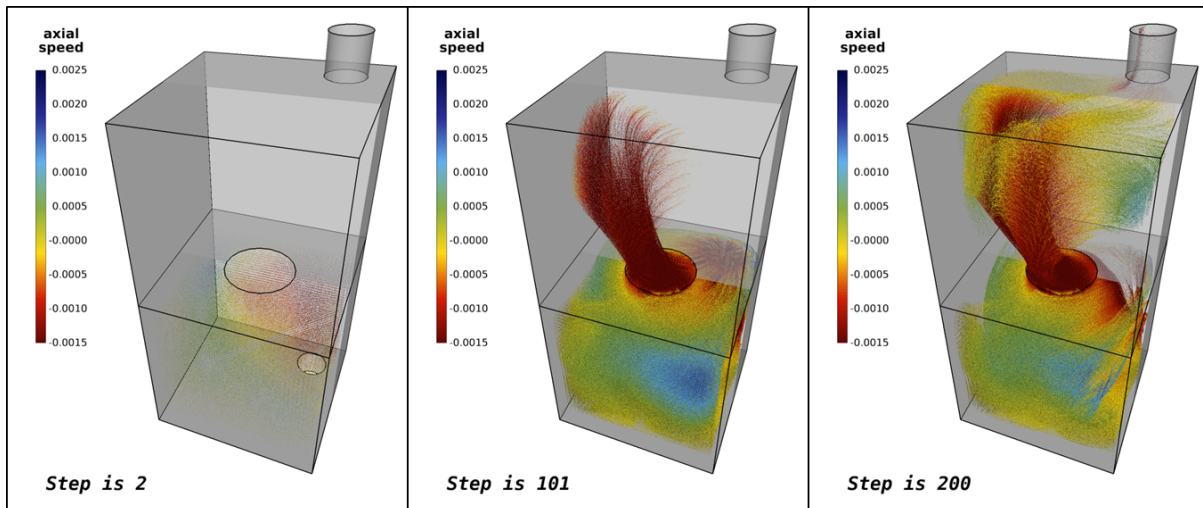
Figure 163 Transient particle path benchmark case

Performance improvements are also delivered for streakline calculations. Since streaklines are rendered (using either the sphere or dot display type) even in batch, streakline calculation times are substantially reduced. In **Figure 164**, a benchmark case depicts streaklines calculated for a biomedical application. A total of 34,560 emitters were created. Particles were emitted at every time step for the first 40 timesteps, producing 1.38 million streakline particles. The time needed to create the streakline,

performing a transient sweep over 200 timesteps in batch was measured to be over 40X faster compared to previous versions of **FieldView**.
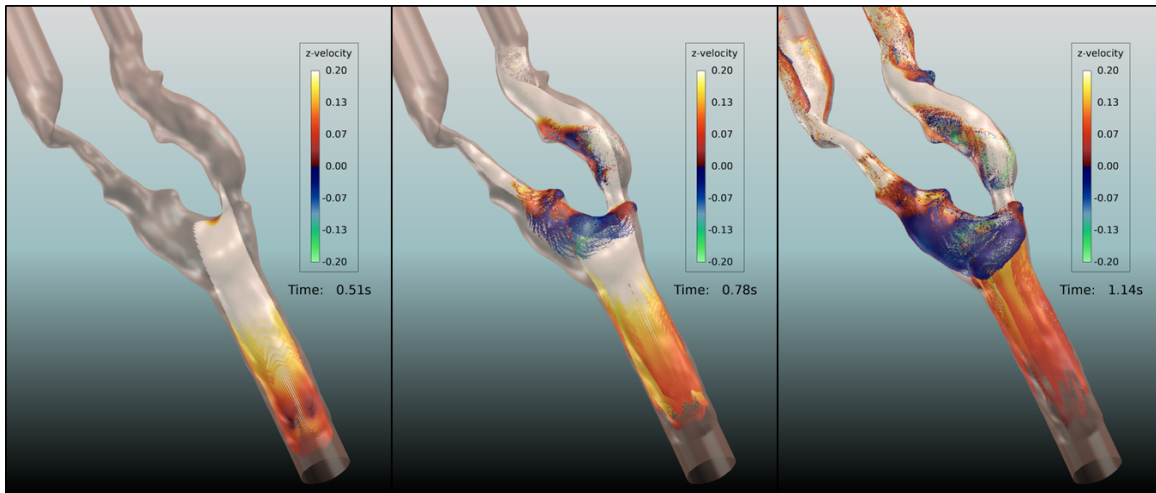
Figure 164 Streakline benchmark case

Displaying Streaklines

Once the streakline data has been exported to a particle path file, it may be read in through the Particle Path panel, only a portion of which is shown in the following figure (**Figure 165**). This portion of the Particle Path panel is used to control which variable is displayed on the particles of the streaklines themselves.

In addition, the Display Options on the Particle Path panel can be used to scale the streakline spheres according to their scalar value. (**Figure 166**). Also, DISPLAY TYPE Polyspheres can be used in cases were better visual precision is needed for understanding whether they are behind or in front of other objects. These Streakline Controls

Other streakline controls for animation are available on the Transient Data Controls panel (**Figure 167**).
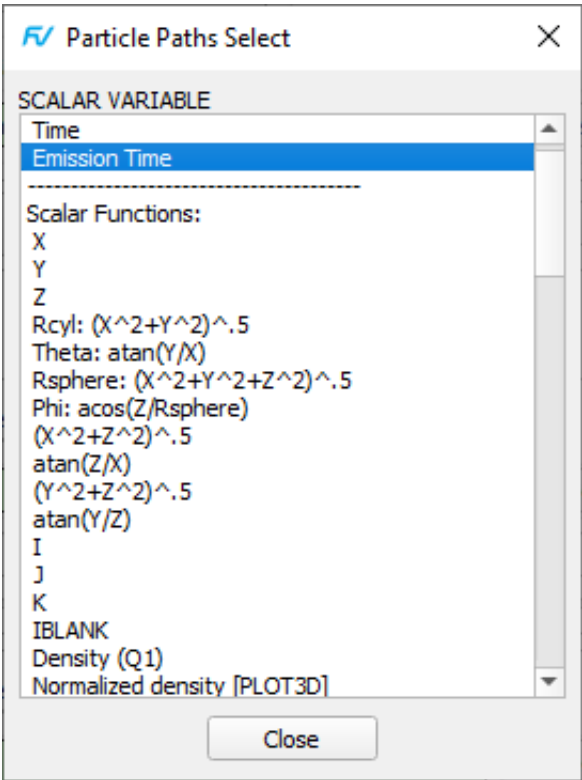
Figure 165   Streakline Display Variables
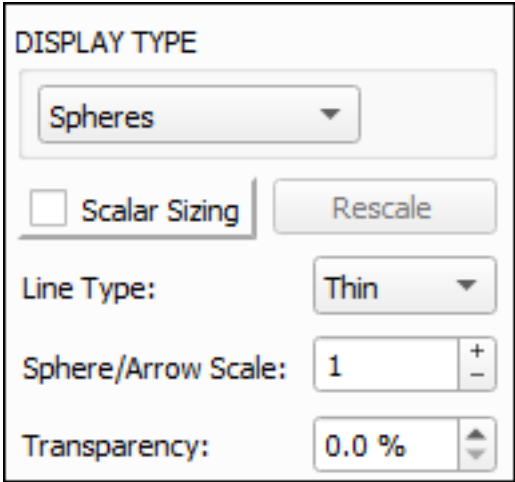
Figure 166   Streakline Scalar Sphere Controls

For streakline calculations, the Begin TIME STEP is ignored. The Sweep occurs from Current to End. Unlike a normal transient sweep, a transient sweep that is calculating streaklines sweeps from Current time step to End time step and stops automatically.

Streaklines are only calculated during a transient sweep. Changing the Current time step with the slider or the type-in field does not cause streakline calculation. In order to browse in this way, the streakline will have to be saved in a Particle Path file.
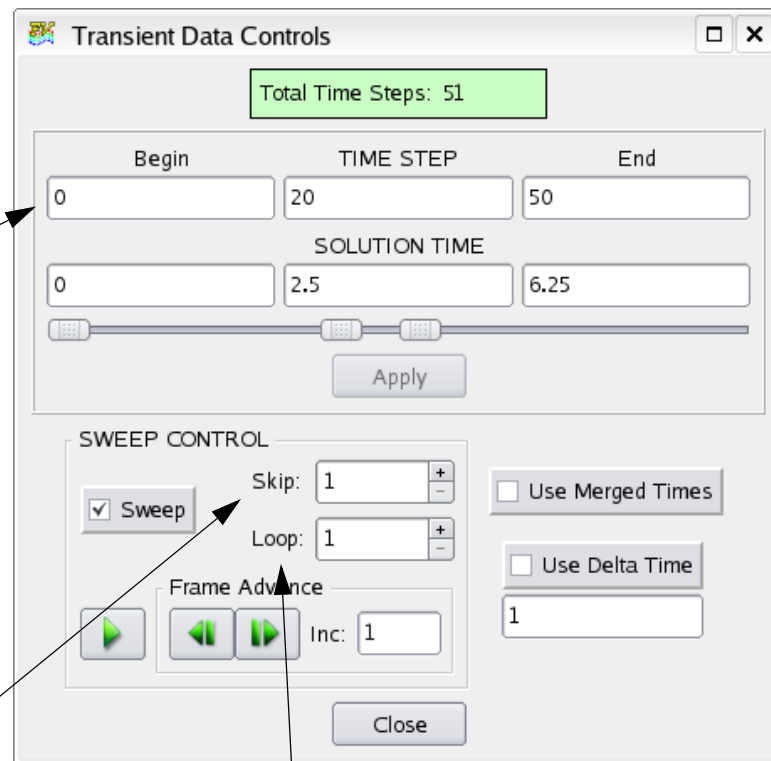
The Skip sets the increment of time steps read in. A value of 1 means read every time step; a value of 2 means read every other time step, etc.

Loop controls the number of times that the dataset is swept through. It will also update the solution time as displayed in annotation titles.

Figure 167   Streakline Controls

During the streakline calculation (transient sweep), the Transient Data Controls panel will be disabled except for the Sweep button. This is to avoid changes to the Current and End time steps, and the Skip and Loop values, all of which are used to initialize the parameters for the streakline calculation.

The streaklines are animated by the Sweep button, as their position in time always matches the time step in memory. That is, if the current TIME STEP of the transient dataset is 20 as shown in **Figure 167**, then the streakline particle positions are displayed for time step 20 only. A different time step will yield a different visualization in the graphics window and a different set of positions of the streakline particles.

Note: If the original streaklines have their Visibility on, then pressing the Sweep button will cause the Streakline Export pop-up to reappear. To avoid this, if you have read in streaklines as Particle Paths, so that you may view the Sweeping streaklines without re-saving, turn off the Visibility of the original streaklines. If the streaklines are being computed for the first time, and not displayed from an imported Particle Path file, then the original streaklines must have Visibility on, otherwise the streaklines will not be calculated.

Note: If the imported particle path (streakline) is deleted through the Particle Path panel, then changing the current time step and pressing Apply will cause the steady state streamlines to appear, but not streaklines. To recompute streaklines, the Sweep button on the Transient Data Controls panel needs to be pressed and the data swept through.

Release Interval and Duration

These streakline parameters are found at the bottom left of the Streamlines panel. Only the Calculation Parameters portion of the Streamlines panel which shows the parameter settings is shown below.
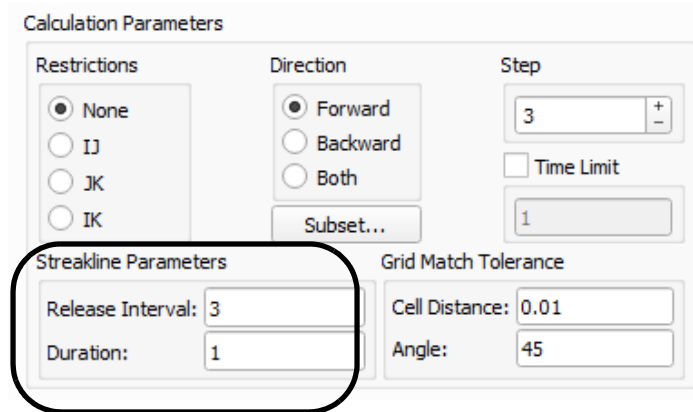
Figure 168   Streakline Parameters

The release interval controls how often to release new particles from the original seed position. The duration controls how long the particles are emitted.

A Release Interval value of 0 means release particles only at the first time step. If the Release Interval is 1, a set of new particles is released at each time step. If release interval is <= 1, Duration is ignored. The Duration must be strictly less than the Release Interval.

When release interval is > 1, the best way to understand the interaction of the two controls is in terms of an On/Off sequence:

Count the initial time step as an ON. So, for example, with a Release Interval of 5 and Duration 3, you should see an on/off sequence of 3/2:

Frame 1   ON   (place seeds at the starting position)
Frame 2   ON   (advance the previous seeds and place new seeds at the starting position)
Frame 3   ON   (advance all previous seeds and place new seeds at the starting position)
Frame 4   OFF   (only advance all previous seeds, no new seeds)
Frame 5   OFF   (only advance all previous seeds, no new seeds)
Frame 6   ON   (5 steps have occurred, release again)
Frame 7   ON   (etc…)
Frame 8   ON

Frame 9   OFF
Frame 10   OFF
Frame 11   ON   (5 steps have occurred, release again)
Frame 12   ON
Frame 13   ON

The word Frame is used to avoid confusion with time step numbering. To see how the particles are emitted, build a flipbook of the transient sweep.

Looping

Looping is only meaningful for periodic datasets, or datasets with N time steps where the solution at the N+1st time step would be exactly the same as the 1st time step.

A Loop value > 1 is most useful in allowing **FieldView** to use the periodic nature of the given time steps to extend the particle trajectory calculations past the given time steps.

With a Loop value of 1 (the default) a streakline transient sweep goes from Current to End (once) and stops. With a Loop value I (> 1), the sweep goes from Current to End I times and then stops.

On loop I (> 1), the sweep is using the periodic nature of the data to produce solution times (and time steps) *beyond* the given data. These "extended" time steps and solution times are exported to the particle path file and are reflected in Annotation Titles using the transient escape sequences %%N and %%T (see **Chapter 8** and **Chapter 14** for more information). However, the slider and text fields on the Transient Controls panel will always show the actual steps and times, not the "extended" ones. For periodic datasets, **FieldView** is not extrapolating the data, rather it is using the periodic nature to extend the data, which is physically viable since the data at the N+1st time step is exactly the same as the 1st time step, but the time step and solution times would be advanced as normal.

The extended calculation uses knowledge of the delta times between the supplied times to "extend" the solution times. If Δt is the delta time between the times for steps 1 and 2, then in subsequent loops, Δt is the delta time between the times for steps N+1 and N+2.

The "wrap-around" delta is the delta to add to the last solution time from the previous loop to get the first solution time for the next loop. The two scenarios considered are:

1. There is a constant delta between solution times: the constant delta is the wrap-around delta.
2. There is *not* a constant delta between solution times, and:

a. On the Transient Data Controls panel, when the streakline transient sweep begins, the End time step value is equal to the max time step value: a pop-up explaining why the data is not suitable for looping is presented. If OK is pressed, the sweep is continued with the sweep using a wrap-around delta computed from the last two time steps. If Cancel is pressed, the sweep is stopped altogether.
b. On the Transient Data Controls panel, when the streakline transient sweep begins, the End time step value is less than the max time step value (i.e. the End slider has been moved to the left): the delta between the solution times for the End time step and the time step after that (End+1 time step) is the wrap-around delta.

Note: The sweep always goes between TIME STEP (current) and End; the Begin time step is ignored.

The Loop value is also used when the transient sweep is not calculating streaklines. In this case, it simply allows the sweep to use the periodic nature of the actual time steps to "extend" the sweep.

For example, when exported streakline data is imported as a particle path group, you would need to set Loop to the same value (or higher) used to calculate the streaklines, in order to see the full particle path trajectory when swept from the Transient Control panel.

There are two functional differences if you use looping and are *not* calculating streaklines:
1. The looped sweep continues until the user stops it. After the I loops, it resets, starting again with actual steps/times and repeating the loops.
2. For Scenario 2b described above, the wrap-around delta computed from the last two time steps is used *without* user confirmation.

Frequently Asked Questions

*Why does **FieldView** "beep" when I try to seed particles?*
When using the interactive seeding, you must click on an object in the graphics window. If you click on empty space or not a valid object (such as a legend or title), a "beep" is sounded and the point is not seeded.

If you select an object which is not in the *current* dataset, it will also "beep".

Also, if you try to select a Coordinate or Iso-surface while the Seed Coordinates are IJK-Integer, a "beep" is sounded and seeding will not be allowed. This is because the nature of Coordinate and Iso-surfaces is that they will not, in general, have IJK-integer values, only IJK-real.

When I used streamline restriction, why didn't my streamlines appear where I expected them?
Streamline restrictions are designed to restrict a streamline to a particular I, J, K surface. If the particles are not at integer I, J, K coordinates, the resulting streamlines would not lie on the restricted surface. Thus, **FieldView** will restrict to the nearest integer I, J or K surface (which ever is appropriate). That is, if you are using IJ restrictions with seeds at K=3.2347, then the streamlines will be restricted to the K=3 computational surface.

When I used streamline restriction, why didn't my particles jump across grids?
Streamlines restrictions are designed to restrict a streamline to a particular I, J, K surface. When a streamline jumps to another grid, it may not be exactly at an integer I, J, K coordinate, and thus will not be able to be restricted to a surface. In this case, the streamline ends, with a warning message printed in the console window.

When I turned on streamline restriction, why did my Ribbons turn to Complete streamlines?

Ribbons are inherently 3-D in nature. If streamlines without surface restrictions already exist and are displayed as Ribbons. And then the streamlines are changed to a *surface restricted streamlines*, the display type is *automatically* changed to a valid type, with Complete being the most closely related type. If OK is pressed in the confirmation pop-up dialog window, then the streamlines will be redisplayed as Complete and the DISPLAY TYPE on the Streamlines Panel will change from Ribbons to Complete to reflect the accepted display.

Ribbons will also be turned to Complete in the case of periodic streamlines. See **Periodic Streamlines** for more details.

When I used add-between in IJK Real mode, why wasn't the resulting line straight?
When add-between is used in IJK Real mode, **FieldView** creates a "straight" line in I, J, K space. This line will usually not be straight in X, Y, Z space.

Why are streaklines not produced from streamline seeds when performing a transient sweep with the keyframe animator?
Streaklines force specific display settings for streamlines and disable any changes to the transient panel during the first sweep while a Particle Path file is being built (for use during subsequent sweeps). Allowing this would unnecessarily complicate the keyframe animation process. In order to display streaklines with the keyframer, you should pre-calculate them and read them in using the Particle Path Import feature. Then you can set keyframe parameters for them as particle paths. See **Chapter 7** for more information about Particle Paths.

Why do my streamlines disappear when I perform a transient sweep with the keyframer?
If a transient sweep is performed with the keyframer, then no streamlines will be shown for the current time step and all successive time steps. A streamline will only be made visible if the time step on which it was created is the current time step and that time step is not part of a transient sweep performed by the keyframer.

Streamline Error Conditions

`X, Y, Z, I, J, or K Coordinate Error`
If X, Y, Z, I, J, or K coordinates are typed in, the typed in values must be within the dataset minimum and maximum. If not, a Coordinate error will be given.

`XYZ location not found in current dataset`
It is possible to type in a value that is within the X, Y, Z minimum and maximum, yet still not have the point be part of the data. For example, if the dataset has a hole in the center, the X, Y, Z locations within the hole are within the minimum and maximum, but the point location is not valid. If a point of this type is typed in, the above error message will be displayed and the seed will not be added.

`Add between error:  Second seed must be different from the first`
When selecting end points to add a line of seeds between, the two end points must be different. If they are not, the above error message will be displayed.

```
Add between error:  For IJK coordinate seeds, the second seed must be in the
same grid as the first seed.
```
When doing an Add Between in I, J, K coordinates, the two particles must be within the same grid.

```
Interactive Seeding Error:  All Seeds on a Rake must belong to the Same
Dataset.
```
When creating seeds interactively, the seeds for each rake must lie on the same dataset.

```
Rake Creation Error:  Cannot create a Rake until a Vector Function is
Defined.
```
In order to create Streamlines, a vector function must first be defined in order to do the integration. If the vector function has not been defined in the Function Specification panel, a rake cannot be created.

Streamline Calculation Error Messages

```
particle has come to rest
```
Particle has reached a location where the vector magnitude is 0.

```
particle trace reached max number of steps: 10000
```
Each particle trace is limited to 10000 integration steps, 5000 in each direction if 'both'. To get a longer trace, reduce the Integration Step on the Calculation Parameters sub-panel.

```
particle trace stopped: too many steps in a cell
```
If a particle remains in a cell for more than 10 times the user defined integration step number, **Field-View** will assume that the particle will never leave that cell. To get a longer trace, increase the Integration Step on the Calculation Parameters sub-panel.

```
particle trace stopped: velocity too large for accuracy
```
Particle has reached a location where the vector magnitude is so large that the integration step length is 0.

```
particle trace stopped:  seed is in an IBlank=0 region
```
The seed was placed in an area where the IBlank flag is equal to 0, indicating a non-flow region. As a result, the particle trace will stop immediately.

```
particle trace stopped:  velocity is not finite
```
If the particle trace reaches a region where the velocity is not finite, the particle trace will stop.

```
particle trace stopped: too many periodic continuations.
```
FieldView advances all periodic streamlines "together", rather than calculating each single streamline all the way from beginning to end through all of its "pieces". This was done to avoid running out of space during the creation of the multiple streamlines from each seed. By advancing all of the streamlines together, each one will generate about the same number of "pieces" before running out of space.

This message means that the limit has been hit on the number of streamlines per rake when you add together the "original" streamlines (at the seed positions) plus all of their continuation "pieces" (due to their being periodic).

```
no particle trace: seed is in a subsetted off region
no particle trace: seed is in a subsetted off grid
```

The seed was placed in a region (grid) which you have turned *off* on the Subset Parameters panel.

'Tufts' - Streamlines for Transient Datasets

This feature lets you display streamlines as 'tufts' commonly used in wind tunnel visualizations. In the figure below, the first set of images show streaklines calculated for transient flow around an F16. The second set of images show time limited streamlines (where the automatic calculation of streaklines is disabled), calculated from the same set of original seeding locations, displayed as ribbons.
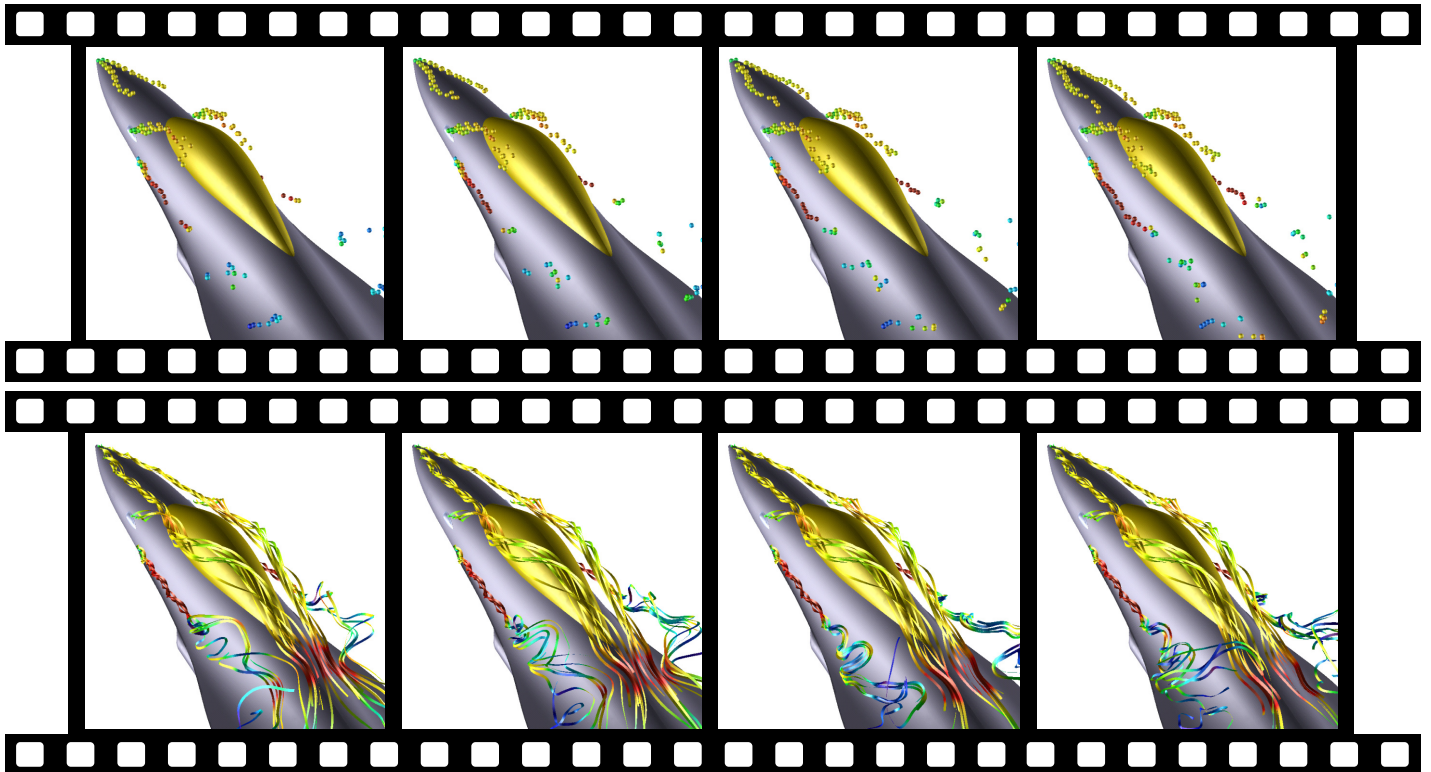
Figure 169   Streaklines (top) compared with Tufts (bottom)

To disable the automatic calculation of streaklines from streamline seeds when a transient sweep is started set the environment variable, `FV_NO_STREAK`. When this environment variable is set:

A normal transient sweep will produce streamlines.

A Keyframe transient sweep will now always produce streamlines if seeded rakes exist.

The **FieldView** Advanced Technology Viewer is unable to view transient streamlines. As a result, special handling is required for Transient XDB operations. When an XDB Build is started from the Transient panel, you will see a warning message printed to the **FieldView** Console window similar to:

```
Extracts data will be exported to /temp/kiva_transient.xdb
WARNING:  Streaklines will not be written to the XDB file.
```

When `FV_NO_STREAK` is set, then for XDB Build for a transient sweep, we will not write streamline rakes to the XDB file.

For a simple XDB write, it is possible to write the streamline rakes to the XDB file regardless of the whether `FV_NO_STREAK` is set.

Finally for an XDB Build where a surface is being swept, but the dataset is not transient, streamline rake(s) if present, will be written to the XDB file regardless of the whether `FV_NO_STREAK` is set.

Changes to Streamline Displays in FieldView 13

The "Dots" display type can be used to display streaklines, and will show performance benefits compared to "Spheres". To illustrate the differences between these display types, the same streakline rake is shown in **Figure 170** and **Figure 171**.

Figure 170 "Spheres" Display type

Figure 171 "Dots" Display type
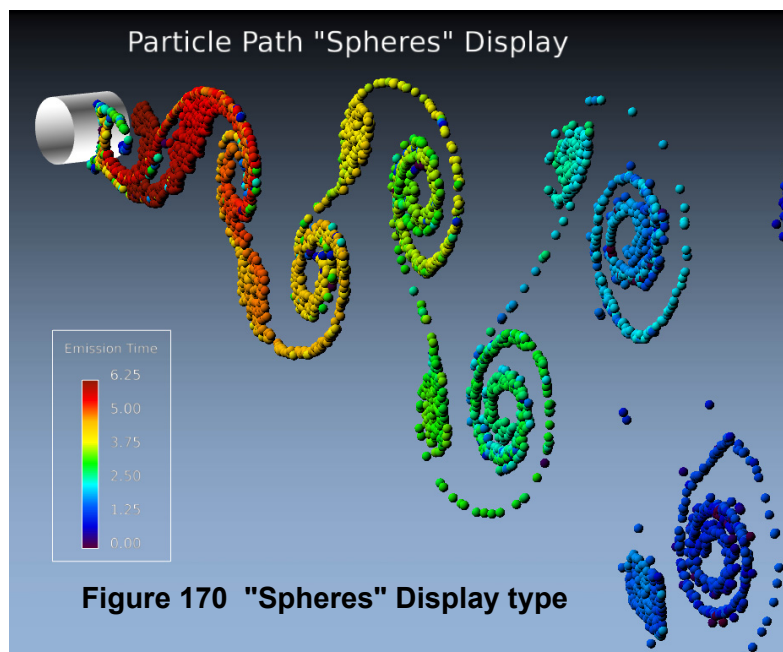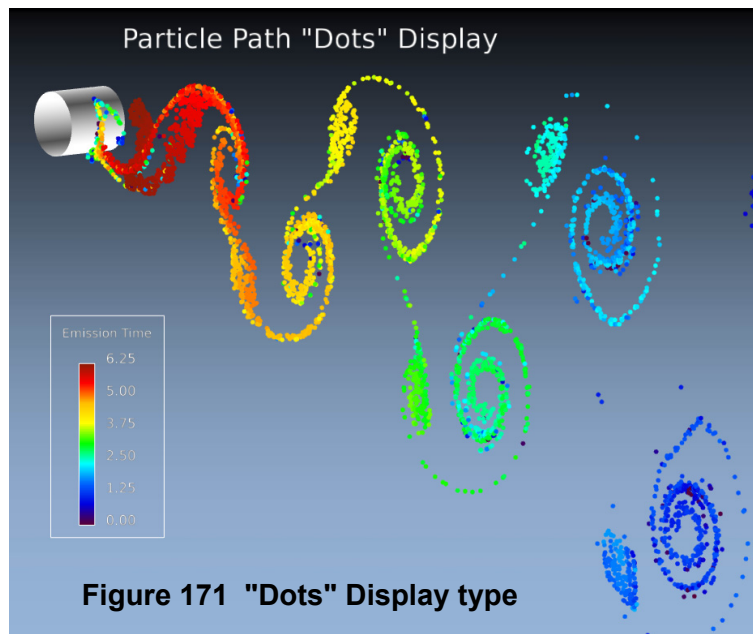
Sweep caching, allowing you to rotate your dataset while streamlines and particle paths are animating, is enabled for all of these new display types.

With **FieldView** 12, special handling was used to resize presentation rendered elements such as spheres, arrows and streamline tubes whenever the viewing distance was changed. In other words, the screen size of the spheres used in streamline displays were maintained as you zoomed in and out. This approach has two drawbacks:

- Rotating or animating streamline displays suffered from poor performance, particularly noticeable with newer graphics cards, and
- In some instances it was possible for **FieldView** to render very large spheres and streamtubes when presentation rendering was being used.

To overcome these limitations, control for resizing presentation rendering elements has been improved by adding a 'Rescale' button on the Streamline, Particle path and Vortex Core/Surface Flows panel. To obtain an acceptable size for streamtubes we recommend that you first manipulate your model to your desired view. Next, hit the Rescale button - when you do this, all of your presentation rendering streamtubes are rescaled. You can still fine-tune how your streamtubes look using the line type selections (Thin, Medium or Thick). To change the size of spheres or arrows, use Sphere/Arrow scale controls, or set the Scalar Sizing option to scale the spheres based on the scalar function.

Chapter 7

Particle Paths

7

The Particle Paths panel (**Figure 172**) is used to display particle track data read in from a file. **Field-View** will allow you to read particle path files of any format (which includes `*.fvp` - **FieldView**, `*.FDPART` - FIDAP, `*33` - STAR-CD (versions prior to 3.1), `*.trk` - STAR-CD (version 3.1 and higher), `*.trk` - CFX-4 and XDB Import) *as long as* the appropriate **FieldView** reader *is licensed*. The particles may be displayed in a number of different formats, colored and/or scaled by a scalar variable (from the flow field or the particle file), and even animated. Particle path creation is detailed in the **Streaklines** section of **Chapter 6**.

Note: If your data was read using a remote server, the browser for your particle path data will also be set for that remote server, except that, for the `fvp` particle path type ONLY, you will have an option to read from the client. In this case, the Particle Paths Data Input panel displays the following two options:

•FV Particle Path (Direct)
•Server

The Particle Paths panel can also be used to display a series of points in space, with an associated scalar variable, for displaying sparse data. This is described in **Appendix L** on the Particle Path file formats.

The Particle Paths panel is different from the Streamlines panel, which is used to calculate massless particle paths through the flow field. For buttons and features not mentioned below, please refer to **Chapter 3**.

Particle Paths Panel

Press the Import... button to select the file that contains the particle data. You will be given a choice of the type of particle file to read. Currently, 6 formats are supported: **FieldView** Particle Path, STAR-CD, STAR-CD .trk, CFX-4, FIDAP and XDB Import.

The Particle Path Display Types are: Complete, Filament, Filament & Arrows, Filament & Spheres, Growing, Spheres, Polyspheres, Lines & Spheres, Line of Spheres, and Line of Dots. Complete paths do not animate. Polyspheres are rendered spheres which provide better visual precision when understanding whether they are behind or in front of other objects.

For any Display Type that contains either Spheres or Dots, the SPHERE DISPLAY area will be enabled. Scale: allows the sphere size to be adjusted. When Scalar Sizing is turned on, the size of the sphere is based upon the variable selected in the SCALAR VARIABLE list on this panel and the value of Scale. This does not require that COLORING be Scalar.

The SCALAR VARIABLE displays the variable that the Particle Paths will be colored by, if the COLORING is set to Scalar. You may select to color the Path based upon the variable in the Scalar register, or based upon any of the variables contained in the Particle file itself. To change the SCALAR VARIABLE, press the Select… button. This will bring up the selection panel shown in **Figure 173**.

Sets the Transparency of the streamlines. A value of 0.0 is fully opaque, while a value of 1.0 is fully transparent. Valid values are 0.0, 0.125, 0.25, 0.375,…,1.0.
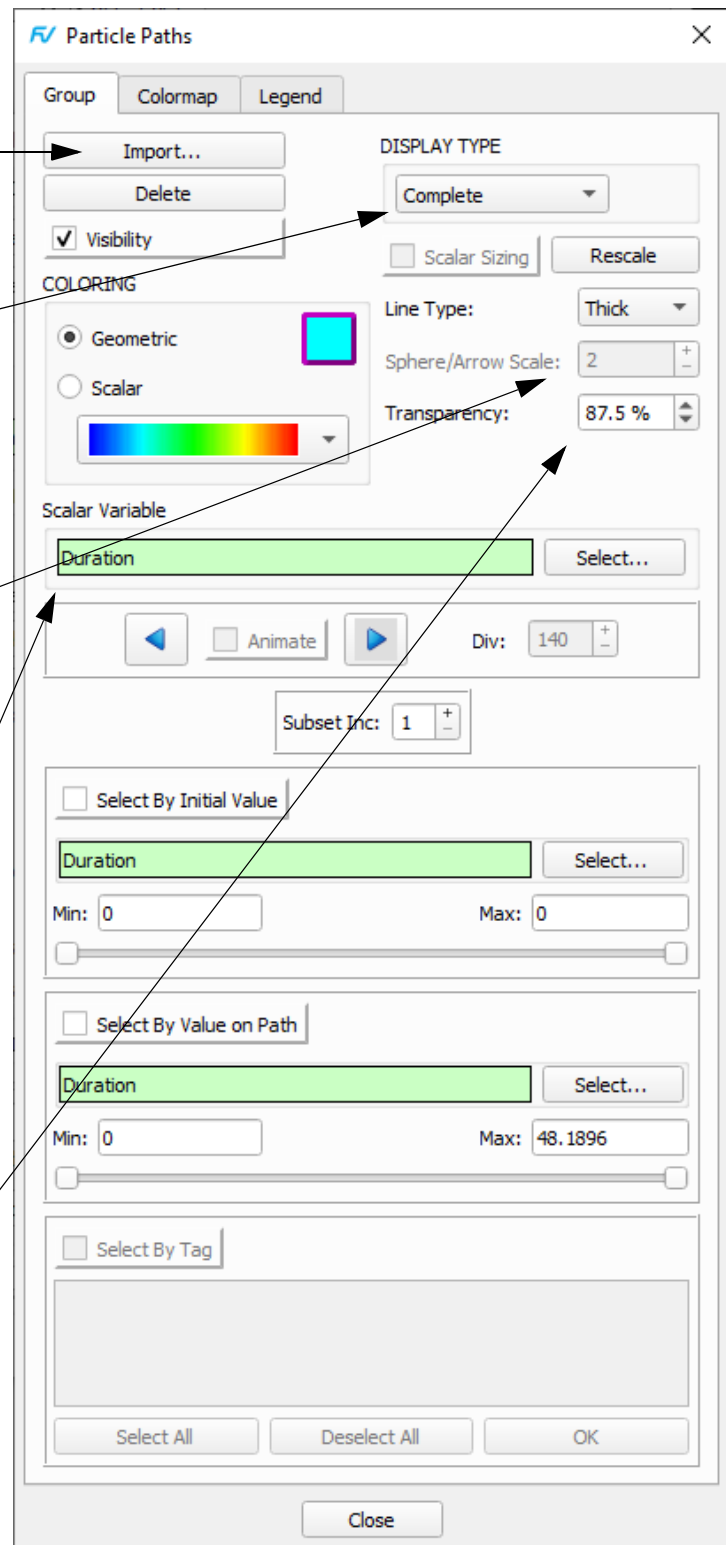

Figure 172   Particle Paths Panel

For Particle Paths, the same DISPLAY TYPE options are available as they are for Streamlines (see table on page 185), with the following exceptions:

- Type "Ribbons" is not supported for Particle Paths
- For Transient Particle Paths, only types "Dots", "Spheres" and "Polyspheres" apply, and the Panel's *Animation* button is disabled. Animation length is dependent on the number of timesteps..
- Polyspheres are rendered spheres which provide better visual precision when understanding whether they are behind or in front of other objects, at the cost of lower rendering performance.

When the Select… button in the SCALAR VARIABLE section of the Particle Path panel is pressed, the Particle Paths Select panel (**Figure 173**) is presented. This allows you to change the scalar variable that is used if COLORING is set to Scalar and/or Scalar Sizing is enabled for any Spheres DISPLAY TYPE. The Scalar register variable listed will dynamically change if the Scalar register function is changed.

Note on Lines of Dots
For the highest performance rendering speed animating steady state (non-transient) particle paths, we recommend using DISPLAY TYPE "Lines of Dots."

This section will show the variable that is currently loaded in the Scalar Register. The possible scalars include any intrinsic, user-defined or solver variable.

This section displays all of the path and dataset variables that may be displayed along Particle Paths data file.
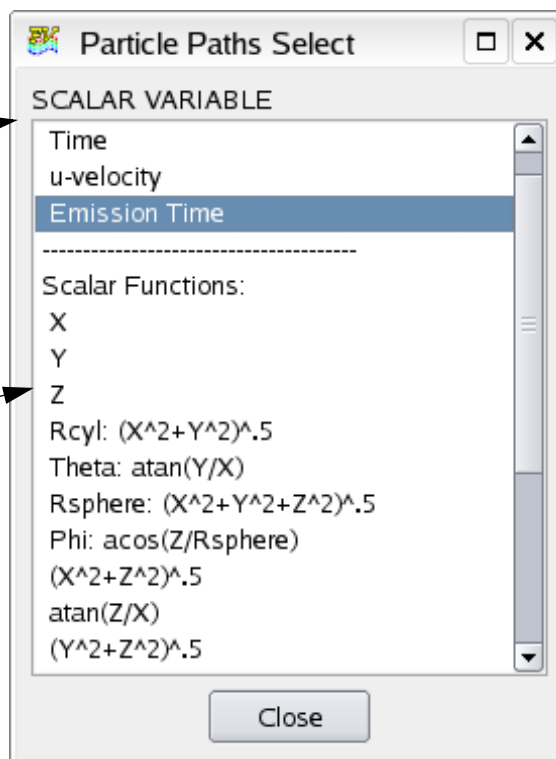

Figure 173   Particle Paths Select Panel

For the above example, the scalar variable `Emission Time` has been selected. If COLORING is set to Scalar, the imported particle paths will be scalar colored showing the particle emission time.
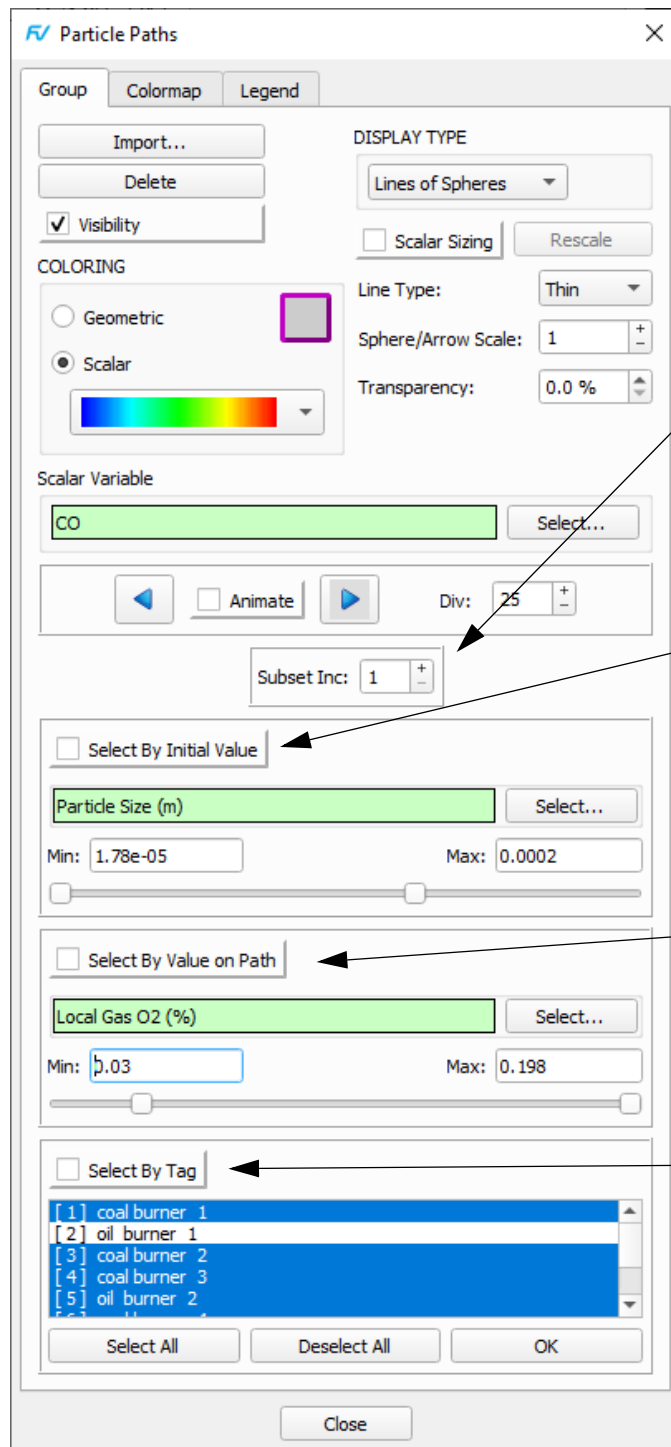
SUBSET allows the user to select out the paths to be displayed. The Inc: amount indicates whether paths should be skipped (not visualized) or not and by how much. An Inc: 1 means to show all of the available paths. An Inc: 2 value means show every other path, etc.

SELECT BY INITIAL VALUE allows you to subset (not threshold) the paths by the initial value of any available path variable. Both the Min: and the Max: values may be separately adjusted. The Select… button allows you to choose the desired variable (**Figure 175**).

SELECT BY VALUE ON PATH allows you to subset (not threshold) the paths by the value anywhere on the path of any available path variable. It is selected and adjusted in the same way as SELECT BY INITIAL VALUE.

SELECT BY TAG (ASCII Particle Path format files only) allows you to subset (not threshold) paths based on the tag which labels the particular path. Tag information is read from the Particle Path file and must be specified when creating the file before using in **Field-View**. All or any number of the Tags may be selected.

Figure 174   Particle Path SUBSET and SELECT BY Fields

The particle path data that has been imported into **FieldView** may be sub-setted by any VALUE ON PATH of the particles. That is, one of the particle path variables (read from the file) can be used to filter out certain paths. The filters available through the Select By panel are the SUBSET BY INITIAL VALUE or the SUBSET BY VALUE ON PATH (shown).
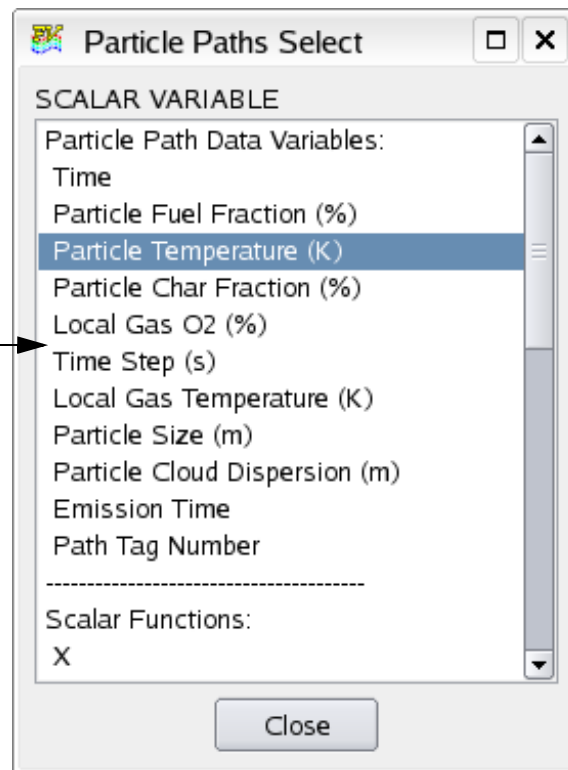
Figure 175   Particle Paths Select By Panel

When the Select… button in the SELECT BY INITIAL VALUE or SELECT BY VALUE ON PATH sections (described below) of the Particle Path panel is pressed, the Particle Paths Select By panel (**Figure 175**) is presented. This allows you to change the variable that is used if subsetting of the paths is desired. The path variables, along with time, as given in the imported Particle Path file, will be listed.

In the above example, the path variable `Local Gas O2 (%)`, from the Particle Path file, has been selected and can now be used to subset by VALUE ON PATH. Each of the two SELECT BY types can have its own unique path variable, or the same path variable can be used for both.

SUBSET
The Subset area on this panel is used to decrease the number of particle paths displayed by skipping over paths. When Subset = 1, all particle paths are displayed. When Subset = 2, every other path is displayed (1, 3, 5, 7, etc.). The default Subset value is 1 as shown in **Figure 174**.

SELECT BY INITIAL VALUE
The SELECT BY INITIAL VALUE area on this panel allows you to control which paths are visible based on the criterion of the initial value of the path. Each particle path contains XYZ locations and path variable values at those locations. The first XYZ location for each particle path defines the initial location. The value of each path variable at this initial location is the initial value of the path (for this

variable). You can change the path variable filter by pressing the Select… button. This will bring up the panel shown in **Figure 175**. Remember that the SELECT BY INITIAL VALUE button needs to be turned on for this feature to function.

The Min: and Max: values can be adjusted. By default, they will show the absolute minimum and absolute maximum for all paths. Adjusting the Min: and/or Max: sliders will cause **FieldView** to draw only paths with initial values in the specified range anywhere on the path.

In **Figure 174**, the value of Max: is set to 0.0002. This means that if there are any paths with an initial value of Particle Size (m) greater than 0.0002, they not be shown, i.e., the entire path will be hidden.

> Note: If the initial value falls within the range set by the values of Min: and Max:, then the entire path will be displayed. If the initial value falls outside the range set by the values of Min: and Max:, then the entire path will be hidden. There are no circumstances for which adjusting the Min: and Max: values will cause **FieldView** to display only a portion of the path.

SELECT BY VALUE ON PATH

The SELECT BY VALUE ON PATH area on this panel allows you to control which paths are visible based on the criterion of the value anywhere on the path. Each particle path contains XYZ locations and path variable values at those locations. You can change the path variable used as the filter by pressing the Select… button. This will bring up the panel shown in **Figure 175**. Remember that the SELECT BY VALUE ON PATH button needs to be turned on for this feature to function.

The Min: and Max: values can be adjusted. By default, they will show the absolute minimum and absolute maximum for all paths. Adjusting the Min: and/or Max: sliders will cause **FieldView** to draw only those paths that contain values within the specified range.

In **Figure 174**, the value of Min: is set to 0.03 and the value of Max: is set to 0.198. This means that if there are any values of `Local Gas O2 (%)` between 0.03 and 0.198 anywhere along the path, the entire path will be drawn.

Note: A path is hidden if all values on the path are outside of the set Min: and Max: range.

SELECT BY TAG

(ASCII Particle Path format files only) The SELECT BY TAG area on this panel allows you to control which paths are visible based on the tag for each path. There can only be as many tags as paths, although more than one path can be assigned the same tag. (Please refer to **Appendix L** of the **Reference Manual** for more information on the Particle Path format and path tagging.)

The tags found in the particle path file will be displayed in the window. By default, none of the tags will be selected. You have the option to Select All, Deselect All or to select any number of them by clicking on them individually. In addition, remember that the SELECT BY TAG button needs to be turned on for this feature to function. The slider bar on the right of this field can be used to navigate longer lists.

Only those paths that have been assigned to the selected tags will be displayed. Therefore, Select All will, by default, show all paths, and Deselect All will show no paths. You must press OK for changes to be effective.

In **Figure 174**, assuming no tags other than those visible are selected, only the paths assigned tag numbers 1, 3 and 4 will be displayed. Any paths that do not have these tag numbers assigned to them will not be displayed, despite any other settings on the panel, as long as the SELECT BY TAG button is on. Also, if Tag Names are specified, **FieldView** automatically provides the path variable `Path Tag Number`.

Functionality
If Time is one of the variables supplied by the particle path import file, **FieldView** will automatically provide a Particle Path Data Variable called Emission Time (in the SCALAR VARIABLE section of the Particle Path panel). The Emission Time of a path is the time associated with the first particle in the path.

Transient Particle Paths

This feature allows you to import transient particle path data. It works if both the solution and particle path files have time values present.

Particle paths will use real solution times to control the animation of these display types under the following conditions:

1. The solution times are provided upon particle path data import.
2. There is one and only one group of paths with supplied solution times.

If these conditions are met, the Div: field is forced to the number of time steps (i.e., the number of solution times) and is disabled.

The binary **FieldView** particle path file provides these solution times. Hence, if a transient sweep of 50 time steps exports streakline data to a particle path file that is subsequently imported, the Div: field is set to 50. See **Chapter 6** for more information about streaklines and export of streaklines into a particle path file. For a description of the binary particle path format, see **Appendix L** of the **Reference Manual**.

If you have transient data, you can create a Binary **FieldView** particle path file (`*.fvp`) by creating Streaklines. Streaklines should be saved as particle paths files, otherwise once created, they have to be calculated over again to be visualized. If they are saved as a particle path file, once read in as such, they may be displayed with all the particle path options and re-animated as desired. See **Chapter 6** for more information about this capability.

'Growing' Display Type for transient STREAKLINE Particle Paths

Particles can be rendered as growing lines, even for transient datasets. This will let users visualize not only the current particle's location but also its trajectory over time. This can be done with the Growing Display Type option. This capability is limited to Particle Paths written to the STREAKLINE format, i.e., for which all time steps have been written to a single file. It is not supported for the high performance

PARTICLE SET format, which stores no information about trajectories. For particles computed in **FieldView** with the Streamlines panel, the environment variable `FV_SINGLE_FILE_STREAKLINES` can be set to write particles information to the single file STREAKLINE format. The growing lines are synced to the solution time. A sphere indicates the start of each growing line.
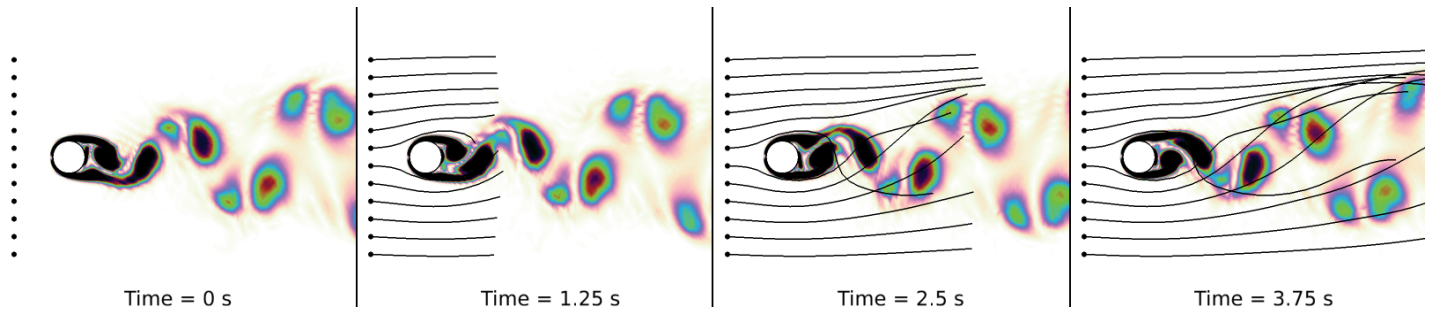

Time = 0 s   Time = 1.25 s   Time = 2.5 s   Time = 3.75 s

Figure 176 Growing lines for an unsteady dataset

It is not recommended to use the "Presentation Rendering" mode with a large number of growing lines, as the resulting shaded tubes will be slow to render and may require a large amount of memory.

Frequently Asked Questions

What are Particle Paths, and how do they differ from Streamlines?
Particle Paths are pre-computed particle results that have been read from a file. The position of the particles can be displayed and animated, and the particles may be colored by Particle Path variables (those contained in the imported file) or the contents of the Scalar register (i.e., any intrinsic, user-defined or solver variable). **FieldView** currently supports **FieldView** Particle Path (`.fvp`), STAR-CD, STAR-CD `.trk`, CFX-4, FIDAP and XDB Import formats. The **FieldView** Particle Path format can be used to read particle paths from other sources.

Streamlines are massless particles that are computed by **FieldView**. The position of the seed points are specified, and then the particles integrated through the velocity field to compute the streamlines. For more information, see **Chapter 6**.

How can I display my Particle Paths?
First, you must specify the file that contains the particle path information. This is done by pressing the Import... button on the Particle Paths panel. The default file filter on the Particle Paths Data Input panel is `*.fvp`, although there is no required filename convention for particle path files. Next, specify the file that contains the particle data, and press OK. Once the data has been read in, the paths will be displayed in the graphics window.

Note: Only one set of Particle Paths is allowed per dataset. You will be prompted for confirmation if re-importing on the current dataset.

What controls the color of my Particle Paths?

If you change COLORING from Geometric to Scalar on the Particle Paths panel, the paths will be colored by the variable shown in the SCALAR VARIABLE section of the panel. This is different from all other visualization panels in that the paths may be colored not only by the contents of the scalar register, but also by any of the variables in the Particle Path input file.

To change the variable controlling the color of the paths (or Scalar Sphere Sizing), press the Select… button on this section of the panel. This will give you the option to use the current Scalar Register function, or one of the functions from the Particle Path file (see **Figure 173**). If you wish to change the Scalar Register function that colors the paths, use the Function Specification panel on the main menu to alter the contents of the register.

How can I control the size of my Particle Path spheres?

If you select a display type that contains spheres, the Sphere Display area of the Particle Paths panel becomes active. This allows you to change the size of the sphere by a single scale factor, or to have the size of the spheres be based upon a variable. If you turn Scalar Sizing on, the size of the spheres will be controlled by the variable selected in the Scalar Variable section of the panel. You may also adjust the overall scale factor, even with Scalar Sizing turned on.

Which Particle Path formats are supported?

Currently, six particle path formats are supported: **FieldView** Particle Path (`.fvp`), STAR-CD, STAR-CD `.trk`, CFX-4, FIDAP and XDB Import.

The **FieldView** Particle Path format describes a set of paths, with each path consisting of a series of points. For each point along a path, an X, Y and Z value is specified, describing the location of the point. In addition, a number of variables may be defined at that location. The format is described in **Appendix L** of the **Reference Manual**.

Can I use Particle Paths to display stationary "point data" in FieldView?

Yes. **FieldView**'s Particle Path import format can be used to read in point-data such as experimental data which can then be displayed as stationary particles. A complete description on how this can be done is given in **Appendix L** of the **Reference Manual**, which describes the two **FieldView** Particle Path formats.

Chapter 8

Annotation

8

The Annotation panel (**Figure 177**) is used to create and display annotation objects of two different types: text or arrows. Text is useful for labeling the graphics window or a particular area of interest and arrows can be placed and rotated into position to point out an area of significance.

The color of the current annotation object is controlled by the Geometry Color sub-panel produced from the Colormap Specification panel. For buttons and features not mentioned below, please refer to **Chapter 3**.

Annotation Panel

The Annotation panel may be used to Create either a line of Text or an Arrow.

To place text in the graphics window, click Create Text to bring up the Annotation Create sub-panel, type into the blank line and click OK. To change the text in the graphics window, edit under Edit Text: then press Enter.

Font controls determine the font type and size for the current text. There are 21 different fonts available (see **Figure 178**). The font size can be controlled by using the +/- buttons or by typing directly into the text field. The font size can range from 1 to 100, with 10 being the default size. These fields will only have an effect when the Annotation type is Text.

The position of the text may be controlled interactively, or by using the buttons in the Align area. The text may be placed flush left, flush right, centered horizontally, and may be placed at the top, center or bottom of the graphics window. The direction of the text may also be changed from Horizontal to Vertical. These buttons will only have an effect when the Annotation type is Text.
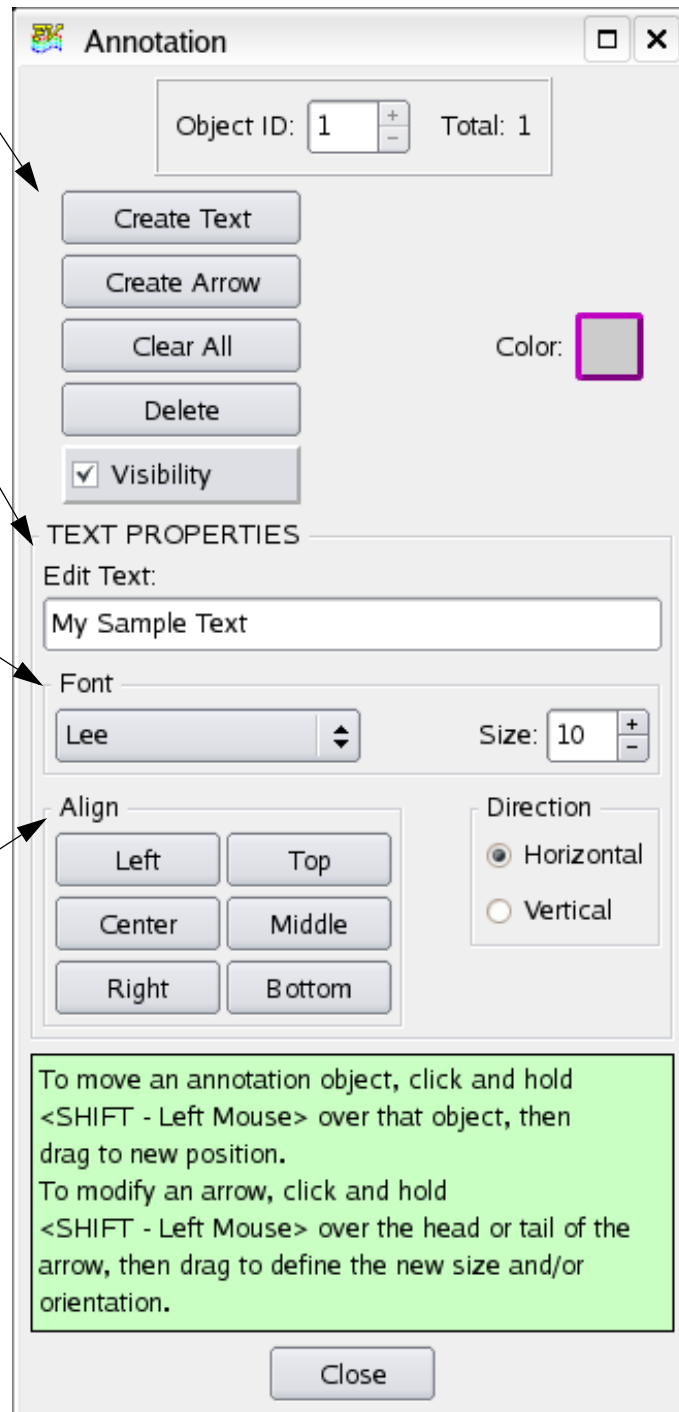
Figure 177 Annotation Panel

A number of high quality fonts are included with this release. The default font is based on an automatic detection of the system's local setting. The Noto font provides international characters. Support for international characters has been accomplished through a multibyte character implementation using UTF-8, an encoding of Unicode. **FieldView** will remember the settings on all font GUIs (including the Annotation panel, the Legend tab, and the Coordinate Surface panel Ruled Grid Options), and use them as defaults in the next session.

These fonts will yield different results from a standard alphabetic typeface. The Math Lower Case and Math Upper Case fonts are shown in **Appendix M** of this **Reference Manual** associated with the keys of a standard US keyboard.
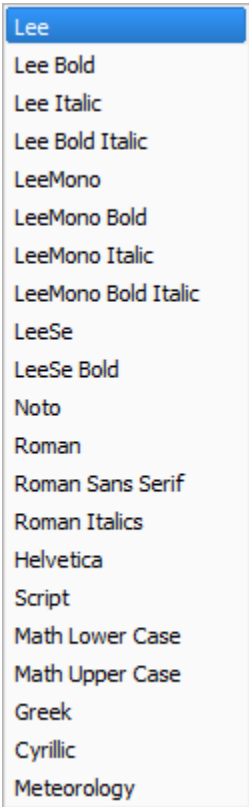
Lee
Lee Bold
Lee Italic
Lee Bold Italic
LeeMono
LeeMono Bold
LeeMono Italic
LeeMono Bold Italic
LeeSe
LeeSe Bold
Noto
Roman
Roman Sans Serif
Roman Italics
Helvetica
Script
Math Lower Case
Math Upper Case
Greek
Cyrillic
Meteorology

Figure 178 Font Popup List

Frequently Asked Questions

How do I create an Annotation?
To create text, click Create Text to bring up the Annotation Create sub-panel, type into the blank line and click OK.

To create an arrow, click Create Arrow and follow the instructions in the green popup.

How can I position the text?
The position of the text may be controlled interactively, or by using the buttons in the Align area to position the text at the left side, right side, top, bottom, or in the center (horizontal) or middle (vertical) of the screen.

How do I create multi-line titles?

Each time a new title is created after the first one, it will be placed below the previously created title. Thus you can easily create multi-line titles by creating each line in succession. After the first line is created and placed as desired, **FieldView** will automatically place successively created titles below it. Of course, any title can be moved to any desired position. Multi-line titles cannot, however, be moved as a group.

How do I control the display of the text?
Using the Font popup list you can select from 21 different fonts, and you can control the size of the font by either typing into the Size field, or pressing the + or - keys. The default Font is based on an automatic detection of the system's local setting, and the default Size is 10. Text may also be created vertically, rather than horizontally, by selecting the corresponding button in the Direction area. For the mapping of the Math Lower Case and Math Upper Case fonts to the standard US keyboard, see **Appendix M** of the **Reference Manual.**

Advanced Annotation

Advanced annotation features let you display the title along with the time for a specific calculation or dataset. Use the `"%%"` escape sequence within titles as a way to show the time. For example, if your dataset is transient and you type in the following title:

Time Step = `%%N`, which is equal to `%%T` seconds

you may see a title in the graphics window as follows:

Time Step = 3, which is equal to 10.36 seconds.

These titles will, of course, update during a sweep of transient data.

Displaying the Dataset
To specify a dataset, you can type in the dataset number in the string after the character notation and the annotation will be attached to that specific dataset.

For example:

Dataset 1, Time Step = `%%N1`, which is equal to `%%T1` seconds

will display:

Dataset 1, Time Step = 3, which is equal to 10.36 seconds

The time step and solution time in this case will be the values associated with Dataset 1.

The following table shows a complete list of title escape sequences.

%%CURRENT	Show current coord, iso or comp value (For Legend Annotation ONLY)
%%SCALAR_FUNC	Show the current scalar func on the legend (For Legend Annotation ONLY)
%%ALPHA	"Alpha" constant for current dataset
%%DN	Display the date ie 12/4/2021 *
%%DT	Display the date ie December 4, 2021 **
%%FSMACH	"Free Stream Mach" constant for current dataset
%%FUNCTION	Name of function file
%%GRID	Name of grid file
%%N	Time step, transient dataset
%%NAMFUNCTION	Name of function name file
%%RE	"Reynolds No" constant for current dataset
%%RESULT	Name of results file
%%SERVER	Server name used to read data
%%T	Solution time, transient dataset

* (equivalent to UNIX "date +'%x %X'")

** (equivalent to UNIX "date +'%B %d %Y, %I:%M:%S %p '")

Note: All escape sequences are case sensitive.

The user can again display the titles for a given dataset simply by specifying the dataset number after the notation. For example, %%RE1 will display the Reynolds number for Dataset 1 and %%RE2 will display the Reynolds number for Dataset 2, etc.

Displaying the Grid Number
The **FieldView** constants escape sequences %%RE, %%FSMACH, and %%ALPHA also have an optional G number modifier which follows the optional dataset number modifier. This G number specifies a 1-based, dataset-relative grid number. If not specified, then Grid 1 is assumed. This is done to account for the case where individual grids within a dataset may have differing values for the standard constants. For example, %%RE1G3 will display the Reynolds number for Grid 3 in Dataset 1.

Special Numerical Annotation (Escape Sequences)

An option to format numerical values is also available. Format specifiers (FORTRAN style E and F masks) for numerical values are supported.

Number Formatting
The maximum number of characters displayed and what is displayed to the right of the decimal point can be controlled. The basic format is `F#.#` where the first number specifies the maximum number of characters that can be displayed and the second number is the total number of characters that will be displayed to the right of the decimal place.

For example,
If the title `%%T1` displays 0.00369019, then `%%T1F12.6` will produce 0.003690

Exponential Formatting
Exponential formatting can also be applied to a numerical title. The basic format is `E#.#` where the first number specifies maximum number of characters that can be displayed and the second number the value of the exponent.

For example,
If the title `%%T1` produces 0.00369019, then `%%T1E12.6` will produce 0.369019E-02
In this case, the total display will be limited to a maximum length of 12 characters.

If an escape sequence which produces a floating point numerical value is not followed by the "E" or "F" FORTRAN-style format masks, the numerical value is output using the default C-style "g" formatting, which is similar to FORTRAN "G" formatting. Also, the E and F format specifiers will display "*" (asterisks) if the number exceeds the specified field width.

Special Note for KIVA Users: When using the k2fv converter to convert otape9 files to a series of **FieldView** unstructured files, the crank angle is stored as the **FieldView** constant, ALPHA. The escape sequence titles will therefore let you display crank angle as a title which can be updated during a transient sweep.

Options for Displaying the Path
The **FieldView** filename escape sequences `%%GRID`, `%%RESULTS`, `%%FUNCTION`, and `%%NAMFUNC-TION` have an optional "P" modifier, which follows the optional dataset number modifier. Here, we show some examples:
    %%GRID   ..will show the grid file name for Dataset 1 (the default dataset)
    %%GRID2   ..will show the grid file name for Dataset 2
    %%GRID2P   ..will show full path to the grid file name for Dataset 2

Note on Restarts
Note that the date and time escape sequence titles are taken at the moment that the title is created within the **FieldView** session. For RESTART support, the actual value of the date and time is saved to the titles RESTART, instead of the escape string. This ensures that the RESTART exactly re-creates the picture, including the date and time that it was created. So,

unlike the other escape strings, the date title will be STATIC - title restarts save the STATIC date and not the escape sequence.

Concerning the `%%SERVER` escape string, if a dataset was read in Direct mode, `%%SERVER` will simply be replaced by the string "Direct".

Changes to Fonts in FieldView 13

We are introducing several new fonts which can be used for annotation and legends. The new fonts, illustrated exactly as they appear, are shown in **Figure 179**.
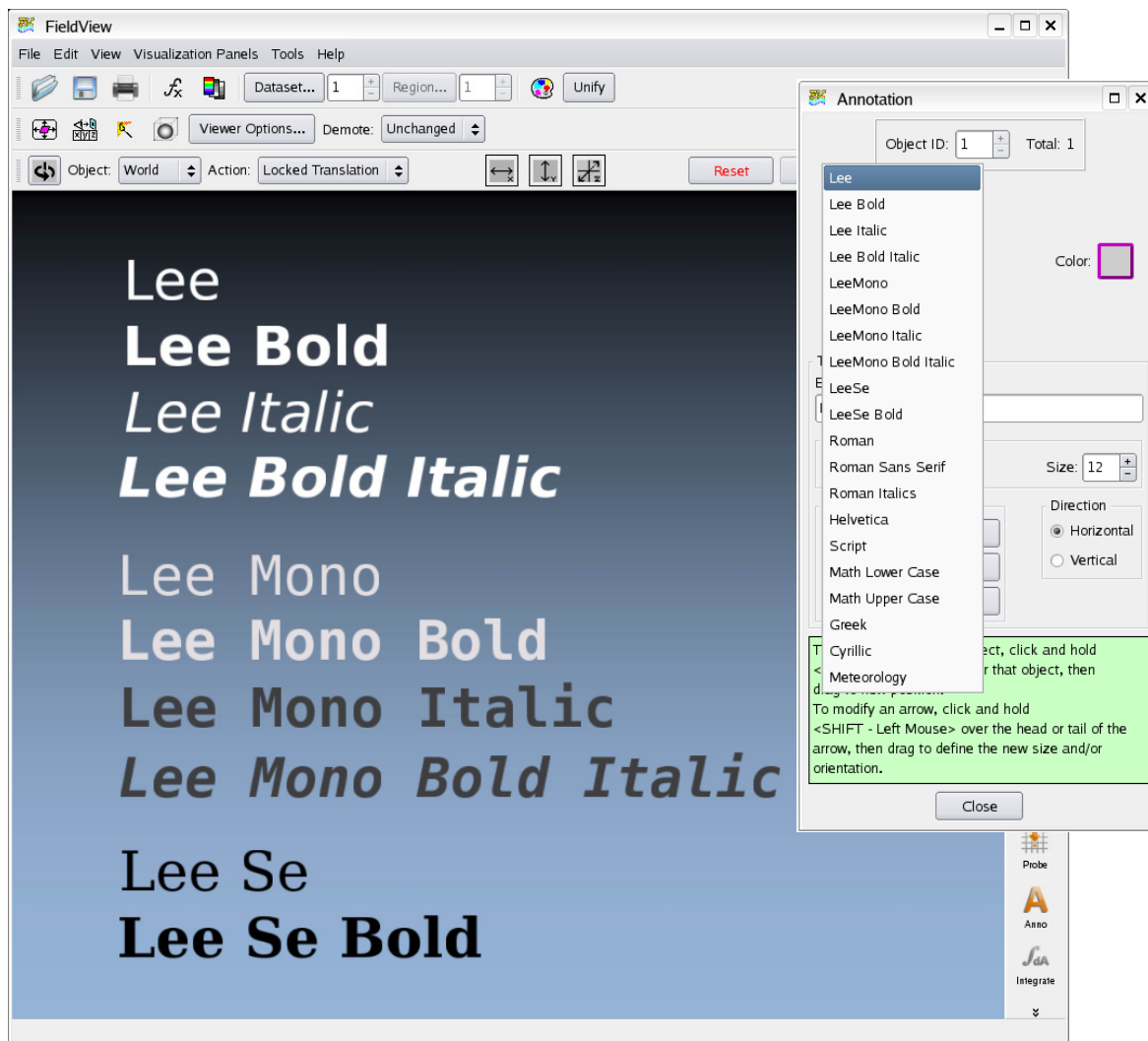

Figure 179 High Quality Fonts

Full backward compatible support is included for our older fonts - this is displayed in the Annotation font pull down menu above.

Chapter 9

Coordinate Surface

9

The Coordinate Surface panel (**Figure 180**) is used to create surfaces of constant X, Y, or Z. Since these surfaces will always be planar (unless data is being displayed in RTZ Coordinate - see Cylindrical Notes below), the various shading options will have little effect on the surfaces. In fact, there is no option for Faceted shading since, for a planar surface, it is identical to Smooth shading.

Except for the shading differences, noted above, Coordinate Surfaces have all of the same display options as Iso-Surfaces. For buttons and features not mentioned above, please refer to **Chapter 3**.

Exporting: Coordinate surfaces can be exported using the Tools/Export command. For information on the format, please see **"Coordinate Surfaces" on page 525** of the **Reference Manual**.

Arbitrary Planes: Coordinate surfaces lie on either the X, Y or Z axis and cannot be positioned arbitrarily. For creating a arbitrary plane surface, see the Iso-Surface **"Cutting Plane Panel" on page 173** of the **Reference Manual**.

Coordinate Surface Panel

Figure 180 Coordinate Surface Panel

This area defines the current value for the coordinate surface. After pressing create, X, Y, or Z must be chosen to define the surface. For the dimension that is chosen, the location of the surface can be controlled by either moving the slider or typing into the Value field. The Min and Max fields will control the bounds of the animation for the surface.

For the dimensions that are not chosen, the Min and Max can be adjusted to clip the surface in either direction. These can be adjusted by either typing in exact values, or moving the slider bars. The Value field has no effect for the non-selected dimensions.

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ sliders become RTZ (Radius, Theta, Z) sliders. See **Chapter 3** for more information.

Displaying Cell Volumes

The cell volumes of a Coordinate surface (often referred to as a Crinkle surface) are easily displayed in **FieldView**. Crinkle surfaces allow users to visualize surfaces by showing the complete elements (or cells) that intersect either a coordinate or iso-surface. They permit an examination of mesh quality along with CFD simulation results on an element basis. This capability is an essential pre-requisite for extending **FieldView** functionality to display things like cell metrics, and ultimately, cell-based data. Boundary surfaces and computational surfaces are both defined at element faces, so these surfaces are not considered for this feature.

Crinkle is a DISPLAY TYPE option in addition to Constant, Smooth, Mesh, Contours, Vertices and Shaded Vertices. Surface attributes such as Show Mesh and Show Contours are also supported. The illustration (**Figure 181**) shows the Crinkle display type for two coordinate surfaces, and the smooth display type for two additional coordinate surfaces.
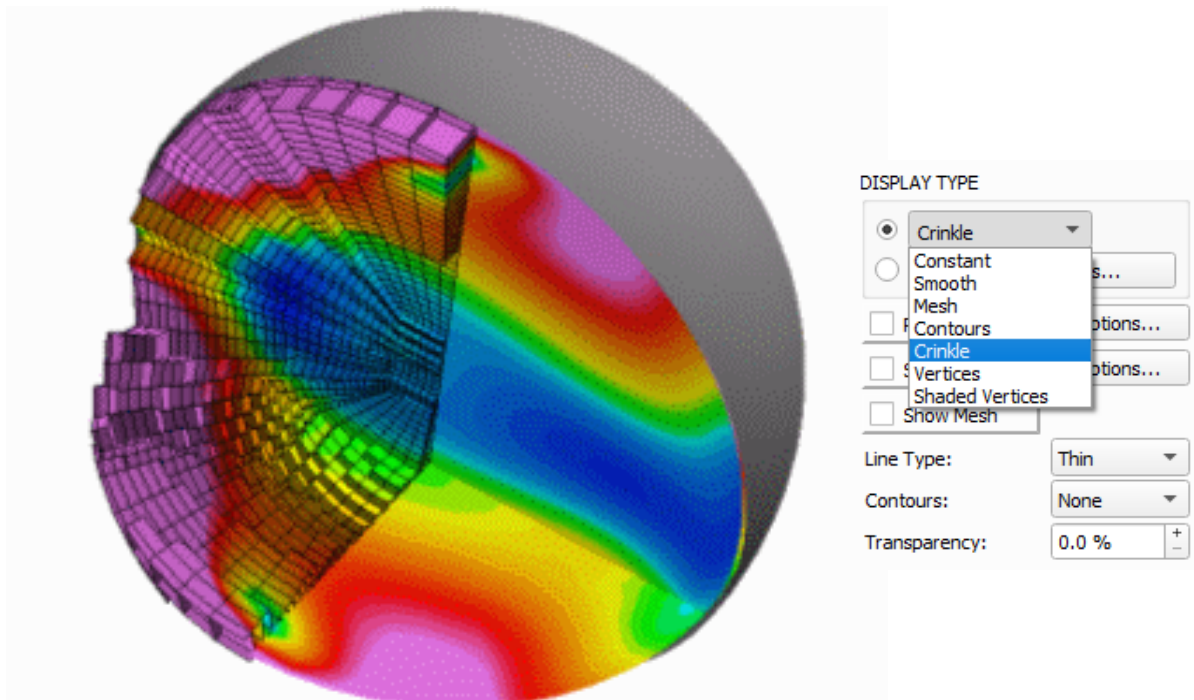
Figure 181 Crinkle Surface and DISPLAY TYPE menu selection

Displaying Crinkle Surfaces

First create a surface inside on the Coordinate surface panel and then choose Crinkle in the Display Type menu.

The Crinkle display type is not available if the surface is Coordinate 'R' or 'Theta' (cylindrical dataset).

Ruled Grids

Ruled grids can be displayed on coordinate planes as an attribute setting, similar to Show Mesh. An illustration of a ruled grid is shown below:
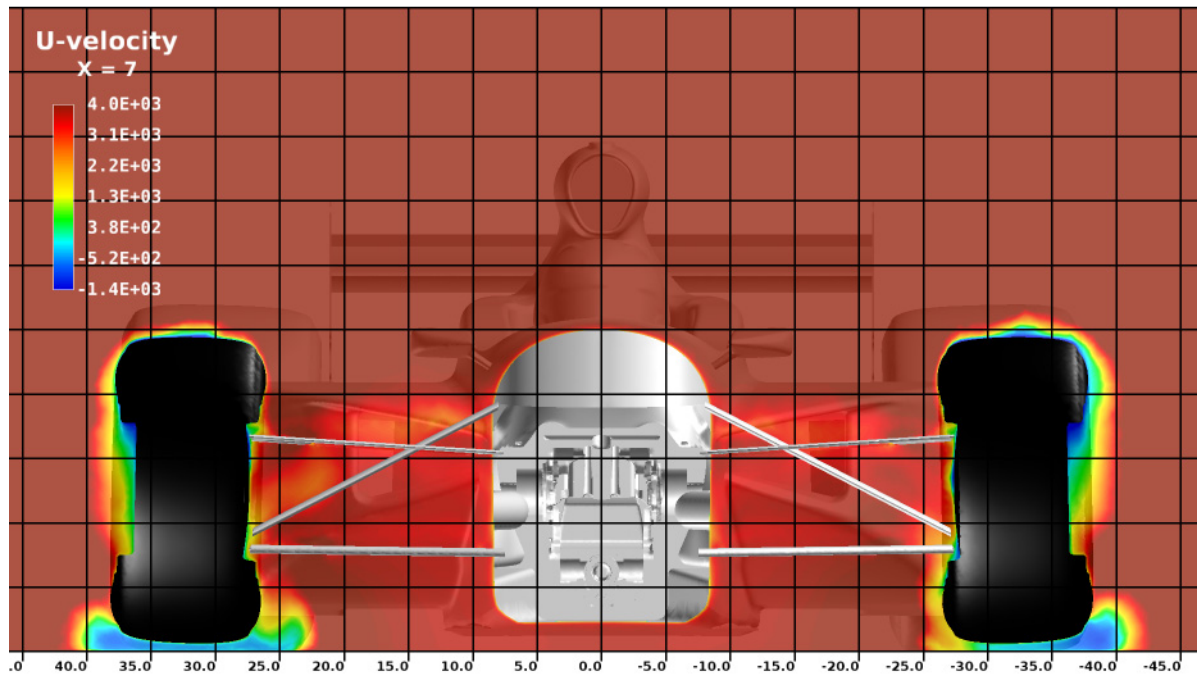

Figure 182 Ruled grid for a Formula 1 simulation

To display a ruled grid on a coordinate plane, turn on the Ruled Grid button on the Coordinate Surface panel, as shown in **Figure 183** below.
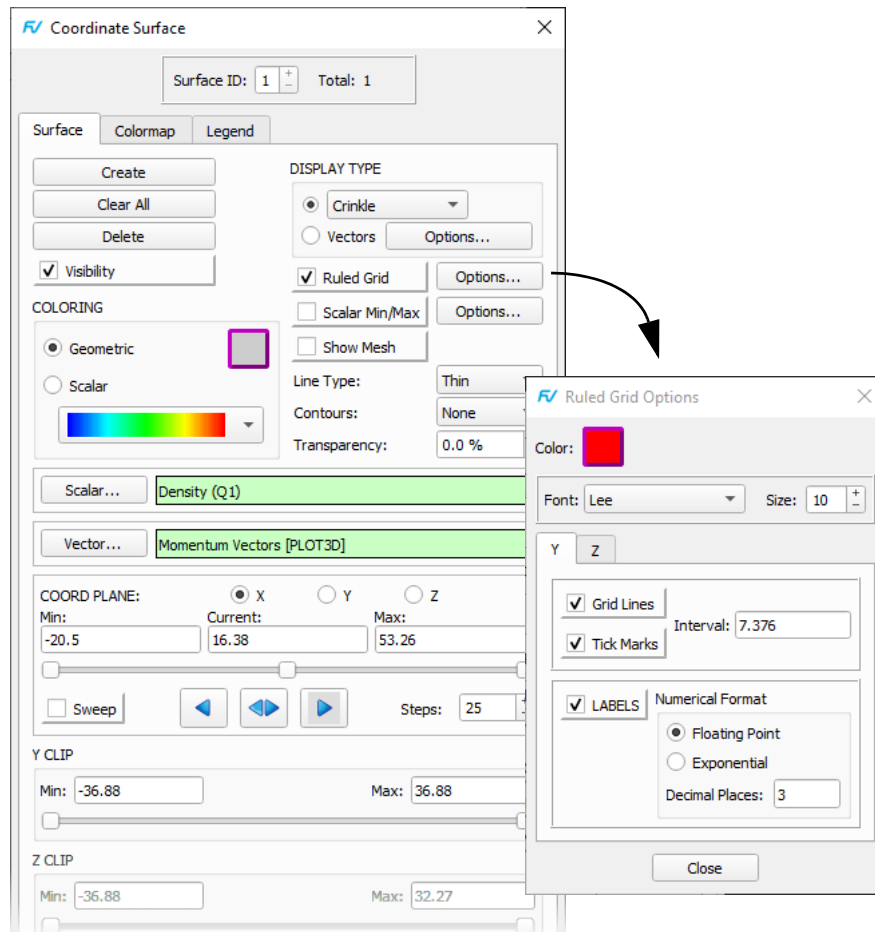
Figure 183 Turning Ruled Grid On and Ruled Grid Options

When a ruled grid is first created, a default interval is calculated independently for each axis, dividing the model extents by 10 using the model units. To control how the ruled grid appears, click on the Options... button to the right of the Ruled Grid button. Color and Font settings apply to both the 'vertical' and 'horizontal' grids on the coordinate surface. For each of the grid axes, independent control lets you toggle the visibility for grid lines, tick marks and labels. The same Numerical Format options for legends give you further control on how the labels appear on the ruled grid. Labels appear at the minimum extent of the coordinate surface - at present it is not possible to change where the labels are drawn on the coordinate plane.

Setting a value of '0' for the interval is not permitted; if attempted, the interval is re-initialized dividing the axis extent by 10. It is possible to specify such a small value for Interval that the number of ruled grid lines to be drawn would exceed the internal limit. If the limit is set to be too small, only one grid line will be drawn. To draw a grid with more than one line at this point, increase the interval size.

Coordinate clipping, such as Y and Z CLIP for an X coordinate surface, lets you control the range of the ruled grid on the coordinate surface.

Thresholding and dynamic clipping do not get applied to the ruled grid. Using thresholding it is possible to remove some or all of the coordinate surface, leaving the underlying grid unchanged.

When applied to a coordinate surface, transparency is not applied to the ruled grid on a coordinate surface. If the transparency is set to 100%, the ruled grid will be displayed even though the coordinate surface will be completely invisible. It will also 'stay' with the coordinate plane during a sweep operation.
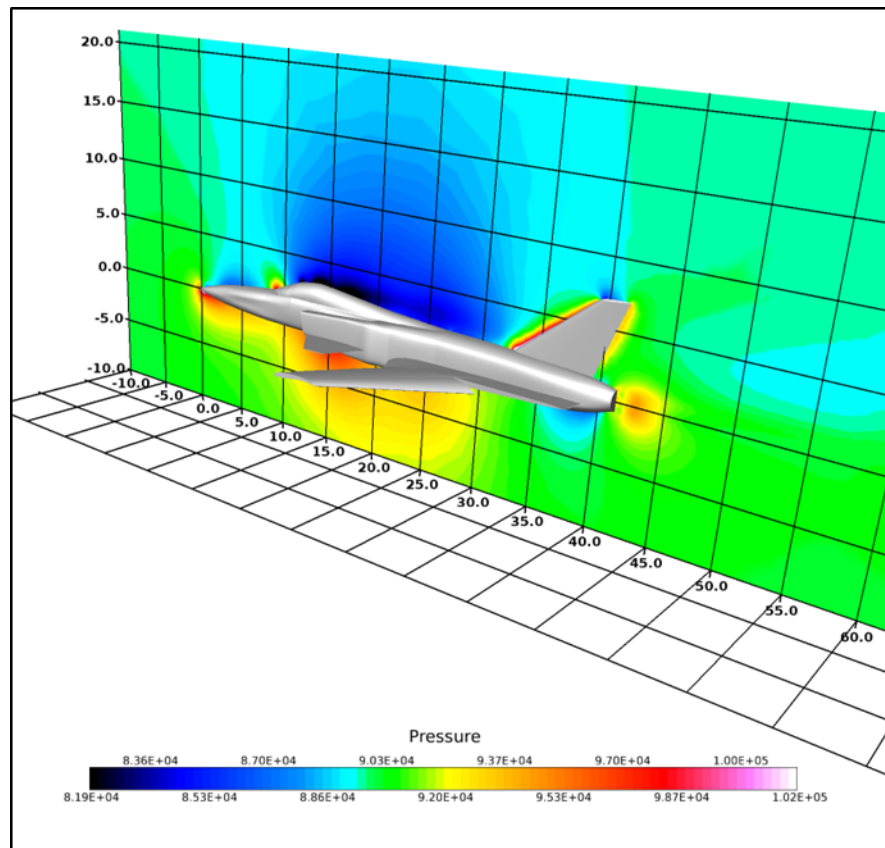
Figure 184 Use of a ruled grid for boundaries

If the coordinate surface is transformed, the ruled grid transforms with it; there is no restriction on keeping ruled grids normal to the graphics window. Labels on the ruled grid are transformed automatically to ensure readability regardless of surface orientation. In **Figure 184** we show how multiple coordinate surfaces, each displaying a ruled grid, can be used to show three-dimensional ruled boundaries.

Once ruled grid settings have been created for a coordinate surface, they are inherited by the next coordinate surface created.

Applying the UNIFY button when the current coordinate surface has a ruled grid displayed will UNIFY the ruled grid settings for all other coordinate surfaces.

Important Notes and Limitations

Ruled grids are available for coordinate planes only and are limited to datasets with XYZ coordinate systems.

Dataset transforms such as scaling and duplication will be applied to ruled grid labels. This will produce undesirable results for labels and if dataset transforms are required, we recommend that labels are turned off for ruled grids.

Frequently Asked Questions

How do I create a Coordinate Surface?
In order to create a surface, simply press the Create button. A coordinate surface will be created on the *current* dataset. Once created, it cannot be "moved" or "switched" to a different dataset. It will have to be deleted and created anew on another dataset. The first time it is pressed, an X surface will be created, with its value at half the distance between the minimum and maximum X. If an X surface is not desired, simply press the button marked Y or Z to change the surface. If a second surface is desired, again first press the Create button. Now, a copy of the previous surface will be created. To change this surface, either move the appropriate slider bar to a different value, or choose a different type of surface to display.

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ sliders become RTZ (Radius, Theta, Z) sliders. See **Chapter 3** for more information.

Transient Note: If you have set clipping on any coordinate surface during any of a dataset's time steps, clipping will be maintained as the surface is swept through time.

How do I animate the surface?
To animate the current surface, press the button marked Sweep in the Sweep Control area. By default, the animation will move forward and start again at the minimum when it reaches the specified maximum. You may also select the animation to be backward or to bounce between the two by pressing on the arrow keys in the Sweep Control area. Pressing on these arrow keys will not only change the direction of the animation, but will also move the plane one step in that direction. The number of steps the animation will take between the minimum and maximum is controlled by the Steps field in the Sweep Control area.

What are the display options for a Coordinate Surface?
Coordinate surfaces may be colored by the current Geometry color, by a Scalar color, or shown as Vectors. If the Display Type is not Vectors, they can be shown with Constant (no lighting) or Smooth shading, as a Mesh, Contours, Crinkle surface, Vertices or Shaded Vertices. Mesh and Contours may be shown in conjunction with any Display Type including Vectors.

How can I show fewer vectors on my Coordinate Surface?

For Coordinate Surfaces, you may show a uniform sampling of vectors across the surface. To do this, bring up the Vector Options sub-panel. On this panel, turn on the button marked Uniform Sampling. By default, 10 vectors will be shown in each direction. To change this value, type the desired number into the type-in fields and press the OK button.

How can I clip the current surface to only show a particular area?
The Coordinate Surface that is created will show the full bounds in the two non-selected dimensions. For example, an X surface shows all Y and Z points for the X value selected. If you would like to clip the plane in one of the other directions, you simply move the minimum or maximum slider (or type into the appropriate field) for the dimension desired.

Chapter 10

Boundary Surface

10

The Boundary Surface panel (**Figure 185**) is used to show the various boundary types (e.g. wall, inlet, outlet, baffle) defined when the data was input. Structured Boundary files enable boundaries on structured data (see **Appendix H** of the **Reference Manual** for more information).

Different types of boundaries may be grouped together and displayed as one surface, with all of the standard display options available for other surfaces. Boundary Surfaces have an additional Display Type that is not available for other surfaces, called Outlines, which will display the edges of the Boundary Surface. For buttons and features not mentioned above, please refer to **Chapter 3**.

Keyboard shortcuts simplify the selection of entries within a list. Use [CTRL][M1] to select individual entries, or select a range of entries with [M1] to mark the start of the range and [SHFT][M1] to mark the end.

Exporting: Boundary surfaces can be exported using the Tools/Export command. For information on the format, please see **"Boundary Surfaces" on page 527** of the **Reference Manual**.

Note: If the data reader did not define any boundary types on input, pressing Create on the Boundary Surface panel will produce the pop-up error message:

```
There are no boundary types defined for any datasets.
```

Note: Outlines will display only as geometry colored with no mesh or contour overlays and is ignored for integration. It is intended as a visual aid only.

Boundary Surface Panel

DISPLAY TYPE options for a Boundary Surface are: Constant, Faceted, Smooth, Mesh, Contours, Outlines, Vertices and Shaded Vertices.

Flip Front/Back - When turned ON, the front and back surfaces of collocated faces will be flipped. Note that if the two sides of surface do not belong to the same Boundary Surface object, this setting needs to be consistent across these two boundary surfaces for proper rendering. (This feature is currently limited to Unstructured Boundary surfaces.)

Displays all of the Boundary Types that have been read from your input file, in alphabetical order. After pressing Create, you must select one or more boundary types to make up the boundary surface. The number of faces for each boundary type will also be listed.

Click [M1] on the list to begin a group selection.

Use [SHFT][M1] to select the end of the range.

Use [CTRL][M1] to select individual entries from the list.

Use [CTRL][SHFT][M1] for multiple group selections.

See the section **Working with Filters (Search Bars)** in **Chapter 3** for information on how to use this search bar to filter boundary type names.

Tabs are provided to set Threshold range, Clipping by geometric X, Y, Z, Theta or R ranges, or to access *Sweep* con-

After pressing Create, and selecting the boundary type(s), you must press OK to display the surface. Other actions, such as changing the Display Type, act as an implicit OK and will cause the surface to be visualized.
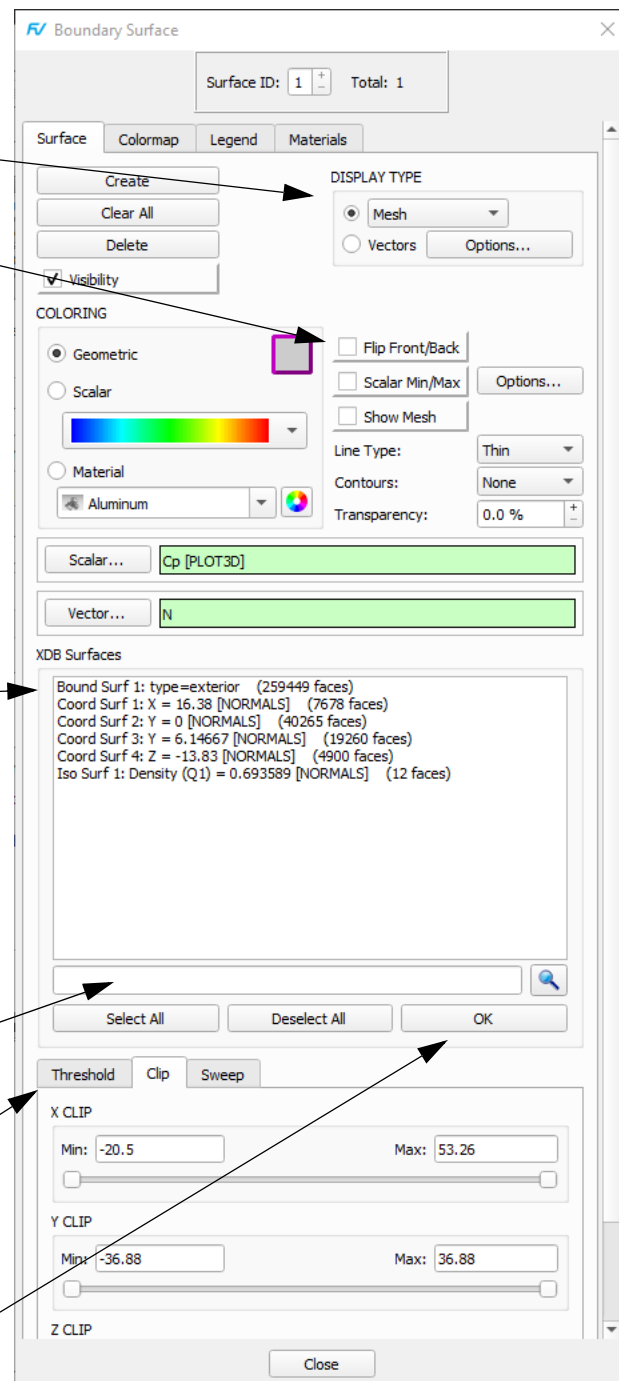
Figure 185 Boundary Surface Panel

Boundary Type Selection

The search bar on the Boundary Surface panel facilitates selecting Boundary Types from long lists using a text string filter; see **Working with Filters (Search Bars)** in **Chapter 3** for further information.

Boundary Types are sorted alphabetically for all datasets. Keyboard shortcuts may be used to select individual entries within a list [CTRL][M1] or a range of entries [M1] and [SHFT][M1].
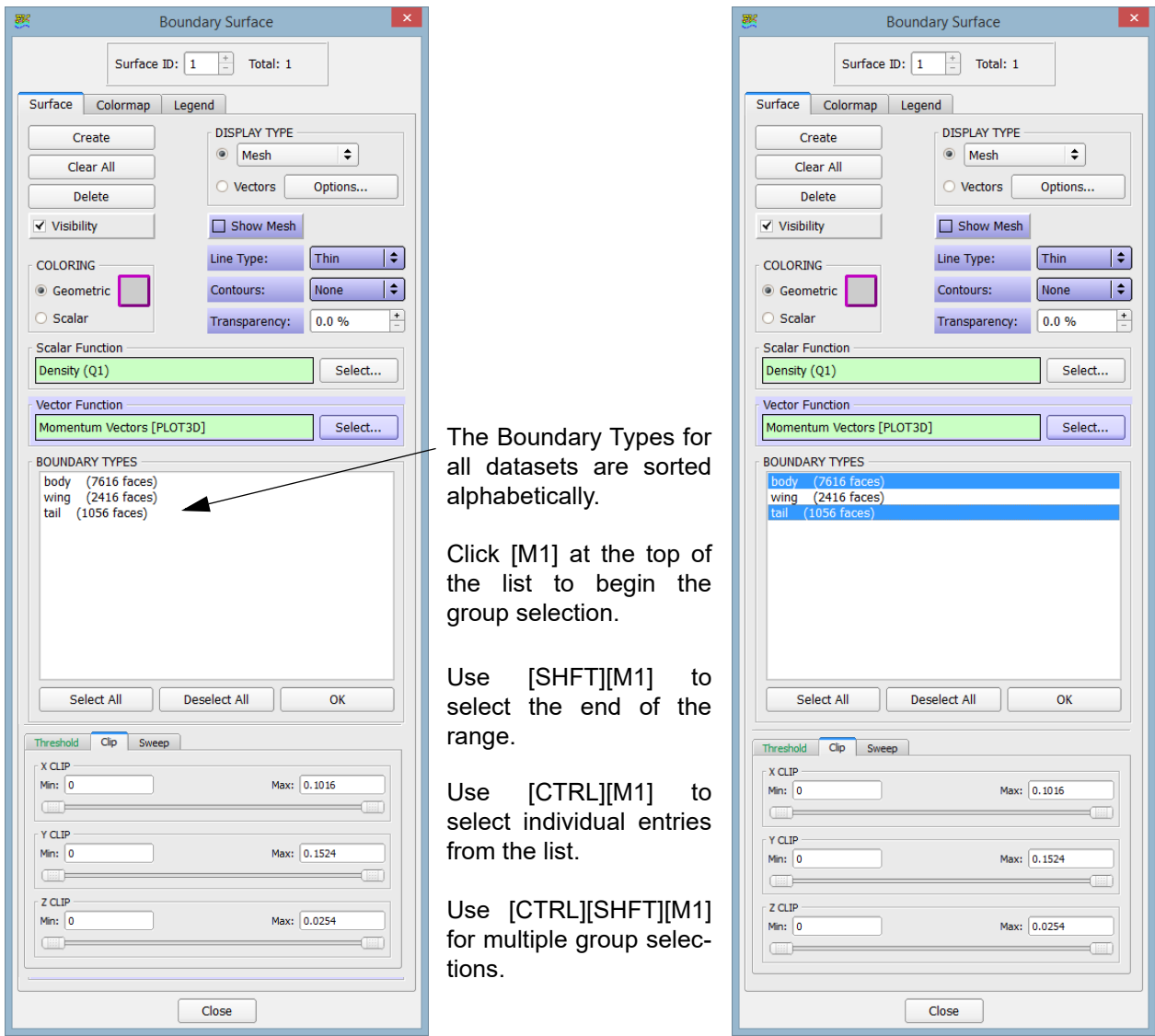

The Boundary Types for all datasets are sorted alphabetically.

Click [M1] at the top of the list to begin the group selection.

Use [SHFT][M1] to select the end of the range.

Use [CTRL][M1] to select individual entries from the list.

Use [CTRL][SHFT][M1] for multiple group selections.

Figure 186 Sorted Boundary Types and Selection

The selection changes apply to the Boundary Types list and the Grid Subset list.

Materials

A Materials tab has been added to the Boundary Surface panel, as illustrated in **Figure 187**.
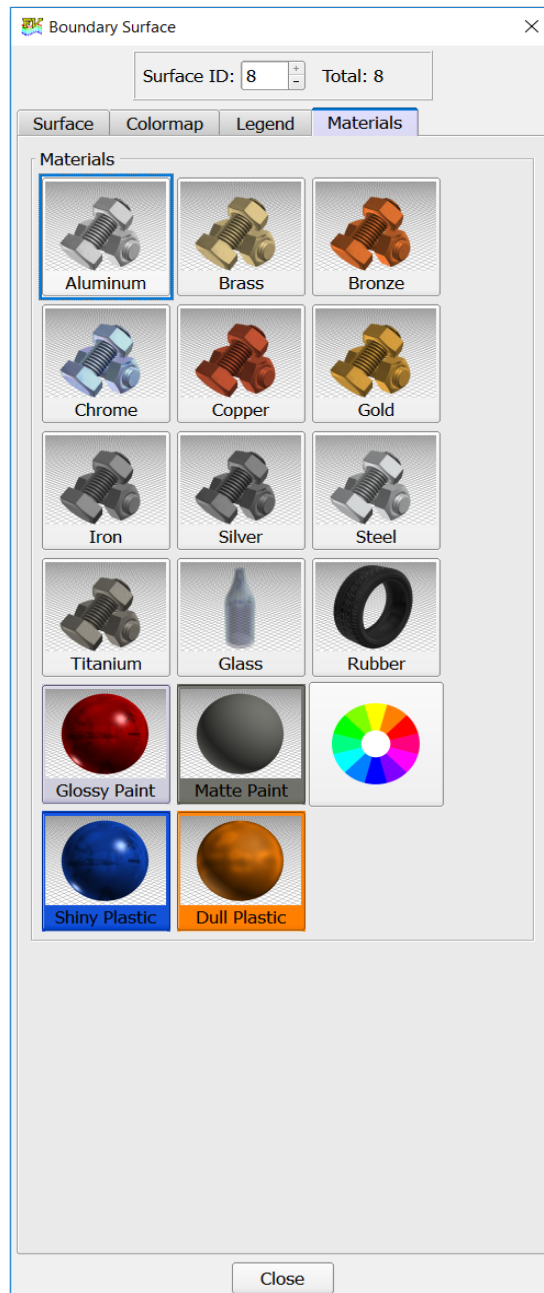
Figure 187 Materials tab on the Boundary Surface panel

Of the 16 predefined materials that are currently available, 6 have editable colors (Aluminum, Glass, Glossy Paint, Matte Paint, Shiny Plastic and Dull Plastic). If a material with editable color is selected (for example, Aluminum is selected in **Figure 187**), a light blue border appears around the material

thumbnail and the wheel (to the right of Matte Paint) appears in color. Clicking the color wheel brings up the Select Color panel, as illustrated in **Figure 188**. If the color is changed, the new color replaces the border color of the material thumbnail. To reset a material's color to its default value, simply select another material and then your material of choice again.
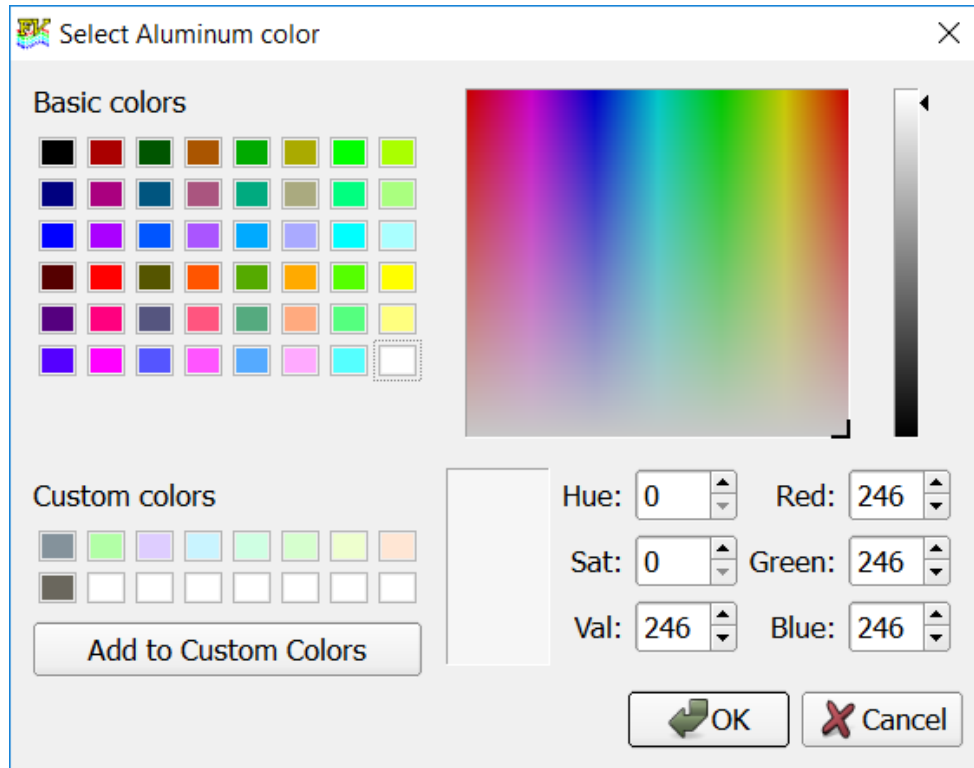
Figure 188 Select Color panel for saving custom colors

All materials except glass are opaque by default. However, their transparency level can be adjusted. Glass has a default transparency of 75%.

Controls for selecting materials and colors are redundantly available from the Surface tab of the Boundary Surface panel, as illustrated in **Figure 189**. A "Material" radio button with a drop-down list and wheel have been added to the COLORING section. This allows users to expedite applying and customizing materials when creating surfaces.   When Material is selected, the drop-down list is enabled (shown expanded in **Figure 190**) and DISPLAY TYPE is forced to Smooth, unless it is set to Faceted.

Note that surfaces with coloring set to Material will not be affected by Presentation Rendering.
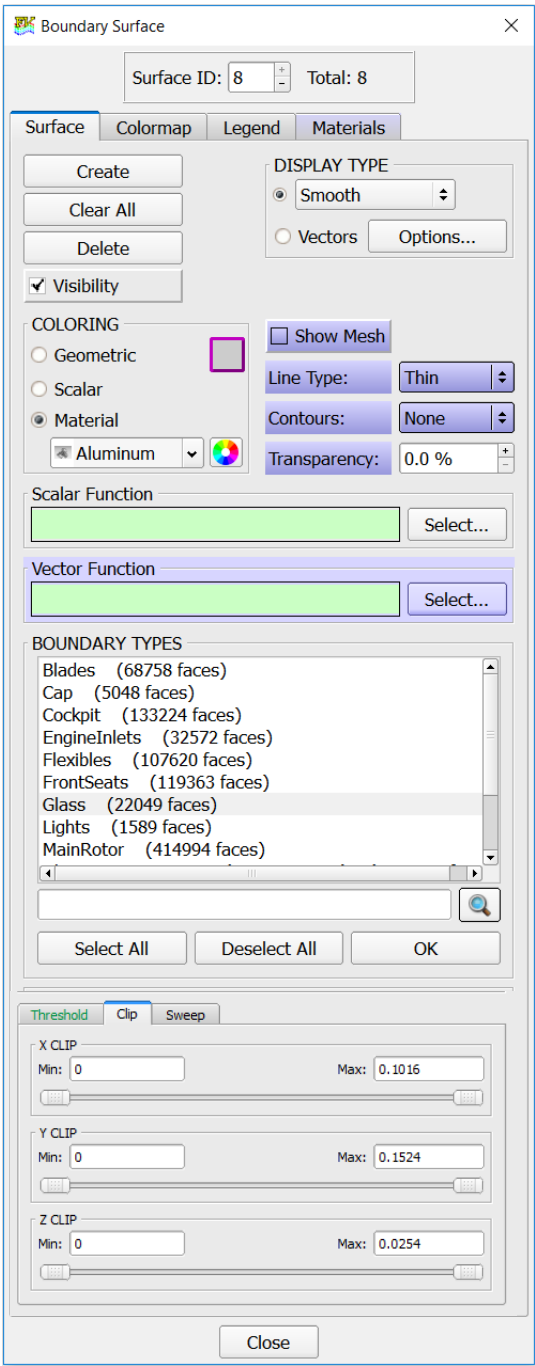
Figure 189 Surface tab with Material controls
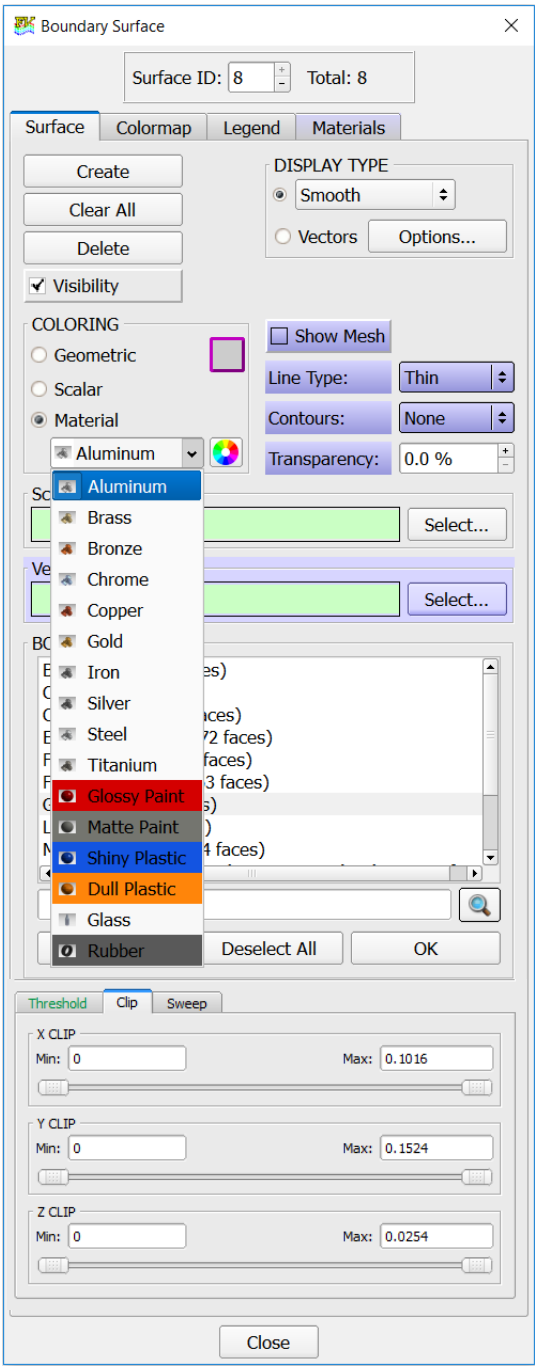
Figure 190 Surface tab with Material drop-down list expanded

Boundary Sweep for XDB Swept Surface

It is possible to sweep through a swept surface in a dataset created with **FieldView** Viewer XDB Build Mode.

To enable the Sweep section on the Boundary Surface panel (see **Figure 6**), do the following:

1. Use XDB Import... to read a dataset containing a swept surface previously exported to XDB during an XDB Build operation.

2. Bring up the Boundary Surface panel and click Create. A list of XDB Surfaces will be displayed. In **Figure 191**, an example with the naming convention used for swept surface export is shown (in this case, "Coord Surf 1").

3. Select one and only one of the swept surfaces, as shown in **Figure 191**.

4. Click OK to enable the Sweep button and controls shown in **Figure 6**.

The Sweep controls move the surface in equally spaced steps from the Current: position to the limits as set by the Min: (minimum) and Max: (maximum) slider positions.

The Min:, Current: and Max: text fields are read-only and correspond to the left, middle and right slider positions. Moving the sliders interactively sets the nearest corresponding XDB swept surface position for the respective field.

The Inc: field is the increment at which the swept surface locations are sampled. The default, 1, displays every surface location (Inc: 2 means show every other position).

To animate the swept surface, turn on the Sweep button. By default, Sweep animates forward and starts again at the minimum after it reaches the maximum. Alternatively, you may change the Sweep direction to animate backward or bounce between the minimum and maximum by pressing the corresponding arrow buttons. Pressing the arrow buttons with Sweep turned off will step through one swept surface at a time.

Flipbook Build Mode on the Tools menu (see **Figure 231**) is supported for this feature. If you turn on Flipbook Build Mode, the Sweep button on the Boundary Surface panel changes to a Build button and you can build a flipbook animation of the swept XDB Surfaces. For more information, see **Flipbook Animation** in the **Reference Manual**.

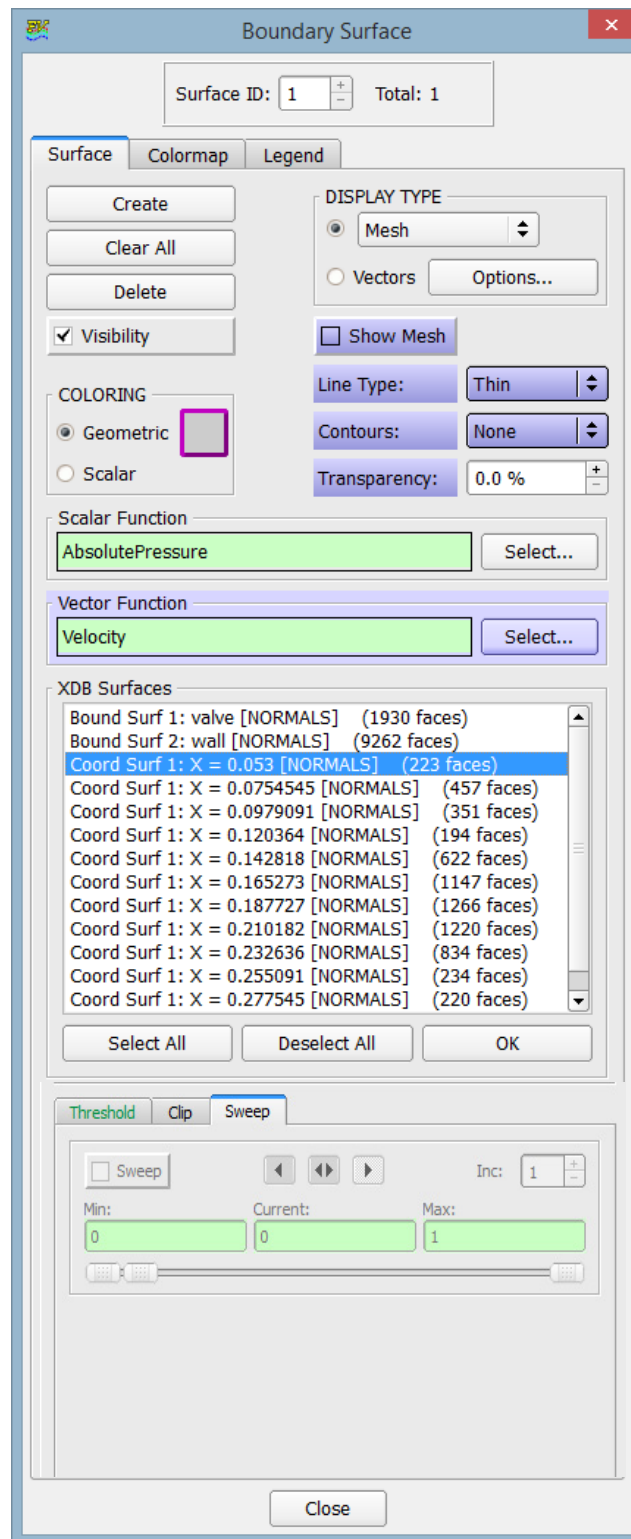This feature is also supported by the `SWEEP` Script command and Sweep Integration.

Figure 191 Boundary Surface Panel after "XDB Import..." and "Create"

Face Data and the Boundary Surface Panel

Face data consists of a single value per face, instead of a value for each node, with a face element made up of 3 or more nodes. Therefore, certain attributes normally available for surfaces are not available for face data. Scalar contours and scalar meshes are not applicable to face data. If a boundary surface has a DISPLAY TYPE of Contours or Mesh and the SURFACE TYPE is Scalar, when the scalar register is changed to a face data function, the DISPLAY TYPE will be forced to Constant Shading and the Contours pull-down will be forced to None. If a boundary surface *already* has face data loaded in the scalar function register, then the Contours and Mesh DISPLAY TYPE options will be grayed out and not available. Similarly, the Contours pull-down menu has all options but None grayed out.

Frequently Asked Questions

Why use the Boundary Surface Panel?
The Boundary Surface panel can display any or all of the boundary types that have been identified in your input file.   This can be used to verify proper boundary conditions by showing the scalar and vector values on the various boundaries. It is also a convenient way to display the geometric shapes that bound the flow.

How do I create a Surface showing the boundaries?
In order to show the boundaries for your model, the input data must identify the boundary types. Currently, unstructured data readers support this option directly. Boundaries of structured datasets are supported with Structured Boundary files (see **Appendix H** of the **Reference Manual** for more information). If the data has the correct information press Create on the Boundary Surface panel to create a surface. Note that this will allow you to create Boundary Surfaces on the *current* dataset *only*. If you wish to create Boundary Surfaces on a different dataset, you will have to change to another dataset. You can do this with the Dataset Controls panel or by "quick-picking" on an object in another dataset. When the Create button is pressed, all of the possible boundary types will appear on this panel. Next, select one or more of the boundary types to be displayed from the Boundary Type selection area. To select a boundary type, press M1 when the cursor is positioned over the name of the desired type. When a boundary type has been selected, the name will be highlighted (shown in inverse video). Clicking on a selected boundary type will deselect (and unhighlight) that choice. Once the boundaries have been selected, press the OK button. Certain actions, such as changing the Display Type, will cause the newly created surface to be visualized and function as implicit OK clicks.

What are the display options for boundary surfaces?
The boundary surface may be colored by a geometry color or a scalar color. The DISPLAY TYPE pop-up menu allows it to be shown as a constant, faceted or smooth shaded surface, a mesh, contour lines, outlines, vertices or shaded vertices. An alternative setting displays Vectors and has a button to control Options. The computational mesh may also be shown on top of the surface by pressing the Show Mesh button.

The *Flip Front/Back* button can be used to "flip" the display of collocated faces.

Boundary Surfaces contain a unique Display Type called Outlines. This option will draw only those edges on the border of the surface (edges shared by only one polygon) or where the surface normal/

crease angle is changing by more than 60 degrees. Thus for smooth surfaces, such as a sphere, no edges will be drawn except at a seam. In this example of a sphere this will result in an arc. In addition, mesh and contour lines may be displayed in conjunction with any surface type including vector.

Note: Outlines will display only as geometry colored with no mesh or contour overlays and is ignored for integration. It is intended as a visual aid only.

How can I see one boundary that is inside of another?
If there are nested boundaries, Transparency can be used to see through the various boundary types. In addition, Thresholding can be used to cut away an outer boundary in order to be able to see an inner boundary. Thresholding in X, Y or Z are often good ways to cut away an outer portion of the model. You may also use the Outlines display type for the outer surface. This will enable you to see the inside surface, while just seeing the outline of the outer Boundary Surface.

Important Note: When integrating Boundary Surfaces, take care not to specify redundant faces in the boundary type selection as these faces will register multiple times in the integration calculation.

Error Conditions

`There are no boundary types defined for any datasets`
If your data reader did not define any boundary types or you have read in a structured dataset without a Structured Boundary file, you will not be permitted to create any boundary surfaces.

Limits

The upper limit for Boundary Surfaces that can be created on a single dataset is 20,000.
The number of Boundary Types within a single file is also 20,000.

Chapter 11

Vortex Cores / Surface Flows

11

This chapter describes the current feature extraction (detection) utilities in **FieldView**. The line based utilities are created by using the Vortex Cores / Surface Flows panel (**Figure 194**) on the Visualization Panels pull-down on the Main Menu. The other remaining utility, Shock Finder, accessed through the Iso-Surface panel, is described later in this section. For general information about the Iso-Surface panel, please see **Chapter 5**.

These utilities are supported by restarts and these line/surface types can be exported. The line based utilities may be re-imported into **FieldView** as Particle Paths. These and other features are described in detail in the separate sections for each utility found below.
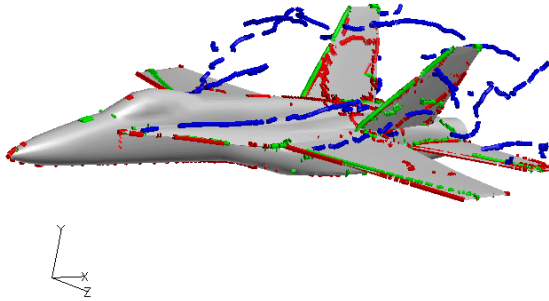
Figure 192 Vortex Cores
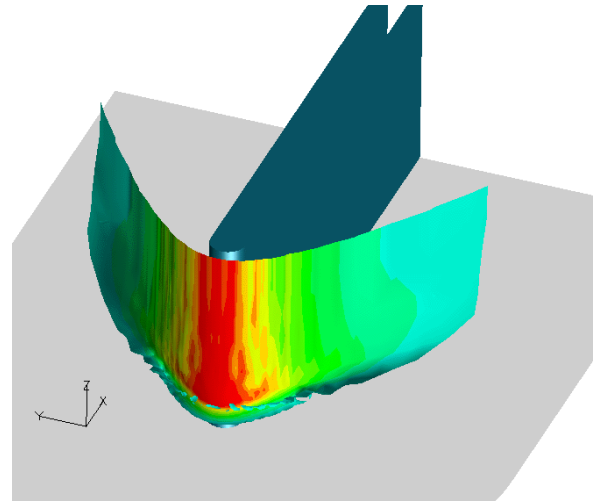
Figure 193 Shock Surface

The Vortex Cores example (**Figure 192 Vortex Cores**) was created using the F18 dataset that is used in the Basic Aerospace Tutorial, **Chapter 3** of the **User's Guide**. The Shock Surface (**Figure 193**) example image was created using the `bluntfin` dataset that is in the `/examples` directory. The shock Iso-surface was scalar colored by Pressure [PLOT3D], thresholded by Pressure [PLOT3D] and clipped in J.

Vortex Cores / Surface Flows Panel

This panel combines several features available with other visualization panels in **FieldView**: ACTION, LINE TYPE, SETUP and COLORING. These common features are fully described in **Chapter 3**, except as noted below.

Utilities

FieldView supports seven different types of vortex cores/surface flows: Vortex Cores (Vorticity Alignment Method), Vortex Cores (Eigenmode analysis), Separation Lines, Re-attachment Lines, Surface Flows (No-Slip), Surface Flows (Euler) and Surface Flows (Offset). These features are choices on the pull-down menu displayed when you press Create on the Vortex Cores/Surface Flows panel.

Except for Separation and Re-attachment lines, all features may be Scalar colored. For the case of Vortex Cores (calculated by either method), you can color by Vortex Strength, or use the contents of the current scalar register. For surface flows, both the duration (see note) and wall shear magnitude are available, in addition to the scalar register.

This panel allows you to Create, Delete or Highlight one or more of the available Feature Extraction Utilities. Pressing the Create button allows creation of: Vortex cores, using either the Vorticity Alignment or Eigenvalue methods, Separation lines, Re-attachment lines, or Surfaces Flows for either No-Slip or Euler boundaries.

Vector Function shows the currently loaded Vector Register which will be used for these calculations. An empty Vector Register will result in an error pop-up.
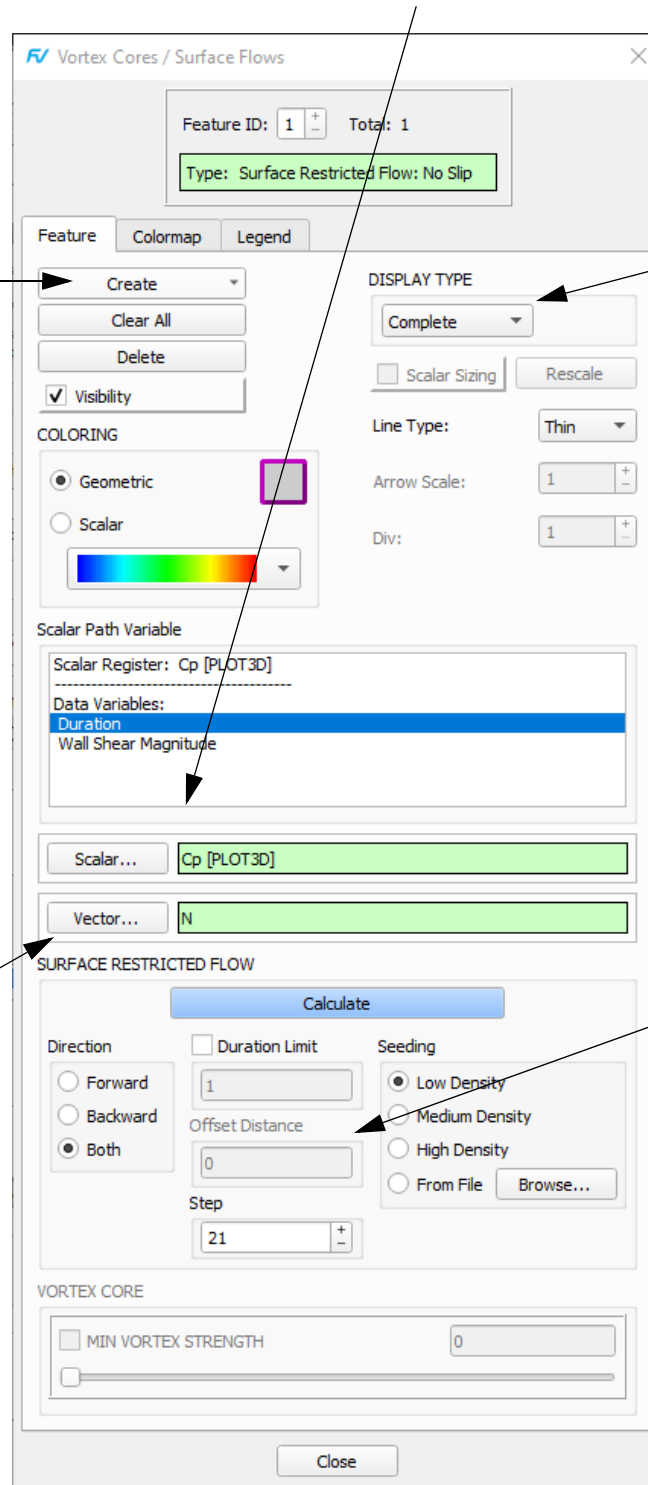
The default Display Type "Complete" corresponds to the single implicit display type used for all vortex cores, separation and attachment lines, and surface flows prior to **FieldView** 18.1. This display type lacks ability to indicate direction.

The Display Type "Arrows & Lines", described in the section **Surface Flow Panel Options**, is also available.

You MUST specify a non-zero Offset Distance in order to calculate Offset Surface Restricted Flows.

If the Offset Distance is set to 0 (zero), surface flows will be calculated using the Surface Restricted Flow: Euler method.

Figure 194 Vortex Cores/Surface Flows Panel
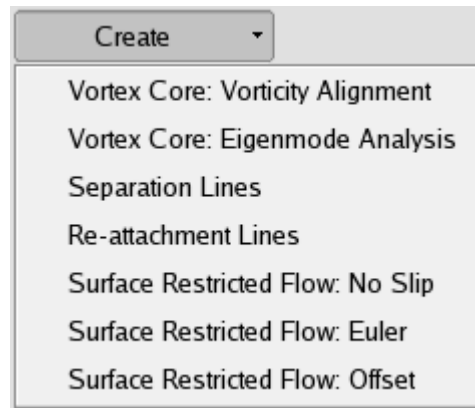
Figure 195 Create Feature Pull-down menu

Vortex Cores

There are two methods which can be used to calculate vortex cores from the above menu selections: Vortex Core: Vorticity Alignment and Vortex Core: Eigenmode Analysis. The Eigenmode Analysis method will sometimes yield a 'cleaner' picture. Vortex Cores are computed for the entire volume of the current dataset on a cell-by-cell basis. Vortex cores represent axes of swirl in regions of high vorticity. They are considered to be **FieldView** surfaces that can be seeded on, exported, restarted and they have script support.

This feature requires that the Vector Register be non-empty. If the register is empty and the Create button is pressed, then an error pop-up will inform you of this problem.

In the Vorticity Alignment algorithm, the Vortex Cores are located by finding cells in which the Vorticity is aligned with Velocity. Cells in which two or more faces of the cell have met this criterion are determined to have been pierced by the core center-line. These core center-lines are then collected and drawn to display the core segments. Please refer to the references listed below for a complete description of these algorithms.

In the Eigenmode Analysis, the real components of the eigen-analysis of the velocity gradient tensor are used to determine the axes of swirl. The Vortex Core is defined as cells where two or more faces of the cell have zero reduced velocities (where the swirl direction has been subtracted from the nodal velocity). If this is the case, then the core center-line is determined to have pierced the cell. These core center-lines are collected and drawn to display the core segments. Please refer to the references which follow for a complete description of these algorithms.

References

R. Haimes and D. Kenwright. On the Velocity Gradient Tensor and Fluid Feature Extraction. AIAA Paper 99-3288, 1999.

D. Sujuki and R. Haimes. Identification of Swirling Flow in 3-D Vector Fields. AIAA Paper 95-1715, 1995.

Vortex Cores can be colored by any Geometric color available (including those created using the Color Mixer). In addition, they can be scalar colored. The scalar used to color the vortex cores is either the function currently loaded in the Scalar Register (this can be changed by loading in a different function, whether intrinsic, user-defined or a results variable), or the special data variable "Vortex Strength" (defined as the magnitude of the Vorticity).

Vortex Cores can be clipped in a way similar to Particle Paths (see **Chapter 7**). The clipping function is Vortex Strength. In order to clip the Vortex Cores, turn *on* the MIN VORTEX STRENGTH toggle button and adjust the value as desired. Since the Vortex Cores are actually line segments, those portions where the value of vortex strength is less than the minimum value set will be clipped.

Note: Vortex cores cannot be moved independently using Surface Transforms. An error pop-up will tell you that you are trying to transform a surface that cannot be transformed.

Surface Flows

There are a total of five methods to display surface flow patterns: Separation lines, Re-attachment lines, and three types of surface restricted streamlines, namely Surface Flow (no-slip), Surface Flow (Euler) and Surface Flow (Offset).

Note: These lines will only be calculated for the visible sections of boundary surfaces. If boundary surfaces are displayed using Outlines, then no surface flow patterns will be shown. No automatic update will occur if the Boundary Surfaces later change.Please note that extracts (XDB Export) will not contain Vortex Core / Surface Flow objects.

Note: Boundary surfaces consist of triangular and quadrilateral faces. The quadrilateral faces are not necessarily planar but may be split into a pair of triangles for the calculations mentioned below.

Separation and Reattachment lines
Separation and Reattachment lines are curves that exist on Boundary Surfaces in 3-D vector fields, specifically where streamlines tend to converge or diverge. Separation and reattachment phenomena generally occur when flow is turned away from (or returns to) a surface such as a Boundary Surface. The algorithm employed here is based on a phase plane analysis (see references below) and is performed by local analysis of the vector field. Essentially, a test is performed to locate where the stream function is zero or singular. Lines segments that cross a given face on the boundary surface are then collected to form the lines for rendering. Each line segment is either a Separation Line (segment) or a Reattachment Line (segment) depending on the way the phase plane intersects the surface, which has been divided up into triangles. Because any number of triangles that make up the surface may, for example, contain Separation Line segments, the Separation Lines shown in **FieldView** are merely a collection of all such line segments, given a unique color (red for Separation lines, by default). The same is true for Reattachment Lines which are shown in green by default. This is why exporting Sur-

faces Flows (to the **FieldView** ASCII Particle Path file - see below) can result in so many individual paths. Please refer to the references which follow for a complete description of these algorithms.

References

D. N. Kenwright, C. Hense and C. Levit. Feature Extraction of Separation and Attachment Lines. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 2, April-June 1999.

D. N. Kenwright. Automatic Detection of Open and Closed Separation and Attachment Lines. *Proc. IEEE Visualization '98*, pp. 151-158, 1998.

Separation and Re-attachment lines can be colored by any available geometric color (including those created using the color mixer). They cannot be scalar colored. Correct normal definition is essential to the classification of surface flow features as either separation or re-attachment phenomena. Any lines **FieldView** cannot properly classify will be colored magenta. Furthermore, surface flow on a zero-thickness internal boundary (such as a baffle) is calculated for both sides at once and are not distinguishable from each other.

Surface Restricted Flows
Surface flows are special streamlines which are restricted to lie in the plane of the boundary surface. Special handling and use of surface normals has been applied to ensure that the trajectories remain within the plane of the boundary surfaces upon which they are drawn. These surface flows can be calculated for either structured or unstructured data.

Two of the choices available to calculate surface flows are No-Slip and Euler. If the surface is No-Slip, **FieldView** considers it to have a zero velocity everywhere on this surface, as applied by a no-slip boundary condition. Surface streamlines are then calculated using the velocity gradient, or shear components which are tangential to that surface. Since the velocity gradient is being used instead of a velocity, the units of duration will not correspond to time. For the case of Euler flows, **FieldView** considers the surface to have a slip boundary condition applied. In this case, the tangential components of the Vector Register are used to calculate the streamline trajectories. In most cases, the Vector Register should contain the physical property, velocity. **FieldView** also allows the user the option of employing their own vector, say for instance, shear velocity, which would include the effects of viscosity in the calculation.

Offset Surface Flows
This method of calculating surface flows resolves a limitation of the no-slip based surface flow calculation when applied to boundary surfaces with non-zero velocities. By specifying an offset distance from a boundary, 'near-surface' flow patterns are displayed for datasets with rotating fans or turbine blades, blowers and wheels.

Surface Restricted Flow: No Slip
No slip surfaces flows computed
using shear velocity (velocity
gradient at surface)
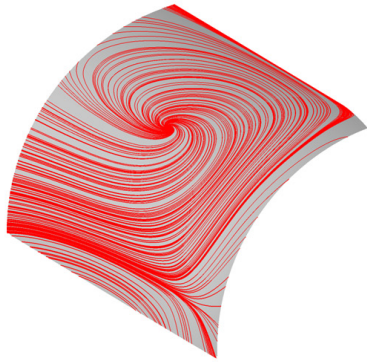
Surface Restricted Flow: Offset Surface
User selects offset distance (real units)
Velocity components are sampled at offset
distance, projected back onto surface, and
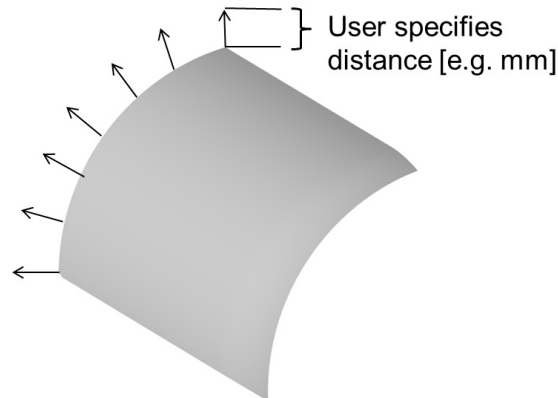Euler surface flow calculation is performed

User specifies
distance [e.g. mm]

Figure 196 Methodology for Surface Restricted Flows: Offset Surface

To use this feature, you must specify an offset distance in the same units as the dataset. **FieldView**
will then sample the velocity field at the offset distance from all of the boundary surfaces, and use
these values as the input to the Euler method for calculating surface flows. Since the offset from the
surface needs to be sampled in a consistent direction, it is a requirement that boundary surfaces must
have consistent clockness. If this condition is not met, **FieldView** cannot determine a consistent dis-
tance or direction to do the sampling, and consequently, no surfaces flows will get calculated.
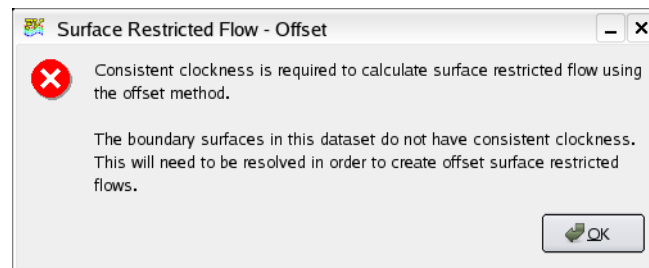
Figure 197 Surface Restricted Flow - Offset Warning Message

If the boundaries in your dataset have inconsistent clockness, you will see a warning message pop-up
(see **Figure 197 Surface Restricted Flow - Offset Warning Message**). In this case, it may still be
possible to obtain offset surface flows by turning off the visibility of either the clockwise or the counter-
clockwise boundary surfaces to ensure consistent clockness.
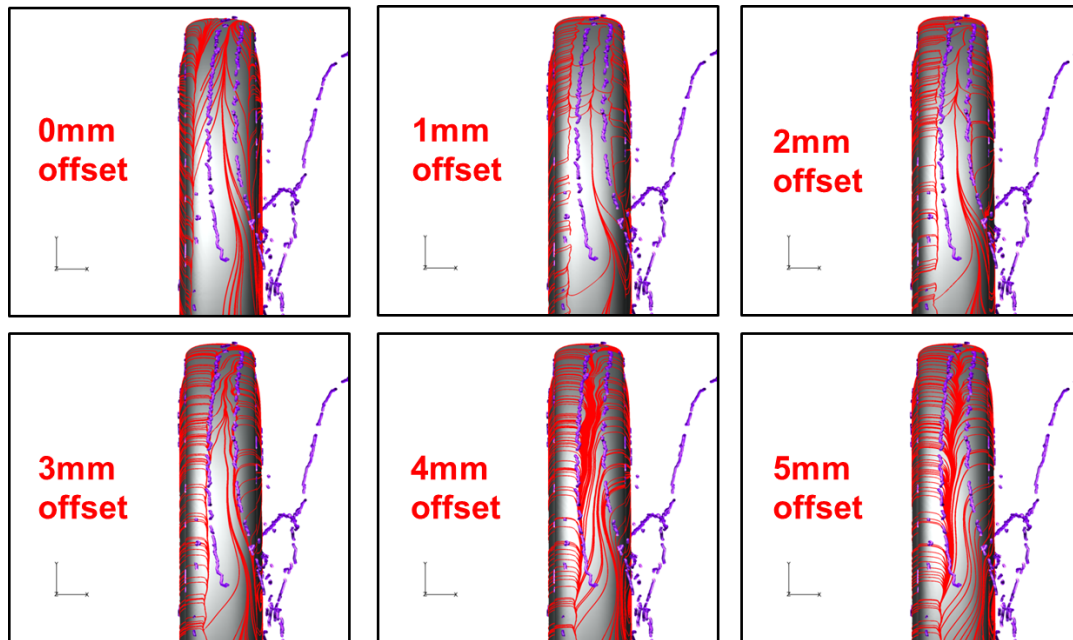
Figure 198 Offset surface flows on the trailing edge of a wheel

In the illustration above, offset flows are calculated at successively increasing distances from the trailing edge of a wheel with a constant rotational velocity applied. The offset surface flows are shown in red, and the vortex cores in the surrounding volume are shown in purple. There are significant differences in the offset surface flow patterns depending upon the offset distance chosen. The choice of offset distance is dataset dependent, and it is recommended that several values are inspected before an offset surface flow visualization is finalized.

The Vortex Cores/Surface Flows panel is illustrated in **Figure 194**. Choose the Surface Restricted Flow: Offset option from the Create... pulldown menu to create surface restricted flows of this type. To perform the Surface Restricted Flow calculation, you will need to press the Calculate button. By default, the Offset Distance is initially set to zero. You will need to supply a reasonable value for this offset distance BEFORE pressing the Calculate button. If you use an offset distance of 0 (zero), the Surface Restricted Flow: Euler method will be used to perform the calculation and no sampling of the velocity field away from the surface will be done.

Surface Flow Panel Options
To calculate surface flows, the user must know which Surface flow option to use (see **Figure 203**). Once this choice is made, the user has the chance to define many streamline properties, such as direction, time limit, and seed density, prior to the calculation. Once the correct option choices have been made, pressing the calculate button will initiate the calculation task. Once this calculation process has been started, it cannot be halted. After the calculation is complete, messages relating to the streamline calculation can be displayed by clicking on the Messages… button.

Surface Restricted Flows can be colored by any Geometric color available (including those created using the Color Mixer). In addition, they can be scalar colored. The scalar used to color the vortex cores is either the function currently loaded in the Scalar Register (this can be changed by loading in a different function, whether intrinsic, user-defined or a results variable), or the special data variable "Duration" or "Wall Shear Magnitude".

The Display Type "Arrows & Lines" is available to show the flow direction, as illustrated in **Figure 199**.

Figure 199 The Arrows & Lines display type for Surface Flows provides insight on the direction of the flow near the wall. Shown here is the rear of a notional car (the Ahmed body).

"Arrows & Lines" is similar to "Filament & Arrows" in the Streamlines panel, except that there are no breaks in the flow lines. "Arrows & Lines" is only available for Surface Restricted Flow types Euler, Offset, and certain cases of No-Slip.

When a No-Slip flow is calculated, if arrows are inappropriate and not allowed, you will see the following 3-line console message explaining why "Arrows & Lines" will be disabled:

```
The boundaries do not have consistent normals.
```
or
```
Some boundaries do not have consistent normals.
Cannot reliably determine the direction of forward flow.
Arrows&Lines display type will be disabled.
```

When "Arrows & Lines" is selected for a surface flow, several additional controls (Scalar Sizing, Arrow Scale and Div) will be enabled that are not available with Display Type "Complete". These controls affect various aspects of the arrow heads. A Display Type may be selected for each Surface Restricted Flow created, whereas arrow attributes are global across all surface flows.

If "Scalar Sizing" is turned on, the arrow heads will be scaled based on the current scalar variable.

"Arrow Scale" controls the arrow head size explicitly (similar to "Sphere / Arrow Scale" on the Streamlines panel). The "Rescale" button will reset the arrow heads' base scaling according to the current view.

"Div" controls arrow head spacing along the lines. "Div" for Surface Flows "Arrows & Lines" works differently than for streamlines. A Div value of 1, the default, will draw an arrow at each surface flow seed location. A Div value of 2 will add to that an arrow along the flow line at 1/2 the duration from the seed to the flow line end in the forward, backward, or in both directions. A Div value of 3 will add arrows at 1/3 and 2/3 the duration, and so on.

When 'From File' is first selected, a file browser will come up automatically. To change from a previously read file, click the Browse... button to bring up the browser explicitly.

The seeds location file can be generated in the following ways:

- Outside of **FieldView**, as a simple text file where each line represents a point location, with 3 columns for XYZ coordinates.
- From **FieldView**, by generating either:
    - A point probe output (see **Working with FieldView**, **Chapter 13**)
    - A 2D Plot export file (see **Working with FieldView**, **Chapter 12**)
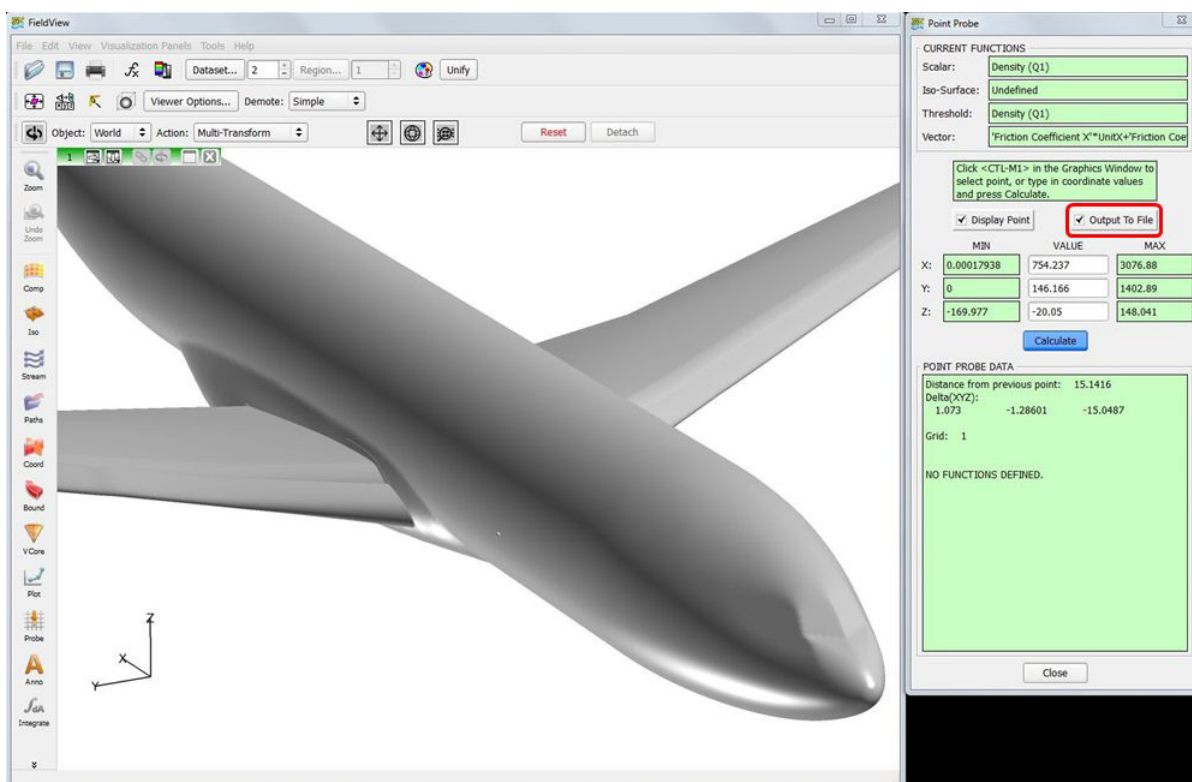
Figure 200 Point Probe output to file
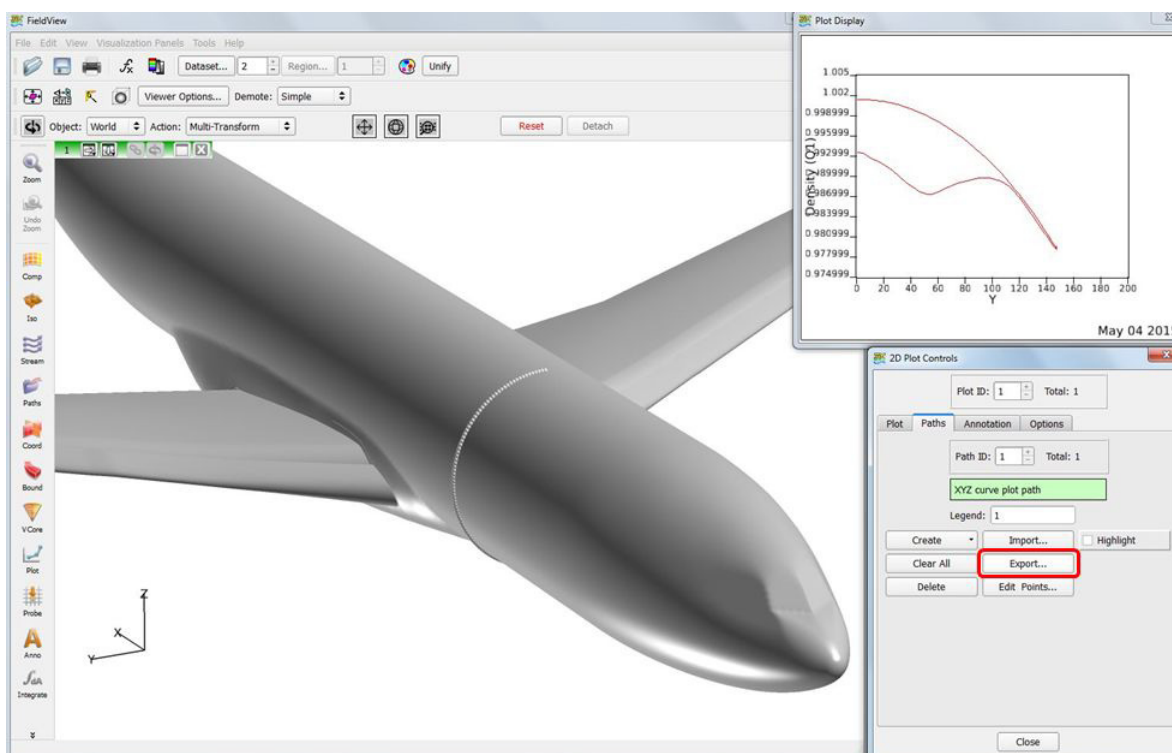
Figure 201 2D Plot Export to file
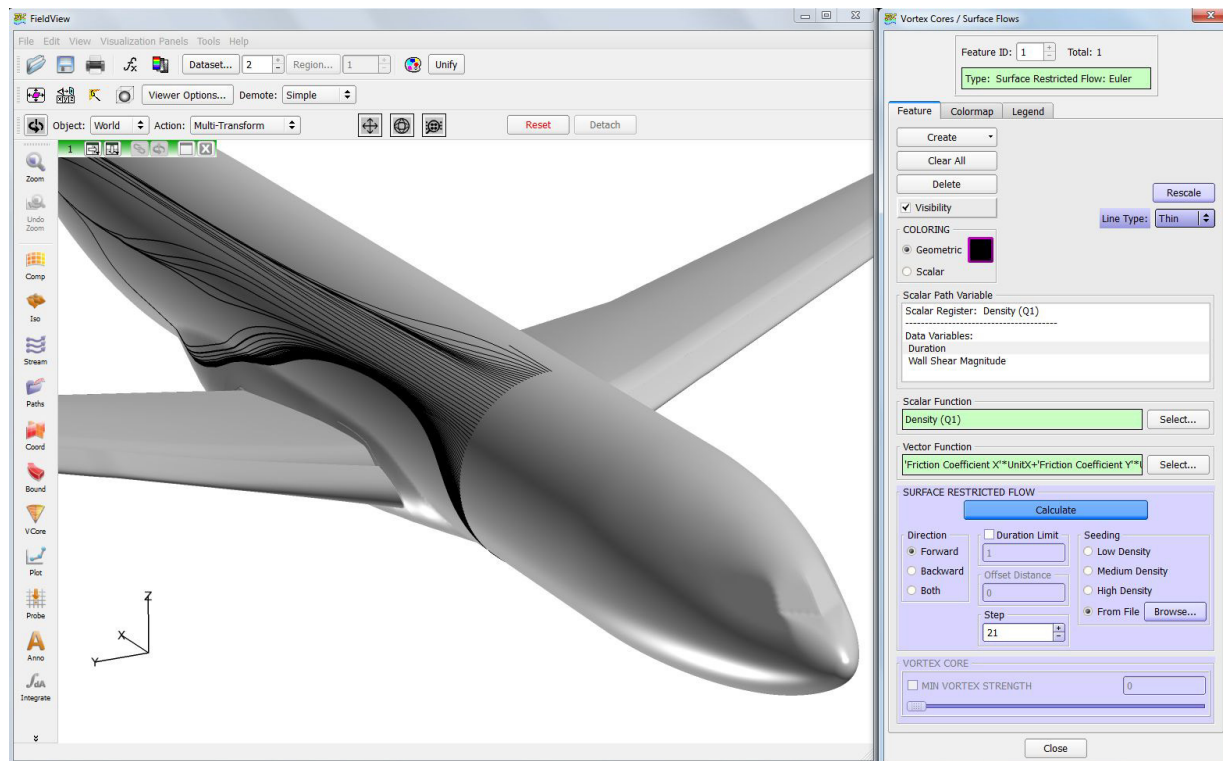
Figure 202 Surface Flows created from a 2D Plot output file

After the seeds location file has been selected with the browser, click Calculate on the Vortex Cores/ Surface Flows panel to read it in and generate and display the flow lines.

Each surface flow object may have a unique seeds location file. Points that do not lie on a visible Boundary Surface will be ignored. File paths will be saved with restarts.

The maximum number of points for user-defined seeding locations for surface flows is the same as the limit for streamline seeds. See **Reference Manual Appendix F**.

Parameters relating to Surface Flows should be selected before pressing the Calculate button. The options include: Direction {default = Both}, Step {default = 3} and Seed Density {default = Low}.

Duration Limit, if turned on, will limit the drawn length of surface flows. If the Direction is Both, 1/2 (half) the specified Duration Limit will be Forward and 1/2 will be Backward (for example, a Duration Limit of 10 will be 5 Forward and 5 Backward). The minimum value must be greater than 0.0; there is no maximum. The units of duration are in time only if Euler is chosen, and the vector register contains velocity. For No-Slip, the (pseudo) duration units will be in units of shear (or inverse time).

Vortex Cores (either method) can be clipped by the minimum strength. Either the slider or the type-in field can be used. The toggle button can be turned on or off.

Seeding has three Density options, Low, Medium and High which correspond to 200, 500 or 1000 randomly placed seeds on a surface. Seed locations may alternatively be read from a file. Seeds will not be shown. Seeding will respect thresholding of the visible boundaries, and this may be used to concentrate seeds in a certain area of interest.
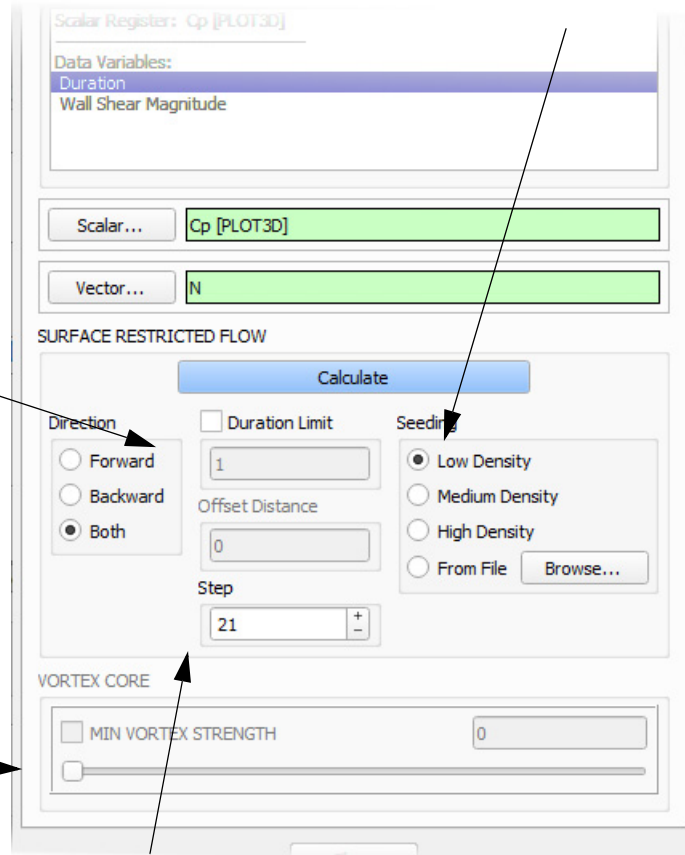
The Integration Step is used to specify the precision of integration within a cell. Setting a value to 10 means that each integration step will be at most one tenth of the cell. The trade-off is between accuracy and time required to calculate the surface flows. For reasonably accurate illustrations, you should increase the step from 3 to at least 9. default from 3 to at least 9.

Figure 203 Surface Flows Panel Options

Note: Clipped portions of Vortex Cores using the MIN VORTEX STRENGTH toggle button will not be exported.

Export

Vortex Cores and Surface Flow lines can be exported (either using Export As... on the Tools menu or the EXPORT script command) by the ASCII Particle Path format. The exported file can be read back in using the Import feature on the Particle Path panel, or sent to other software you desire. These features and the format are described in **Chapter 7** and **Appendix L** of the **Reference Manual**. The registers (function) values that are exported will include the Vortex Strength for vortex cores or Duration and Wall Shear magnitude for Surface Restricted Flow, as well as any non-empty register (Iso, Scalar, Vector and Threshold). The Vector register will be exported and upon import through the Particle Path panel, will allow you to select any of the three components of the vector function as a scalar.

Note: Here, export of your visualization object is being discussed. Please note that extracts (XDB Export) will not contain Vortex Core / Surface Flow objects

Presentation Rendering

Vortex Cores and Separation and Re-attachment lines can be Presentation Rendered. They have a separate field on the Presentation Rendering Options panel (see **Chapter 2** for more information). Since they exist as line segments, when Presentation Rendered, they will display as stream 'tubes', similar to streamline display type Filament.

Note: Surface Restricted flow lines, calculated using either method (No-slip or Euler) will not be Presentation Rendered even if it has been selected for Vortex Cores/Surface Flows on the Presentation Rendering Options panel.

Restart and Script Support

Note: Surface Flow lines are dependent on the Current Boundary Surface(s) on which they are based. An error message will be printed to the terminal if a restart for this panel is read without the corresponding Boundary Surface(s).

For further information on restart files, see **Vortex Cores / Surface Flows** in **Chapter 5** of the **Reference Manual**.

Vortex Cores / Surface Flows have the same basic script support as other **FieldView** surface types. This includes a RESTART command for a surface restart, a SELECT command to make them *current*, a SHINE command for Presentation Rendering (see **Chapter 2** and **Chapter 6** for more information) and an EXPORT (see also **Appendix I** of the **Reference Manual**) command for exporting. These script commands are covered in **Chapter 5** of the **Reference Manual**.

Regions

All Vortex Cores and Surface Flow lines belong to the dataset, not to any specific region. They cannot be confined to a particular region, nor are they transformed with any region transforms. They are also unlike other surfaces in **FieldView** in that they cannot be transformed with Surface transforms.

Shock Surfaces

The Shock Surfaces feature uses several algorithms to determine the location of shocks in the flow field. This feature is also supported by environment variables that allow you to control various aspects of the filtering done on the surface (for more about filtering shocks see **"Shock Filter Functions" on page 268**). Although this surface is actually a type of Iso-Surface, it is primarily covered in this chapter. The Shock Finding feature has certain functional requirements for use.

Shock Surfaces are determined by using the pressure gradient to find the value of the Mach number normal to a shock. The shock surface normal is parallel to the pressure gradient vector, the normal Mach number being the component of the Mach number in this direction. This information is then used to locate shocks by defining them to be the region where the normal Mach number is greater than or equal to one (1). This technique can fail to capture moving shocks associated with transient solutions. For transient solutions, the Mach number is calculated with respect to the moving shock. This produces a correction term to the steady algorithm to account for this motion. In addition, the resultant shock surfaces are filtered. For more information about this filtering process, see the Environment Variables section at the end of this chapter. Please refer to the reference which follows for a complete description of these algorithms.

Reference
D. Lovely and R. Haimes. Shock Detection from Computational Fluid Dynamics Results.
AIAA Paper 99-3285, 1999.

You can create a Shock Surface for the current dataset by using the Iso-Surface panel. This is done by loading the Iso Function register with one of the 'special' Shock Finder functions. These functions only exist if certain variables (and variable names) exist for the current dataset. The Shock Finder functions are indicated in the Function Selection panel with a specific naming convention. All of these functions begin with the words Shock Finder followed by a description label for the specific type available. The type available depends on what solver results variables exist and also whether the dataset is steady state or transient. Because of this, not all datasets will have these functions available.

The Shock Finder function is a true Iso-function. However, the only viable value for this function is 1. Therefore, the Iso-Surface panel will show the ISO FUNCTION slider as having a Min, Value and Max all equal to 1. Part of an Iso-Surface panel which has a Shock Finder function loaded in the Iso-register is shown in **Figure 204**. None of these three fields will accept any value other than 1.
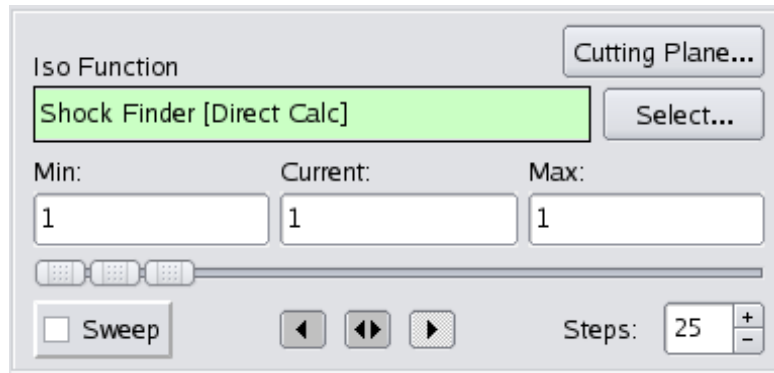
Figure 204 Shock Surface Iso-Surface Panel Slider

Note: For structured datasets, the value of Inc for I, J, and K on the Iso-Surface Subset Parameters panel will be forced to 1. All subsequent iso-surfaces created, whether using Shock Finder functions or not, will inherit this Increment value, unless you explicitly change it. If the Increment has been changed to a value other than 1, and a Shock Surface is created, a warning will be issued to this effect and the surface will not be properly "filtered" (it will be overly broad). In this case, you should change the Increment value back to 1.

There are three types of steady state Shock Finder functions. These are described in the following table:

Type	Description
Shock Finder [Q File]	This function will only appear if a PLOT3D Q file is present for the current dataset. It uses the PLOT3D derived functions for velocity, pressure, Mach number, density and total enthalpy, using the normal PLOT3D perfect gas and constant gamma assumptions.
Shock Finder [Perfect Gas]	This function will only appear if the dataset has five (5) scalar quantities called Q1, Q2, Q3, Q4 and Q5. These should match the meanings of the 5 Q variables in the PLOT3D Q file (density, x-momentum, y-momentum, z-momentum and stagnation energy per unit volume, respectively). These quantities must either be in consistent units and be consistent with the length units for the grid, or they can be normalized by the free stream conditions in the same way as the PLOT3D Q file. The quantities Q1,…, Q5 can be the names of variables or they can be names of user-defined formulas (i.e. constructed using the "CFD-Calculator"). The function names can be in upper or lower case. This technique uses the same shock calculations as the [Q File] shock finders and has the same assumptions.

| Shock Finder [Direct Calc] | This function will only appear if the dataset has two scalar functions: `pressure [shock]` and `mach [shock]` and one vector function called `velocity [shock]`. These can be the names of variables or user-defined formulas in upper or lower case. This technique makes no assumptions about perfect gas or constant gamma.
Note: Merely having a function called `pressure` does not fulfill the function requirements, for example. You will need to create a new function with the "CFD-Calculator" equal to `pressure` and give it the name `pressure [shock]`. |

There are six types of transient Shock Finder functions. These functions are divided into two groups. The first group requires exactly the same functional requirements as the steady state case:

| Shock Finder [Q File] [Isentropic Transient] | Shock Finder [Q File] [Transient] |
| Shock Finder [Perfect Gas] [Isentropic Transient] | Shock Finder [Perfect Gas] [Transient] |

The other two types require the same functional requirements as the steady state case plus additional functions. These are covered in the following table:

Type	Description
Shock Finder [Direct Calc] [Isentropic Transient]	In addition to the functional requirements of Shock Finder [Perfect Gas], the [Isentropic Transient] version also requires the scalar function: density [shock].
Shock Finder [Direct Calc] [Transient]	In addition to the functional requirements of Shock Finder [Perfect Gas], the [Transient] version also requires the scalar functions: density [shock] and total enthalpy [shock].

The [Isentropic Transient] functions assume Isentropic conditions. The [Transient] functions do not make these assumptions, but are slower and assume good grid refinement for various derivable calculations.

Environment Variables
The shock finding algorithm accepts two environment variables. The default values of these two variables have been set to values that find the shock surfaces in the most consistent manner. You may adjust these parameters if shocks are not found (decrease FV_SHOCK_PRESSURE_CHANGE) or if there is too much clutter in the result (increase FV_SHOCK_PRESSURE_CHANGE), for example. If these envi-

ronment variables are set when starting **FieldView**, this information will be echoed to the standard output (your `xterm` window).

First, the Shock Finding algorithm calculates shock regions (where the normal Mach number is greater or equal to one (1)). The `FV_SHOCK_MIN_NODES` environment variable removes all shock regions with less than the specified number of cells (or partial cells).

The `FV_SHOCK_PRESSURE_CHANGE` environment variable removes all shock regions that fall below a certain threshold of the pressure gradient. The pressure gradient, for these purposes, is defined as the pressure change (max(pressure) - min(pressure)) in a connected shock region and uses the global dataset pressure as a normalization factor.

Variable	Definition
`FV_SHOCK_MIN_NODES`	The minimum number of nodes in a connected shock region. The default is 4.
`FV_SHOCK_PRESSURE_CHANGE`	The minimum pressure change: max(pressure) - min(pressure) in a connected shock region. It is expressed as a fraction of the total pressure range for the entire dataset, i.e., (shock pressure range) / (dataset pressure range). It therefore has a range of 0 to 1. The default is 0.01.

For information about how to use these and other environment variables, see **Chapter 1** of the **User's Guide**.

Shock Filter Functions

A new shock filter function has been added to complement the automated calculation of the Shock Finder function. Once the pressure, mach no. and velocity (and additionally density and total enthalpy for transient cases) are established, the Shock Finder Function is available as a variable for iso-surfaces. When the Shock Finder Function has been calculated, the desired Shock (Finder) Filter Function is then determined from:

```
Shock Finder Filter = grad("Shock Finder []") dot grad("pressure []")
```

A Shock Finder Function will be automatically calculated depending upon the type of dataset which is read in and whether the dataset is steady or transient. If a shock function can be calculated (by satisfying the necessary criteria), then the Shock Finder Filter will be automatically available. The Shock Filter function is used to threshold an iso-shock surface showing only values less than zero in order to filter out extraneous surfaces created by "expansion shocks" showing only compressions shocks. As a result, the Min slider button should be set to a number just smaller than zero. To calculate the above

Shock Finder Filter expression, we need to specify how pressure will be calculated. There are a total of nine cases to consider:

Shock Finder [Q-file]	use pressure [PLOT3D]
Shock Finder [Perfect Gas]	use pressure = f(Q1,..,Q5)
Shock Finder [Direct Calc]	use pressure [shock]
Shock Finder [Q file][Isentropic Transient]	use pressure [PLOT3D]
Shock Finder [Q file][Transient]	use pressure [PLOT3D]
Shock Finder [Perfect Gas][Isentropic Transient]	use pressure = f(Q1,..,Q5)
Shock Finder [Perfect Gas][Transient]	use pressure = f(Q1,..,Q5)
Shock Finder [Direct Calc][Isentropic Transient]	use pressure [shock]
Shock Finder [Direct Calc][Transient]	use pressure [shock]

The Shock Filter function is listed as an available scalar with the name Shock Filter [ ]. The contents of the square brackets will indicate the basis for its calculation.

Note: The Shock Finder [ ] function is only available for iso-surfaces and does not show up in the general list of scalars. The filter, Shock Filter [ ], will show up however in the general list of scalars, even before an iso-surface of the shock has been created.

Reference
Shih-Ping Liou, Stefan Mehlig, Ajit Singh, David Edwards and Roger Davis. *An Image Analysis Based Approach to Shock Identification in CFD*. AIAA 95-0117, January 9-12, 1995.

Chapter 12

2D Plot Controls

12

The 2D Plot Controls panel (**"2D Plot Controls and 2D Plot Paths Tab" on page 271**) is used to make two dimensional plots (graphs) of the current scalar function on a straight line in X, Y, Z space, or along a curve on a particular surface for the current dataset.

Support for multiple 2D plots per dataset is provided; the maximum number of 2D plots is 9. The maximum number of plot paths per 2D Plot is 10. Each 2D Plot can display a different scalar function.

Because of this change, the 2D Plot Controls panel in this release is more consistent with the panels for other surfaces and rakes. To create a new 2D plot, first press the Create... button on the 2D Plot Controls panel (see). Next, create a 2D plot path within a 2D plot, by pressing the Create... button on the 2D Plot Paths panel.

2D Plot Controls and 2D Plot Paths

Press Create on the Plot tab of the 2D Plot Controls panel to create a 2D Plot.

Switch to the Paths tab to be able to create a path on a 2D plot.

Show Path will display the permanent markers in the graphics window showing the points used to create the plot.

Show Plot will display or embed the 2D Plot in the graphics window.

When Right Axis is pressed, the variable loaded into the Thresholding register will be used as the right-hand axis of the 2D plot.

Use PLOT SELECTION on the 2D Plot Controls panel to navigate between 2D Plots. The maximum number of plots is 9.

Parameters associated with Annotation, Left, Right and Horizontal Axes, Sampling and Border control the appearance of each specific 2D Plot, and are applied on a per 2D Plot basis.

Press Create on the Paths tab to create a path within a 2D Plot.

Horizontal Axis/Plotting Direction is applied to all Plot Paths within a specific 2D Plot. This will change whether the plot is shown as a function of distance, or X, Y or Z. Note that for line plots, selecting X, Y, or Z will constrain the location of the endpoints.

Pressing the Create button allows creation of Plot Paths of the type: Line Path, Open Curve Path, Closed Curve Path, XYZ Closed Curve Plot Path, and XYZ Curve Plot Path on the current dataset only. The maximum number of Plot Paths per 2D Plot is 10.

Import is used to import the 2D plotting information from an external file. The Export button writes out 2D plot information for only the current plot path to an external file.

The Path Style option menu can be used to select how the line in the plot should be displayed. Options are: Lines, Lines and Symbols, and Symbols.
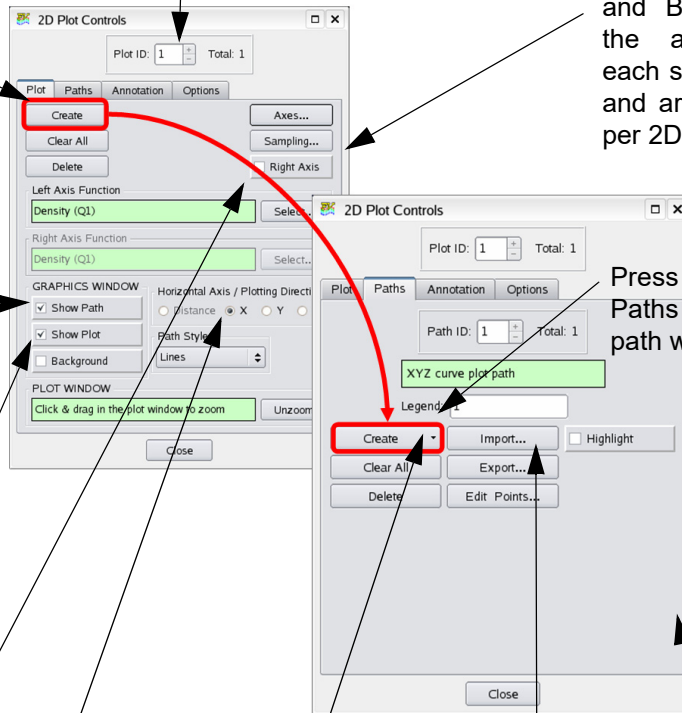
Figure 205 2D Plot Controls and 2D Plot Paths Tab

Important Notes:
The 2D Plot Paths panel can be brought up from the 2D Plot Controls panel by pressing the Plot Paths... button any time after a plot has been created.

Plot Window zooming is enabled only when there is a single plot.

When cylindrical coordinates are specified using an FVREG file (Region definition), the Horizontal Axis/Plotting Direction XYZ buttons become RTZ (Radius, Theta, Z) buttons. See **Chapter 3** for more information.

2D Plot Controls
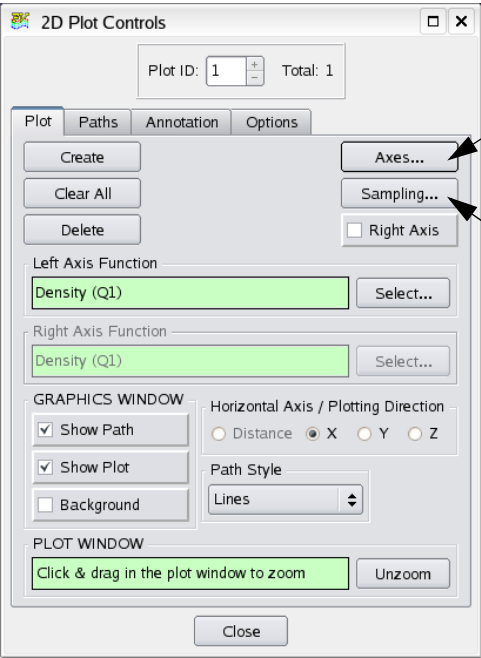
The Left, Horizontal and Right Axes each have a sub-panel which controls the labels, and minimum and maximum values for either the horizontal or vertical axes. Control over grid lines, including line styles and colors, and major and minor units for tick marks has also been added

The points may be plotted either every time a cell boundary is crossed, or at uniform spacing by using the Sampling sub-panel.

Figure 206 2D Plot Parameters

2D Plot Annotation

The content and the color of the Title and Subtitle for each 2D Plot can be specified.

The Annotation sub-panel is used to add titles and subtitles to the 2D plot, as well as displaying the date and legend on the plot.

This controls the appearance and color of the border surrounding a 2D Plot.
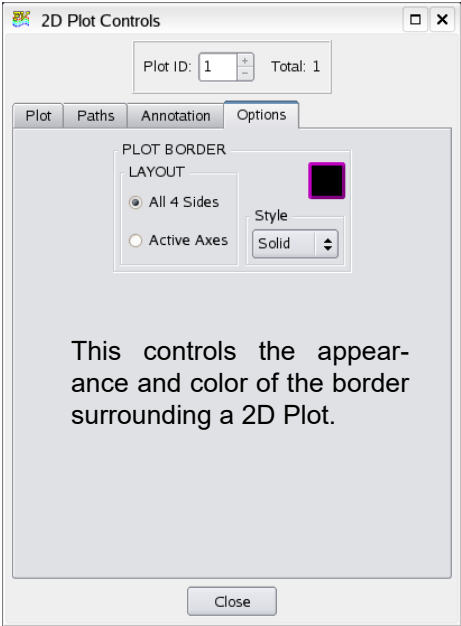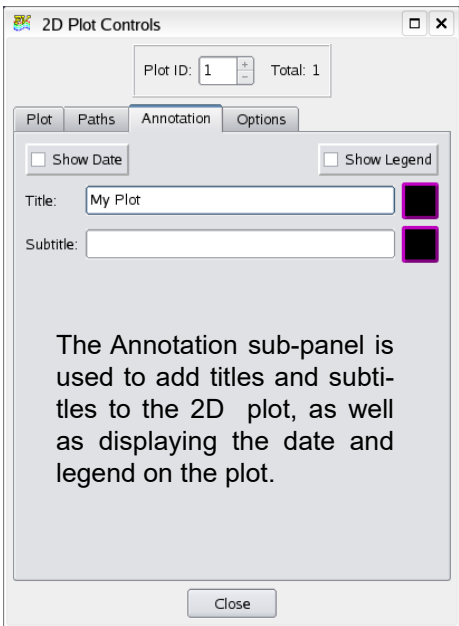
Figure 207 2D Plot Parameters
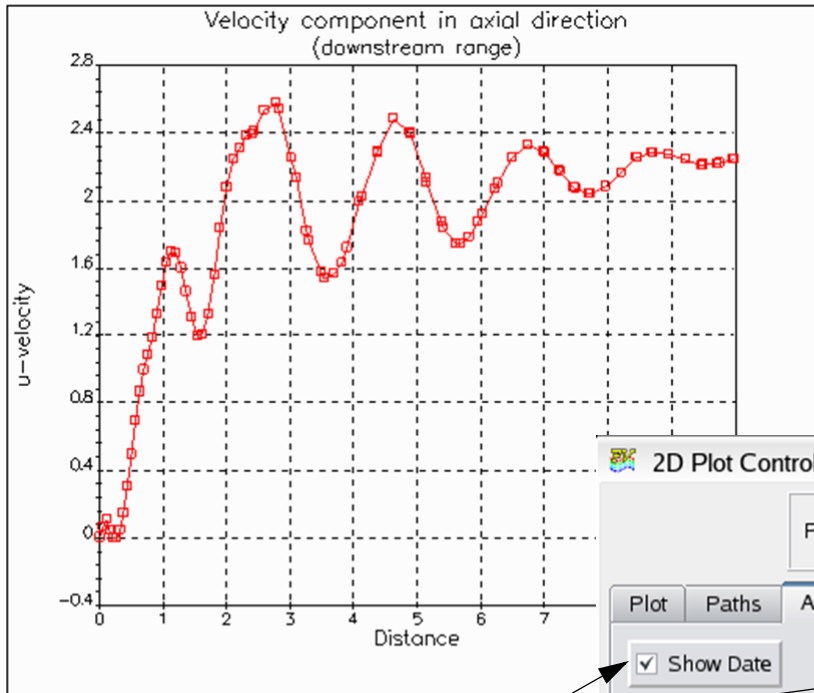
Pressing the Annotation... button on the 2D Plot Controls panel will bring up the 2D Plot Annotation panel. Annotation changes are made on a per 2D Plot basis, and can be independently specified for each 2D Plot..

The color of the TITLE and SUBTITLE for the current 2D Plot can be changed.

By default, the plot will contain the current date and a legend, if the plot style is symbols or lines and symbols. These can be turned off using these buttons.

These type-in fields can be used to display a title and sub-title above the plot.
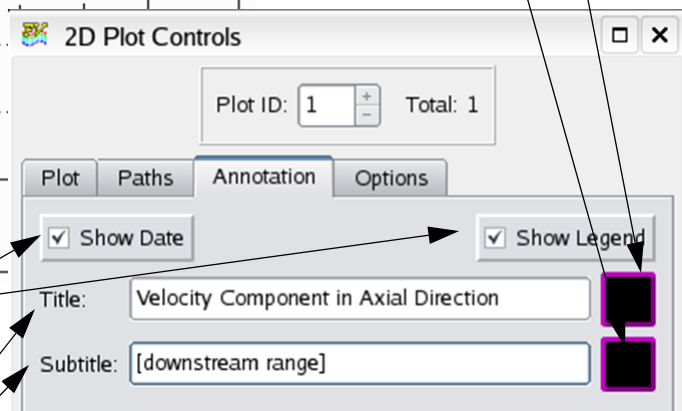
Figure 208 2D Plot Annotation Panel

2D Plot Axes

The Left, Right and Horizontal Axes panels are designed to be very similar in terms of layout and functionality. The Left Axis controls panel will be described in detail below. Comments will be applicable to control of the Right and Horizontal Axes. Axis settings are applied specifically to the current 2D Plot.

A color selection for either the plot axes, grid lines or annotations to the same color as the background is automatically prevented. If the background color is changed to the same color that is being used for the plot axes, grid lines or annotations, **FieldView** automatically attempts to recolor the elements of the plot that would be rendered invisible to a suitable contrasting color.

The AXIS TITLE can be changed by typing into this field. Backspacing to erase the current contents of a field and hitting <Return> will restore the default value.

The Minimums and Maximums of the plot can be changed by typing into one of these fields. Backspacing to erase the current contents of a field and hitting <Return> will restore the default value.

The Invert button will flip the axis so that the minimum value will be at the top instead of the bottom. The values on the axis will not change.

The color of the AXIS TITLE for the current 2D Plot can be changed.

The Normalize value will scale the axis by remapping the plot of the axis into a new range: min/scale to max/scale.
The plot will not visibly change, but the tick marks will show a new range as the value of Normalize changes. The values of Min and Max in the panel will not change as Normalize changes.

The line style (either dashed or solid), the color, and the presence or absence of major and minor GRID LINES can be specified here.

The presence or absence of TICK MARKS on the plot axes can be controlled here. The interval in between the major and minor tick marks is also specified here.

The tick mark interval is applied to the display interval of the major and minor grid lines. Any change to the major or minor tick mark interval will cause a change in the appearance of the 2D Plot if grid lines are turned on.

If the range for either the major or minor tick mark is not explicitly specified, a change to the Minumum or Maximum will cause a recalculation of the tick mark interval. This may lead to a significant change in the plot appearance - it is recommended that if a particular interval is desired, it should be typed in so that it will not be auto-scaled.

Figure 209 2D Plot Axes Panel
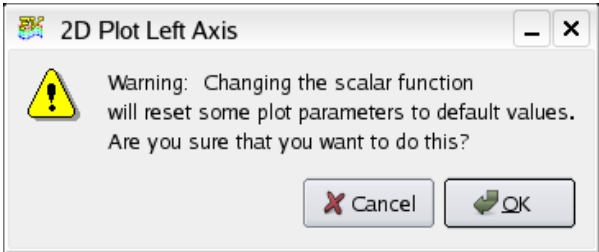
Figure 210 2D Plot Axes Warning Message

If you attempt to change the scalar function while there is a current 2D Plot, you will receive a warning message indicating that certain default parameters will be reset. For instance, if you have set a particular major or minor tick mark interval and grid lines are turned on, it is likely that your plot appearance will change significantly. Similar warning messages will be issued for any changes to either the horizontal axis through a change in the plotting direction or to the right axis, through a change in the threshold function.

2D Plot Sampling

As a plot is drawn between the defined points, the intermediate points are placed either: every time a cell boundary is crossed, or at a specific uniform sampling rate, as defined by these options.

Note: Uniform Sampling is the only valid option for 2D datasets. Cell Crossings will result in no plot.
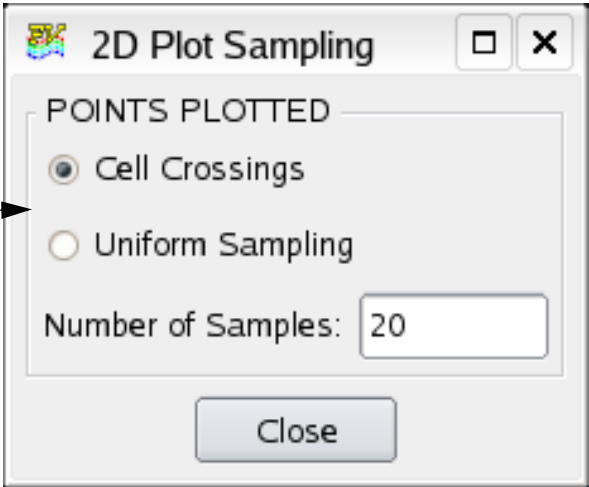
Figure 211 2D Plot Sampling Panel

2D Plot Edit Points

Point 1 and Point 2 have two functions: to allow typing in specific X, Y, and Z coordinates and to display the coordinates of the selected points. This area is only available for Line Plots.

Once the two points have been typed into the above area, the Calculate button must be pressed to create the Line Plot

This area will display messages for interactive point seeding.

Figure 212 Edit Points

2D Plot Borders

There are two options to display the external border on a specific 2D plot:

All 4 sides can be shown:

2D Plot showing All 4 Sides active as solid lines.

The color of the Plot Border for the current 2D Plot can be changed here.

The style of the Plot Border for the current 2D Plot can be changed here.
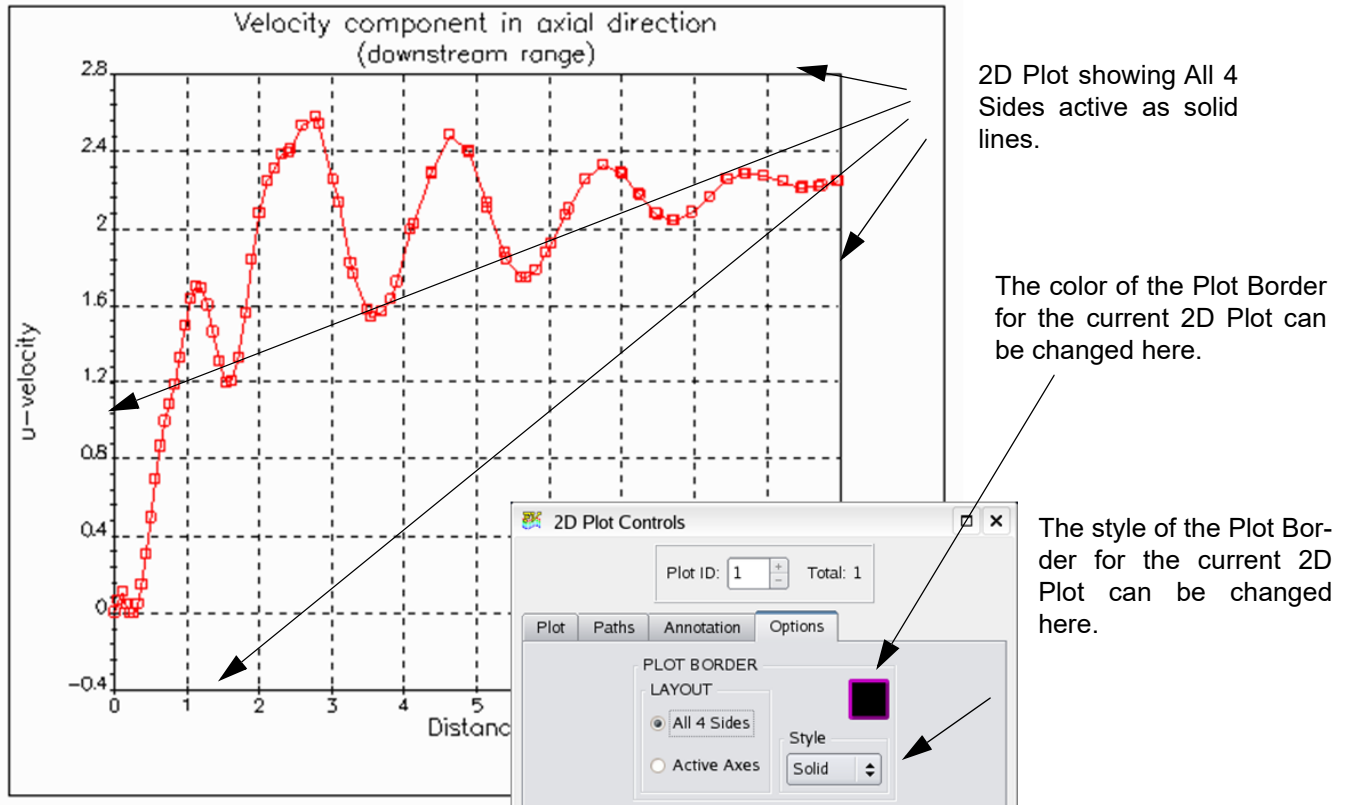
Figure 213 2D Plot Border Layout, all 4 Sides

Or only Active Axis can be shown:
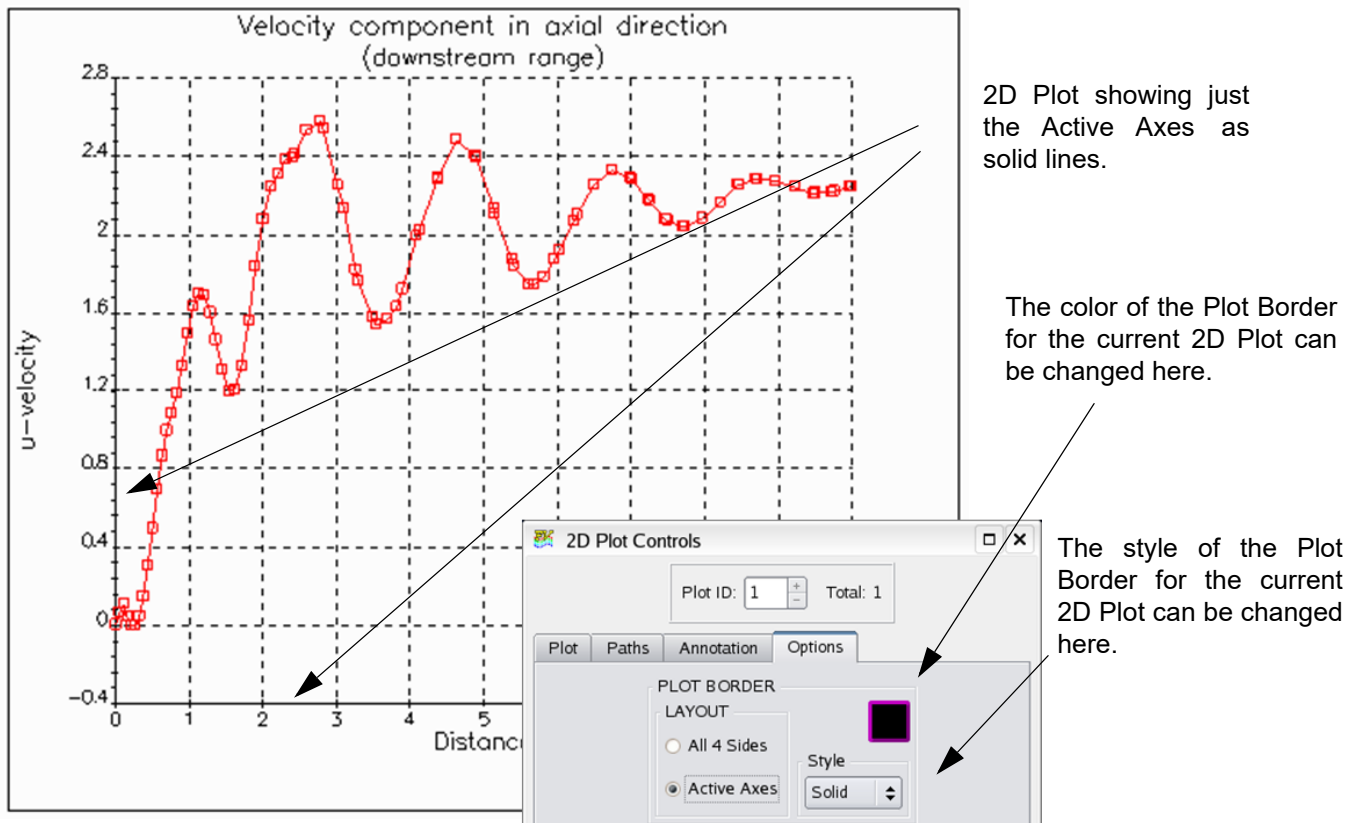
Figure 214 2D Plot Border Layout, Active Axes

2D Plot Layout for Multiple Plots

The 2D Plot window is capable of accomodating up to 9 separate 2D Plots, and within each 2D Plot, up to 10 Plot Paths can be shown. For a specific 2D Plot, the left and right axes are fixed. So, if you would like a 2D Plot of temperature versus distance and another 2D Plot of pressure versus distance, you will need to create two 2D Plots, each having one Plot Path. The layout of multiple 2D Plots in the 2D Plot window is automated and depends on the number of plots. The position of each 2D Plot is determined by the order in which it is created. The following figure shows layouts for all possible numbers of 2D Plots. For example, if there are a total of 4 2D Plots, the layout is fixed as a 2x2 configuration.

If a 2D Plot is deleted, then the remaining plots are renumbered, and the correct auto-layout is applied, based on the total number of 2D Plots.

Note: When embedding the 2D Plot Window in the graphics window, all 2D Plots are treated as a single object. Therefore, scaling and translation of the embedded plot within the graphics window will be applied to the group of 2D Plots for the current dataset.
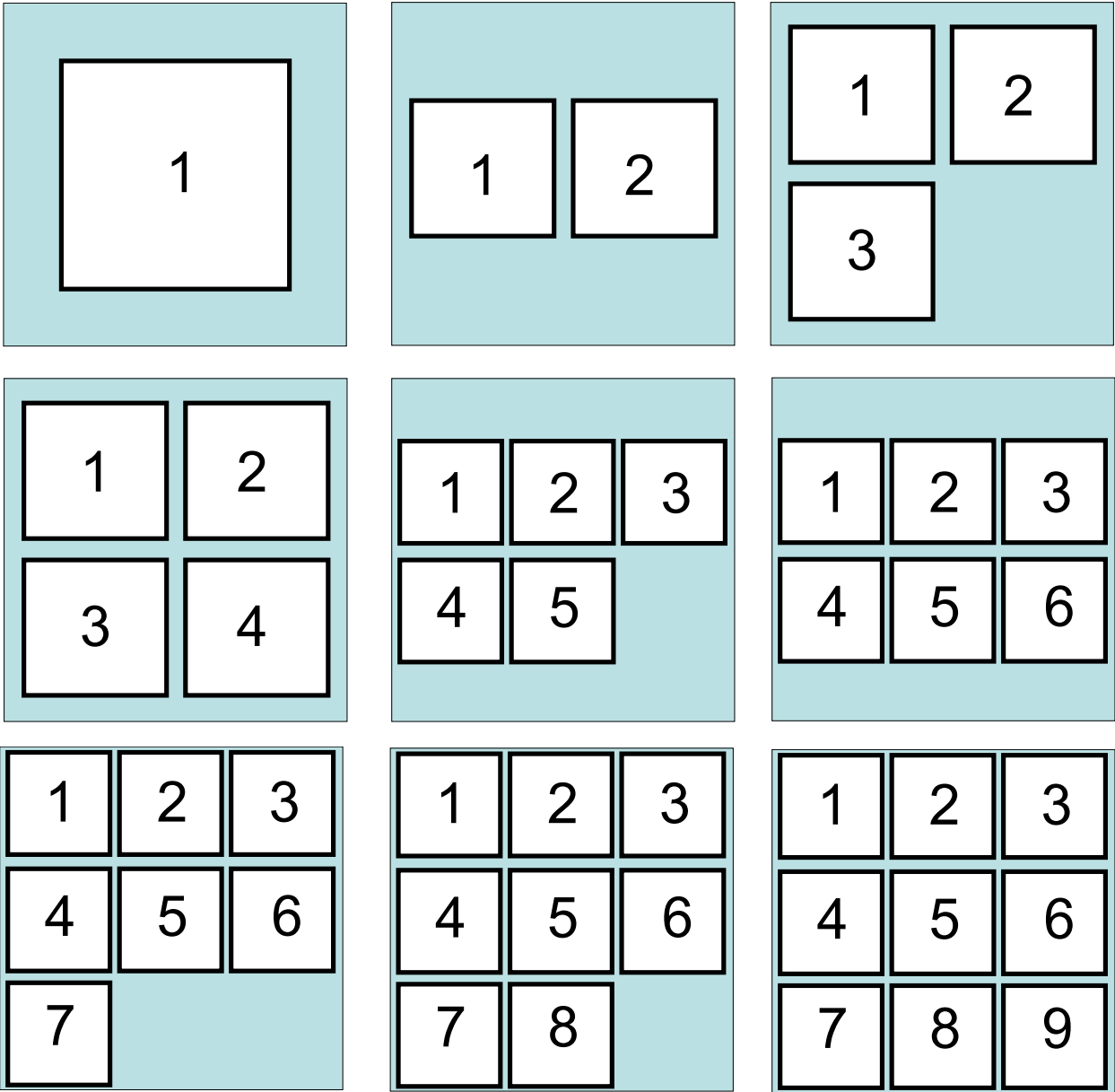
Figure 215 Auto-Layout for multiple 2D Plots in the 2D Plot Window

An example of a 4 2D Plot case, based on the Vortex Shedding Tutorial, is shown below:
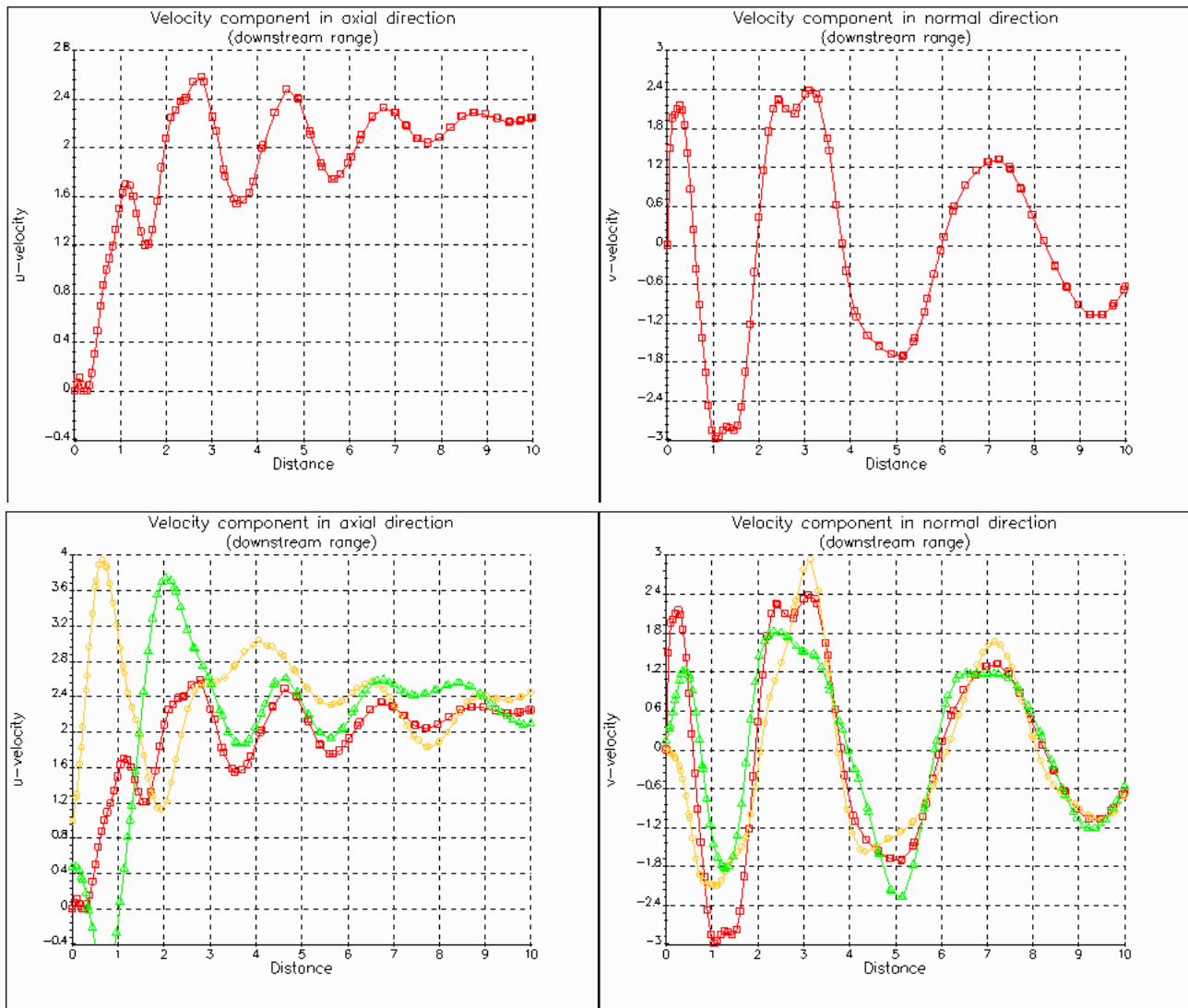
KEY
Plot #1: U-velocity vs. distance, single plot path
Plot #2: V-velocity vs. distance, single plot path
Plot #3: U-velocity vs. distance, multiple plot paths
Plot #4: V-velocity vs. distance, multiple plot paths

Figure 216 Multiple 2D Plots

Embedded 2D Plot

To display or embed the 2D plot within the graphics window use the Display Plot button on the 2D Plot Controls Panel. This button is located within the graphics window section of the 2D Plot Controls Panel.

By default, the plot has a transparent background. If you want the plot background to cover everything else underneath it, turn on the Background button. The intent of this control is to ensure that the 2D Plot sits above all other graphical elements within the graphics window.

The color of the plot lines will be the same as the current geometric color. To change the color of the lines in the plot, you will need to change the geometric color in the Colormap Specification Panel.

There is a limit of one embedded plot per dataset. If a surface plot (based on a computational surface) has been created, and has already been placed within the graphics window, you will receive the message shown in the figure below.
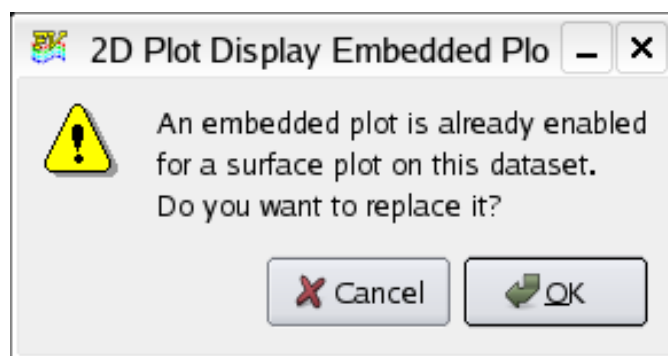
Figure 217   2D Plot Display Embedded Plot Warning

As mentioned above, only one 2D plot per dataset can be created. As a result, only one 2D plot per dataset can be embedded. To overcome this limitation, the same dataset can be re-read into **FieldView**, and additional 2D plots can be generated for each (appended) dataset. Since **FieldView** maps to appended datasets which are already read into memory, when they are recognized as identical, little additional memory usage penalty is incurred by using this work-around.

The embedded plot is designed to be in synchronization with the 2D Plot window. For instance, if you zoom into the 2D Plot window, the embedded plot within the graphics window will update at the same time. If the dataset is transient, and you are sweeping through time, the embedded plot will update for each time step as the data changes.

Embedded plots can be scaled and translated using the Transform Controls by setting the object from the normal default Object: World to Object: Plot. The left mouse button is used to translate the embedded plot, and the right mouse button is used to control the plot size within the graphics window. There are limits to the location of the plot within the screen, and on the minimum and maximum plot size.

2D Plots and Transient Data

All plot axes range (minumum and maximum), as well as the settings for the major and minor tick marks are automatically updated to keep the 2D plot fully displayed in the 2D plot window. For example, suppose you have plotted temperature at the first time step and the range is 270 to 330K. If either the minimum or maximum value changes during a transient sweep, the vertical axis will be updated. While this does offer the benefit of dynamically scaling the 2D plot, the side effect is that the plot itself may jump around during the sweep, making it more difficult to interpret. To fix the plot axes, turn the Fix Axis button on for each or all of the 2D plot axes you want to fix during a transient sweep. From the 2D Plot controls Plot tab, click on Axes, and then navigate to the plot axis that you would like to fix.
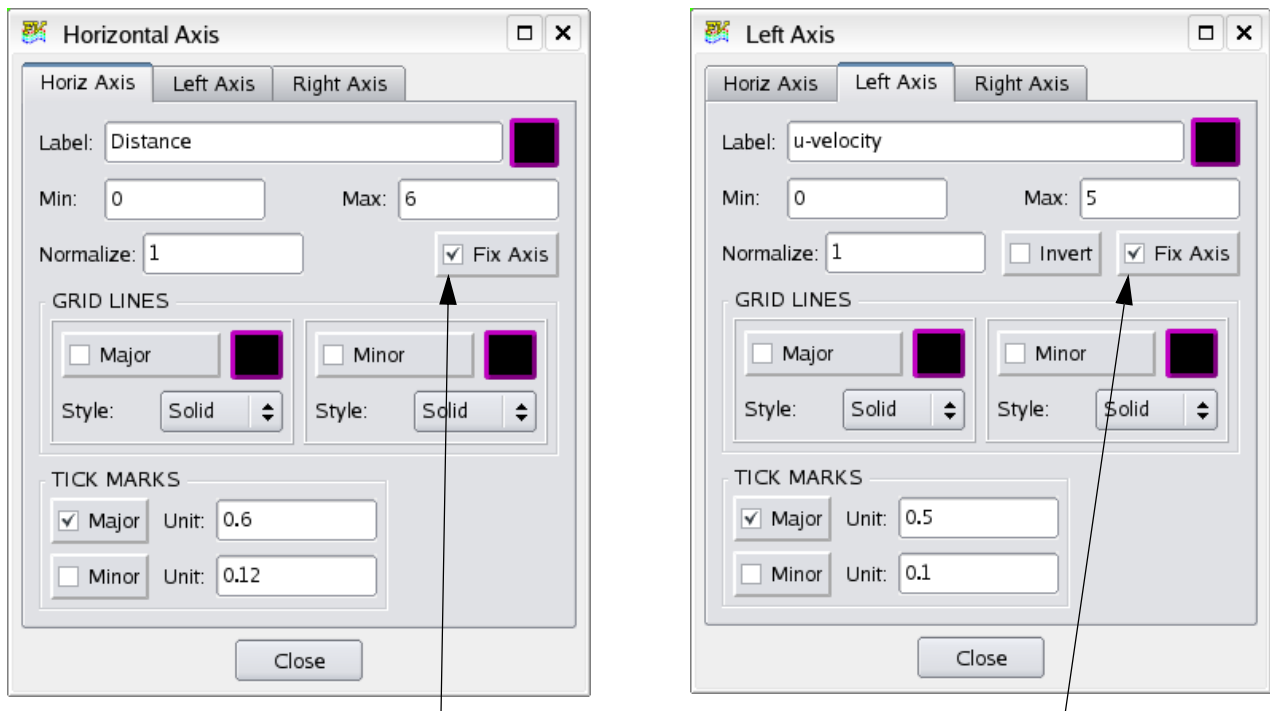
Set the Fix Axis button to independantly fix the default auto-range behavior for the plot Min and Max, and the Major and Minor tick marks.

Figure 218 Fix Axis Button for Transient Sweep

The status of the Fix Axis button is stored in the 2D Plot restart file. Older 2D Plot restarts will use the default behavior; the Fix Axis button will be turned off.

 Using the 2D Plot Paths Panel

FieldView supports six different types of 2D plot paths: Line Path (volume), Line Path (surface), Open Curve Path, Closed Curve Path, XYZ Closed Curve Path, and XYZ Curve Path. These are found on the Create… pull-down menu on the 2D Plot Controls Panel (**Figure 219**).
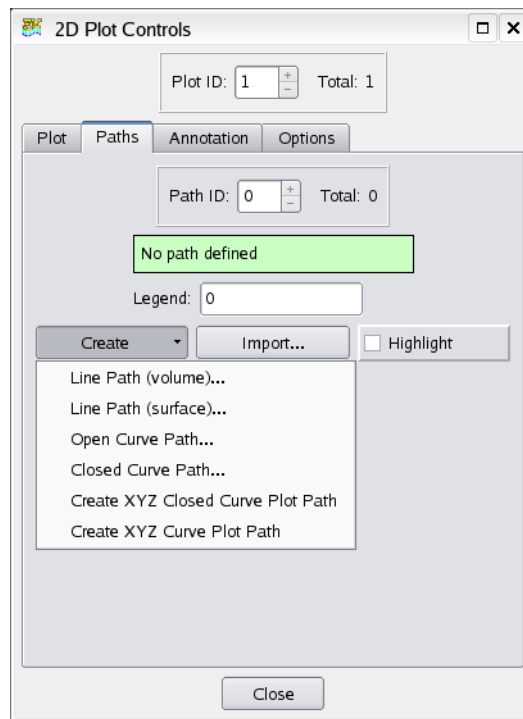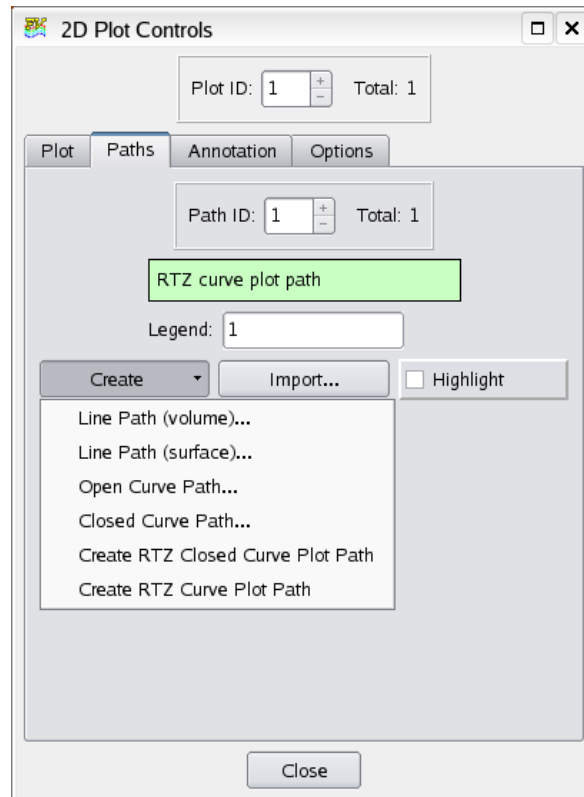
Figure 219 Create Pull-down Menu

The paths are defined as follows:

Line Path: (volume) Select two points in space. **FieldView** will draw a line between the points and plot the points along the line in XYZ space.

Line Path: (surface) Select two points on a flat surface. **FieldView** will draw a line between the points and plot the points along the surface.

Open Curve Path: Select 3 points on a single surface (i.e. computational, boundary, iso). **FieldView** will construct the plane defined by the three points, intersect the plane with the selected surface, and plot the intersection points, from point 1 through point 2 and ending at point 3.

Closed Curve Path: Select 3 points on a single surface. **FieldView** will construct the plane defined by the three points, intersect the plane with the selected surface, and plot the intersection points, from point 1 to point 2 to point 3 and back to point 1 again.

XYZ Closed Curve Path: Select a surface and an X, Y or Z plane at a specified location, if possible. **FieldView** will compute the intersection of the plane with the selected surface and plot the intersection points. The plot will be "closed" in that the first point and last point plotted will be the same.

XYZ Curve Path: Select a surface and an X, Y or Z plane at a specified location, if possible. **Field-View** will compute the intersection of the plane with the selected surface and plot the intersection points. This differs from a XYZ Closed Curve Path in that the start and end plot locations will not be the same point.
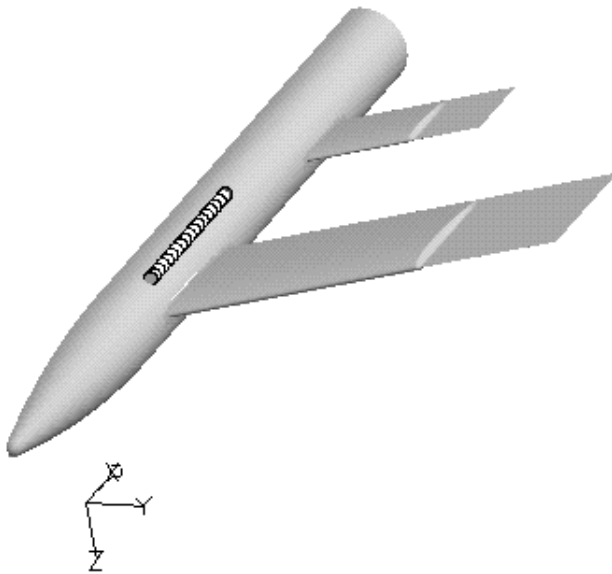
Line Plot   Open Curve Plot
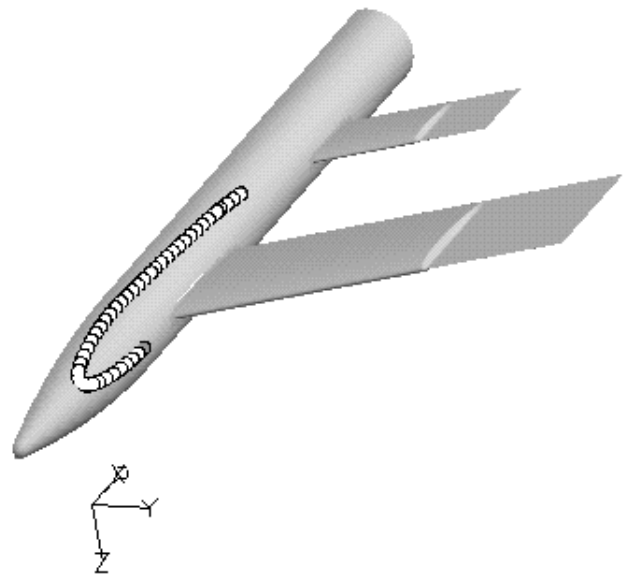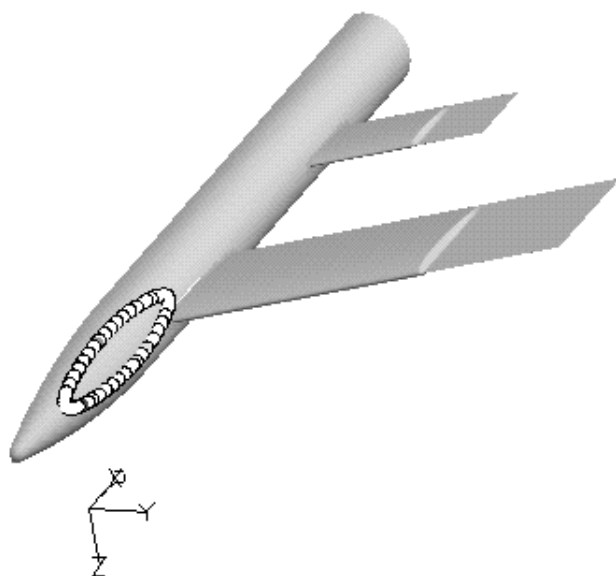
Closed Curve Plot
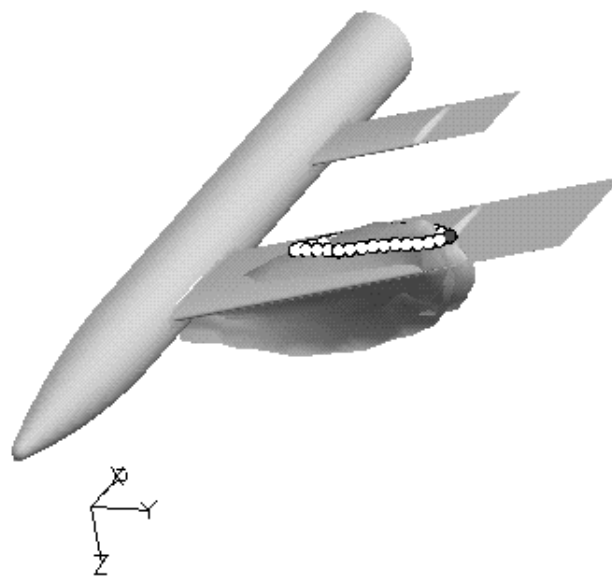
XYZ Closed Curve Plot
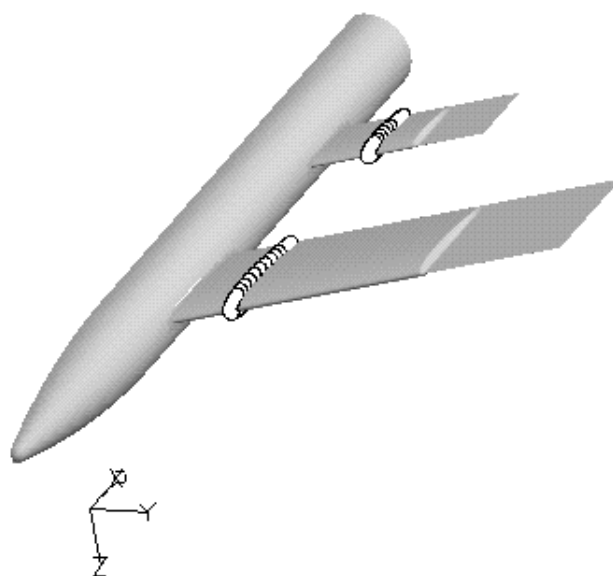
XYZ Curve Plot

Figure 220 XYZ 2D Plot Examples

Using the 2D Plot Path Controls in Cylindrical Coordinates

When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ Closed/ Curve Plot Path options become RTZ (Radius, Theta, Z) plots. See **Chapter 3** of the **Reference Manual** for more information. There are different plot options when cylindrical coordinates are specified.

Figure 221 RTZ Create Pull-down Menu

The paths are defined as follows:

Line Path: (volume) Select two points in space. **FieldView** will draw a 'line' (i.e. constant *radius*) between the points and plot the points along the 'line' (arc) in RTZ space.

Line Path: (surface) Select two points on a flat surface. **FieldView** will draw a line between the points and plot the points along the surface.

Open Curve Path: Essentially the same as for XYZ plots.

Closed Curve Path: Essentially the same as for XYZ plots.

<u>RTZ Closed Curve Path</u>: Select a surface and an R, T or Z plane at a specified location, if possible. **FieldView** will compute the intersection of the plane with the selected surface and plot the intersection points. The plot will be "closed" in that the first point and last point plotted will be the same.

<u>RTZ Curve Path</u>: Select a surface and an R, T or Z plane at a specified location, if possible. **Field-View** will compute the intersection of the plane with the selected surface and plot the intersection points. This differs from a RTZ Closed Curve Path in that the start and end plot locations will not be the same point.

See the RTZ 2D Plot Examples in **Figure 222** below.
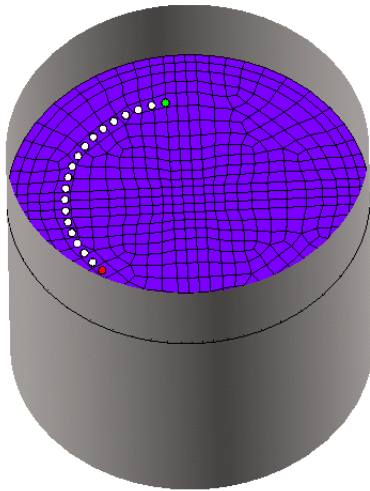
Line Plot (constant radius)   Open Curve Plot
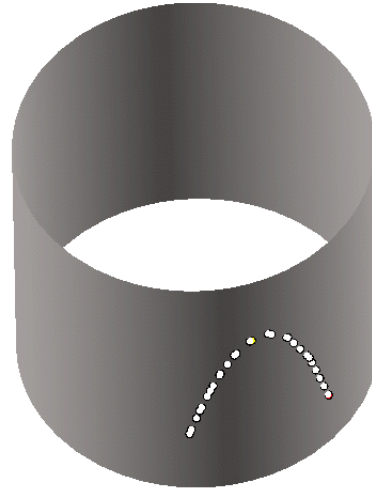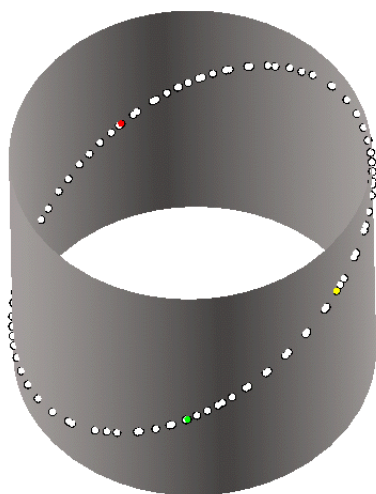
Closed Curve Plot
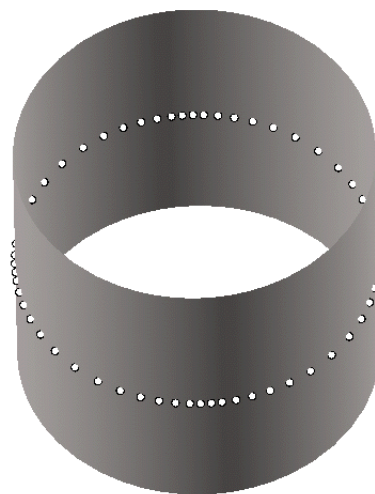
RTZ Closed Curve Plot
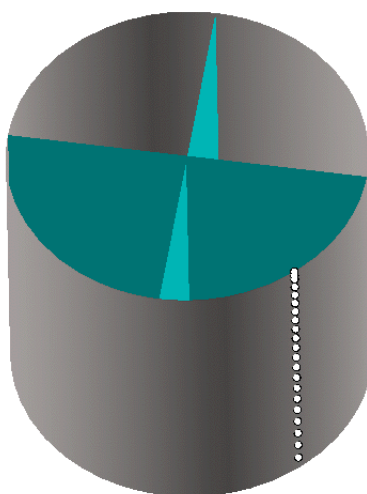
RTZ Curve Plot (theta = 315 degree cut

Figure 222 RTZ 2D Plot Examples

Frequently Asked Questions

How do I create a 2D Plot?

In order to create a 2D Plot, first you must specify the type of plot you wish to make by pressing the Create button, and specifying the plot type. A 2D plot can only be created on the current dataset. The techniques described below are essentially the same for RTZ 2D Plots.

Line Paths: After pressing Create, you need to specify the start and end locations for the plot. This may be done either interactively, in the graphics window (see the next section for more information), or by typing in the exact locations. Once the two points have been selected, the plot will be displayed in the 2D Plot Display.

Open and Closed Curve Paths: After pressing Create, you need to specify the three points to define the slicing plane. These three points must be selected interactively in the graphics window (see the next section for more information), and must all lie on the same surface. Once the three points have been selected, the plot will be displayed.

XYZ Closed Curve and XYZ Curve Paths: After pressing Create, you need to specify which surface you wish to plot along. This is done interactively by clicking on the surface in the graphics window. Next, you must specify whether you wish to slice the surface with an X, Y or Z plane, and the exact value for the plane (i.e. an X=5 plane). Once this location has been specified, the plot will be displayed. The Cutting Plane is aligned with one of the 3 axes of the dataset. The Cutting Position specifies where, in the range of the axis chosen, you wish the plane to slice through.

How do I pick the end point locations in the Graphics Window?

To pick a plot location in the graphics window, simply move the mouse to position the cursor in the graphics window over the point you wish to plot. Pressing the control key and M1 at the same time (CTRL-M1) will place the first plot point (Point 1) at the current location. You then use CTRL-M1 to seed the second and third points, if necessary. The X, Y, and Z coordinates of the end points will be displayed for Line Plots only. After the points have been selected, you may move them by using the Edit Points sub-panel. To do this, click on the point with CTRL-M1 and while holding M1 down, move to the new desired position. This will change the location of the point, although the point will *not* be dragged during the mouse motion.

How can I have a second vertical axis?

By default, the points for the 2D plot are shown based on the variable in the Scalar Register in **FieldView** (see the Functions panel for more information). By pressing the Secondary Axis button, another plot of the same points will be displayed, based on the variable in the Threshold Register. The second plot will be shown as a dashed line in the 2D Plot Display.

How can I read/write the plot information to/from a file?

By pressing the Export button, you will be prompted for the name of an output file for dumping information about the current plot path only. The format for this file is shown in **Appendix G** of the **Reference Manual**.

The Import button is used to read external data into the 2D Plot Panel in **FieldView**. This data may have come from Point Probe, experimental data or another program. The format for this file is the same as the Export file, and is shown in **Appendix G** of the **Reference Manual**.

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the XYZ Closed/Curve Plot Path options become RTZ (Radius, Theta, Z) plots. See **Chapter 3** of the **Reference Manual** for more information.

Note: The plot label information of the imported plot will be used for the vertical-axis label. If there is already a 2D plot when the imported plot is displayed, the vertical-axis label will be changed to that of the imported plot file. A warning pop-up will be issued if the Scalar functions of the current 2D plot and that contained in the imported file differ.

What does the Legend type-in field do?
When a plot is created, and displayed as Symbols or Lines and Symbols, a legend is shown at the bottom of the plot to help distinguish the plots. The Symbol for the plot is shown along with a number for the plot itself. If you wish to have your own descriptor for the plot, you may type it into the Legend field on the 2D Plot panel. The string that you type will then be shown next to the symbol for that plot, replacing the default integer value.

How do I specify end point locations numerically?
For Line Plots only, you may type in the X, Y, and Z coordinates (and dataset number if necessary) on the Edit Points sub-panel. Once the two points have been typed in, press the Calculate button to update the Line Plot display.

How do I see the line/curve I am plotting along in the graphics window?
When points are selected interactively, the end points of the plot are shown using temporary markers in the graphics window. These markers will be deleted on the next graphics refresh. In order to see the exact locations of all of the plotted points, turn on the Display Points option on the Line Plot Controls panel. This will leave a marker at each point location in the graphics window until Display Points is turned off or the panel is exited.

What is the difference between 2D Plot and the Surface Plot on the Computational Surface Panel ?
Surface Plot, on the Computational Surface panel, will plot only along lines of I, J, and K, and only on Computational Surfaces. This allows for plotting along surfaces in structured grids that may be curved in Cartesian space. 2D Plot will plot along any arbitrary straight line; or along a cut through a surface, and plots the points in X, Y, Z coordinates.

How can I set my horizontal axis to X, Y, or Z?
The Horizontal axis controls on the 2D Plot Panel have a different meaning depending upon the type of Plot you are creating. For Line Plots, setting the Horizontal axis will restrict the second end point to have 2 of the same coordinates as the first point. This will force the plot to be along a straight line in X, Y, or Z.

For Curve Plots, setting the Horizontal axis will simply change the axis on the 2D plot itself such that the horizontal axis values of each point on the plot will be in X, Y, or Z coordinates and not total distance between the points. The Distance choice will be disabled if any XYZ Curve plot paths are defined.

How can I change the distribution of the plotted points between the two plot locations?
By default, 2D Plots are created by sampling the scalar variable each time a cell boundary is crossed by the line or curve connecting the points. If an even spacing is desired, press the Sampling button on the 2D-Plot panel, and then select Uniform Sampling on the Sampling sub-panel. You can then specify the exact number of points to be plotted between the beginning and end points. This is disabled if any XYZ Curve plot paths are defined. For 2D datasets, Uniform Sampling is the only valid option. Cell Crossings will result in no plot.

How can I change the axis labels on the plot?
Pressing the Axes button will bring up the 2D Plot Axes sub-panel. This panel will enable you to type in new labels or new values for the Horizontal or 2 Vertical minimums or maximums.

How can I zoom in on an area of interest in the 2D Plot Display?
To zoom in on a specific area in the 2D Plot Display, move the mouse so the cursor is located in the window. Press and hold M1 and, as the mouse is moved, a zoom box will be drawn in the 2D Plot Display. To zoom in on the area surrounded by the box, release the M1 button. To zoom back out all the way, press the Unzoom button on the 2D Plot Controls panel. Zoomings can be done multiple times.

Possible Problems

*Why does **FieldView** "beep" when I try to select a begin/end point?*
When using the interactive point selection, you must click on a valid object in the graphics window that belongs to the current dataset. If you select a point in empty space or click on an invalid object (such as a surface belonging to a different dataset, or legend or title), a "beep" will be sounded and the point will not be selected.

Why does the scalar variable plotted differ from the one displayed on the surface where I created my plot?
The plot displayed in the 2D Plot Display is based on the variable that is currently loaded in the Scalar Register. For example, if you create one surface displaying Pressure, and a second surface displaying Temperature, then create a plot on the first surface, the plot will show Temperature, not Pressure. This is due to the fact that Temperature is currently loaded into the Scalar Function Register.

Why doesn't my line plot follow the surface where I selected my end points?
The Line Plot may not be located on a particular surface, even if both end points are located on the surface. This is because the Line Plot is always a straight line connecting the two end points. If a surface which contains both end points is not flat, the plot will not follow the surface. For curved surfaces, you should be using either an Open or Closed Curve Plot, which will follow along the curved surface.

Why are there gaps in my line plot?

It is possible for the line to cross an area where there is no grid, or where the grid has been Iblanked. If this occurs, you will see a gap in the plot where it crosses these regions.

Why does my Display Path not run between the points that I picked?
For surface-transformed surfaces, the Display Path will be displayed between the end points at their *non- transformed* locations in the dataset.

Error Conditions

`Could not find a valid path connecting the 3 points...`
There are 3 reasons why you may get a message similar to the one shown above:

*   Curve plots are defined as the intersection between a plane and a surface. If the surface itself is relatively flat, the plane will graze the surface rather than intersect it, and a plot cannot be generated. For flat surfaces, you should use the Line Plot option rather than the curve plot.
*   For surfaces that are topologically 2D, a closed curve plot may not be a valid option. This is because it may not be possible to create a closed curve from point 1 to point 2 to point 3 and back again to point 1.
*   There may be gaps in the grid somewhere along the specified path.

`Points are collinear.  Cannot calculate the curve.`
When selecting the three points for a curve plot, the points may not all lie on the same line. If a straight line is desired, the Line Plot option should be used.

`Could not locate point on curve.`
Curve plots are defined as the intersection between a plane and a surface. If the surface itself is relatively flat, the plane will graze the surface rather than intersect it, and a plot cannot be generated. For flat surfaces, you should use the Line Plot option rather than the curve plot.

`You did not select a valid surface for curve plotting`
Curve plots are only valid on Boundary Surfaces, Iso-Surfaces, and Computational Surfaces.

`Warning:  the selected point is outside the dataset`
This is often caused by picking on a surface which has been subsetted. Since subsetted surfaces are only approximations of the actual surface, the actual location of the selected point may be outside of the dataset. For more accuracy, you should ensure that your surface is not subsetted. It may also be caused by picking in the interior of a curved face of a grid boundary. Try picking only on grid lines (mesh lines).

`Too many paths - new path would exceed the limit.`
The total number of plot paths has a limit of ten (10) per dataset. In addition, lines within the 2D Plot window appear in color.

`Picked point is not in current dataset`

When either typing in or interactively selecting the plot locations, the points must all lie within the same dataset. If the points are in different datasets, an error message will be displayed and the point will not be selected.

2D Plot SCRIPT Commands

Two script commands help to automate 2D Plots.

`PLOT_SIZE width height`

This command along with its required arguments will cause the 2D Plot Window to be resized to the specified values. The width and height specifications are for pixel dimensions. The command when used with the `PRINT PLOT` script command, will generate an output file based on the specified dimensions.

`SELECT PLOT plotnum [pathnum]`

This command will permit selection of a particular 2D Plot from one of many 2D Plots based on the argument supplied for `plotnum`. The optional argument `pathnum` permits selection of a specific plot path within a specific 2D Plot. If a non-existent value is specified for either the `plotnum` or `pathnum`, the script will silently pass over the error.

Important Note: The `SELECT PLOT` command from a script may not work correctly if there are multiple datasets unless the `SELECT DATASET` command is present. A warning message will be printed to the xterm window in this case.

The `SCRIPT` command, `SELECT LINE [pathnum]` is deprecated in favor of the `SELECT PLOT` command. However, it will still work and will be interpreted to select pathnum on the current 2D plot for the current dataset.

Known Limitations

2D Plot Window zooming is enabled only when there is a single 2D Plot for the current dataset.

There is no control over font size or font style, symbols or line colors, used for Plot Paths, or line thickness.

There is no direct **FVX** support for 2D Plotting. The **FVX** `fv_script` command can be used to execute 2D Plot RESTARTS.

Changes to 2D Plotting in FieldView 13

First, the 2D Plot Controls panel has changed, with functionality being accessed through four tabs.
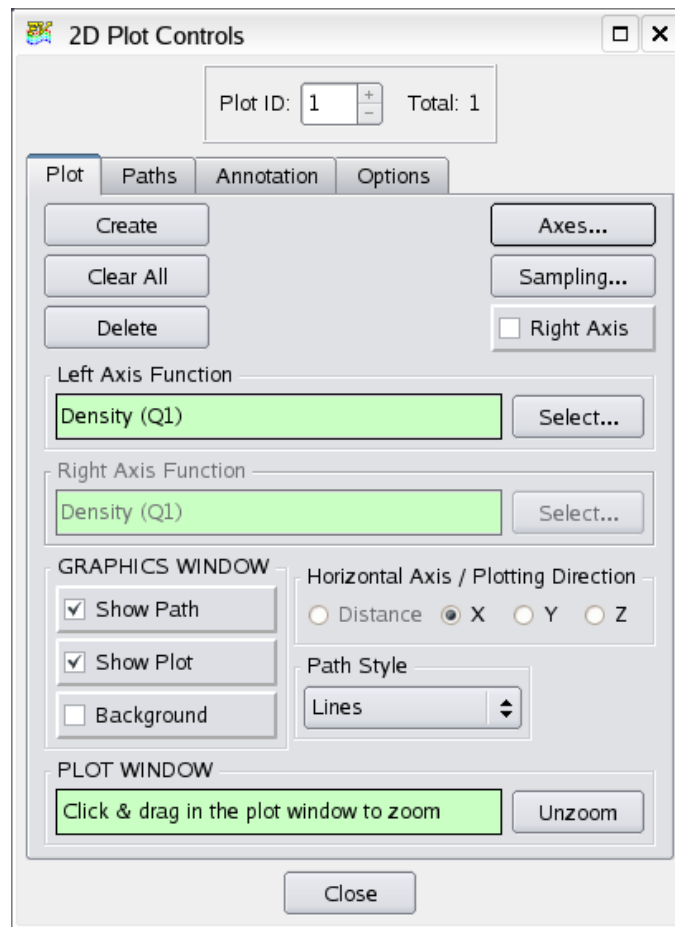
Figure 223 2D Plot Panel

The left axis and right axis functions are also displayed directly on the Plot tab of the 2D Plot Controls panel. The Plot tab lets you manage plots; plot paths are contained within plots. Since the path style is an attribute of the plot and not each individual path within a plot, controls to change the line style appear on the Plot tab.

For plot paths that are surface-based, the scalar values used in the plot are calculated by probing the surface scalar values instead of the entire volume. This change improves 2D plotting by making it more accurate and allows it to work on boundary-only grids, or boundary-only datasets like XDB imports. Plot path types which are surface based include: Open Curve Path, Closed Curve Path, XYZ (RTZ) Closed Curve Plot Path and XYZ (RTZ) Curve Plot Path.   Since Line Paths can span across a dataset volume and are not always limited to surfaces, we use volume scalar functions for Line Paths.

Curve plot paths that are created interactively by picking points on a surface have specific rules for the default left and right plot axis functions. When a curve plot is first created, the scalar on that surface is

used for the left axis. If the surface does not have scalar data, the 2D plot is created using the current scalar function register. The scalar used by the plot path can be easily modified by using the select button (Left Axis) to choose a scalar from the Function Selection panel. Until a plot path is created, the Left Axis and Right axis functions will be greyed out. Similarily, if all paths are deleted from a plot, the axes will be Undefined and the function selectors are again greyed out.
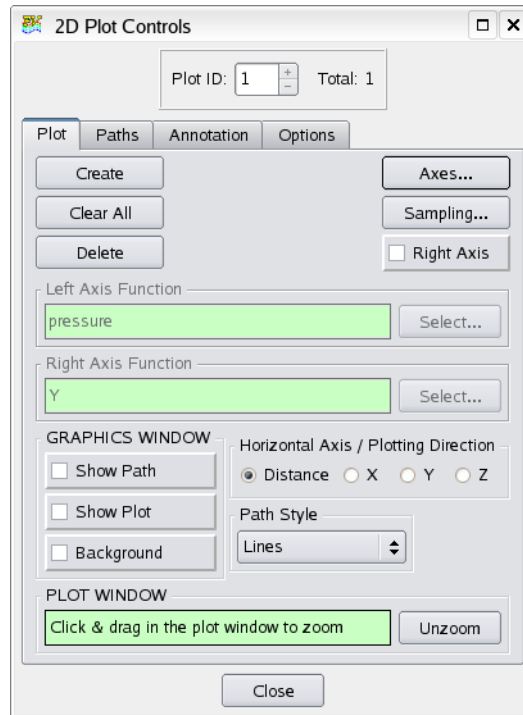
Figure 224 2D Plot Path Undefined

Changing the scalar on the surface, after a plot path is created will not change the scalar being used for a plot path. Plot paths added to an existing plot path within a 2D plot will use the same scalar function as the first plot path. This implementation allows you to create surface based plots on different surfaces using a common scalar function.

There are also considerations with regard to the right axis function. When a curve plot is first created, the threshold on that surface is used for the right axis. If the surface does not have threshold data, the 2D plot is created using the current threshold function register. The right axis scalar used by the plot path can be easily modified by using the select button (Right Axis) to choose a scalar from the Function Selection panel. It is possible to toggle the visibility of the right axis off, change the right axis function and then toggle the visibility back on. When this is done, the right axis will match what was most recently selected.

Plotting of face-based scalars is possible, as is illustrated in the Figure below.
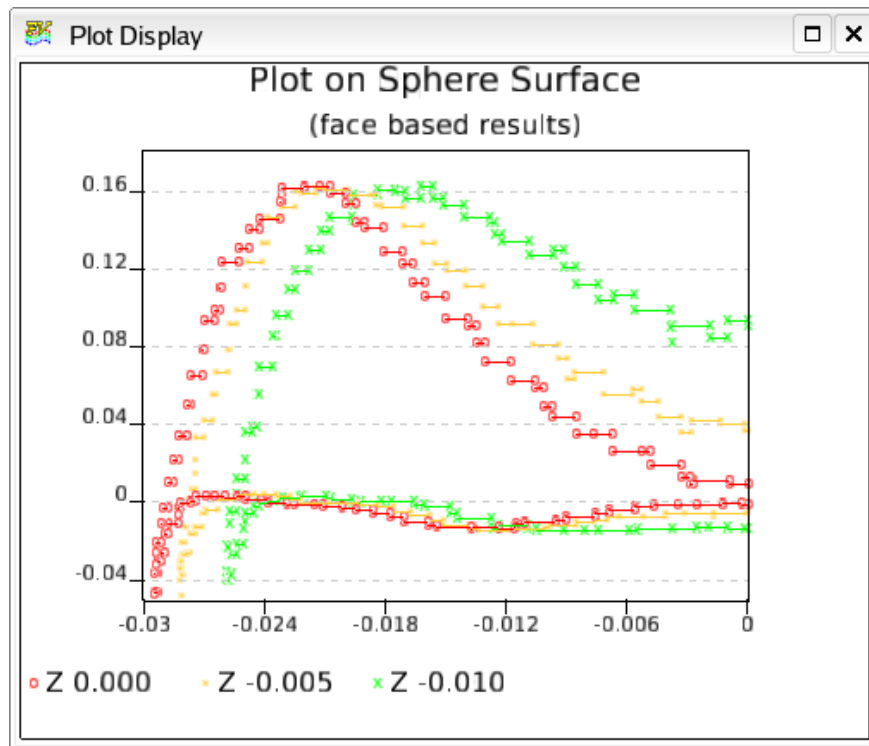
Figure 225 2D Plot Path Undefined

In order to a get a plot for a face-based boundary variable, all of the paths must be curve plot paths or external (imported) plot paths -- no line plots are permitted. Furthermore, all of the curve plots must be based on boundary surfaces. When the above conditions are true, the function registers behave as if the Boundary Surface panel was the current panel. The function selectors will show any face-based boundary variables in addition to the usual variables. If the first curve plot path is defined on a boundary surface that has a boundary variable stored as its scalar, then that boundary variable will be used as the initial left axis for the plot. Similarly for the initial right axis. Regardless of the initial axis, you should be able to change back and forth between boundary variables and ordinary variables as long as the above conditions are true. Boundary variables are plotted as horizontal line segments since the values are constant within each boundary face. This looks like stair-steps, except that the stair-steps do not have vertical lines connecting them. Symbols (if turned on) are drawn at both ends of each horizontal line segment. If the left or right axis is a boundary variable, then exporting a plot path will result in a separate "zone" for each line segment, in order to preserve the stair-steps.

The ability to select the function to be used for either the left axis or the right axis can be specified on the 2D Plot controls panel. However, when the 2D plot window is up, you can also change the func-

tions on the 2D plot using the Function Specification Panel.   The Left Axis function and the Scalar

Set the Left Axis function using either the Select... button on the 2D Plot Controls panel or the Scalar register selector on the Function specification Panel.

Set the Right Axis function using either the Select... button on the 2D Plot Controls panel or the Threshold register selector on the Function specification Panel.
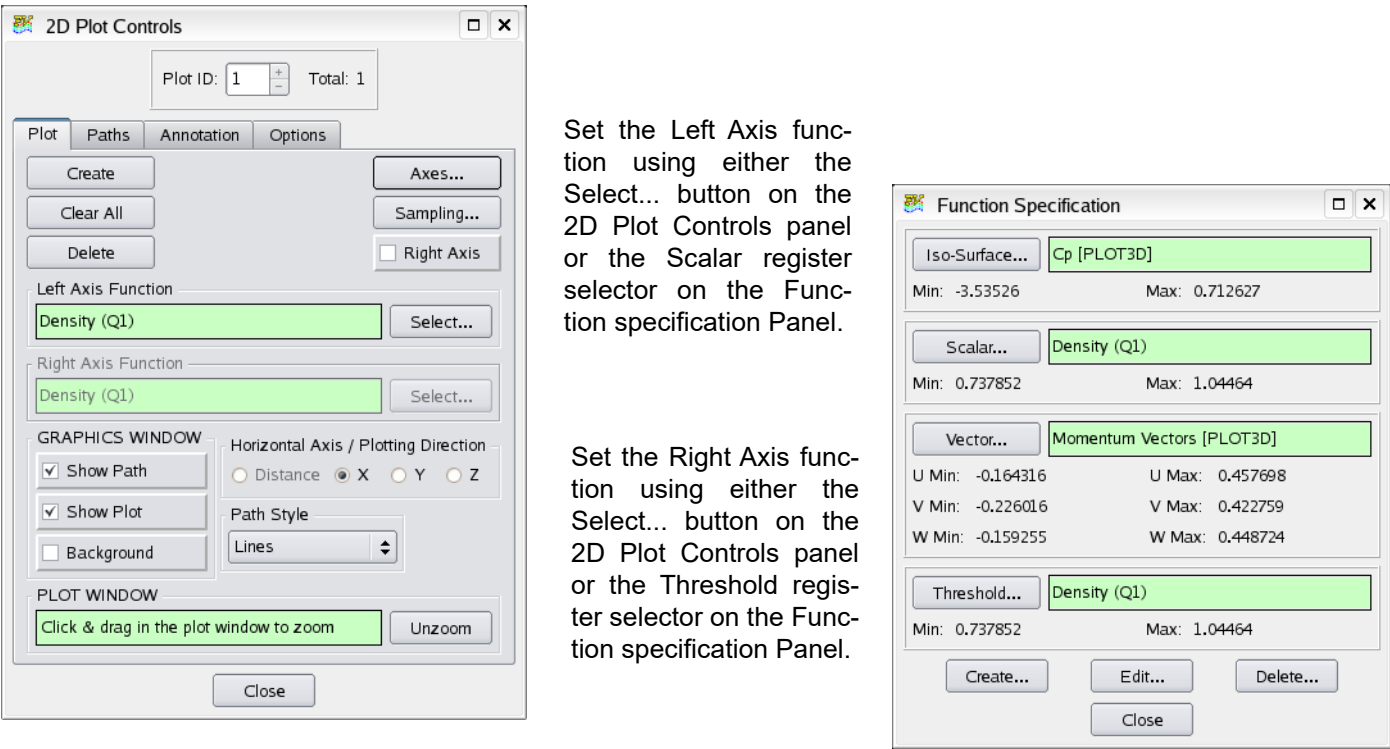
Figure 226 Setting the Left and Right Axis Functions

Register field will always be consistent with each other when the 2D Plot Window is up. The same consistency holds true for the Right Axis function and the Threshold register field.

Chapter 13

Point Probe

13

The Point Probe panel (**Figure 227**) is used to interrogate the quantities at an arbitrary location in the current dataset. By clicking in the graphics window, or typing in a specific value, the scalar and vector quantities at that point can be displayed.

For information on using **FieldView** to analyze results from MAT-File, CSV format files, or plain text, refer to the Tools menu entry **Point Query...** in **Chapter 14**.

Point Probe Panel

Displays the current contents of the current dataset's registers.

Use the Control key and the left mouse button (CTL-M1) to probe in the window.

Display Point: When this button is on, a permanent marker will be displayed in the graphics window showing the location of the point that was probed.

Output To File: When this option is turned on, the results of the point probe will be written to a specified file. These results can then be used as input to the 2D Plot Controls panel. Note that if this button is checked and you sweep transient data, the output file will contain data for each time step of your transient solution.

The outermost fields show the minimum and maximum values of the X, Y, and Z coordinates of the data. You may not type in to these fields.

The center area has two functions.   It allows you to type in a specific X, Y, Z location to be probed, and displays the coordinates and register values (if any) of the point that has been picked.

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition),   the   XYZ   labels become RTZ (Radius, Theta, Z) labels. See **Chapter 3** for more information.

When a location is probed, this area shows the values of I, J, K (if structured) and all scalar and vector functions currently loaded in the data registers for the current dataset, evaluated at the probe location.

Note: Region information will only be present for dataset types that have a valid FVREG file defined.

Pressing the Calculate Button after typing in valid coordinates will probe the **volume** data. It is not necessary to use this button when selecting points interactively in the graphics window, where data is retrieved directly from the visualization object, rather than the volume.

Figure 227 Point Probe Panel

Note: This panel looks slightly different for unstructured grids.

Important Note: The values returned by the Point Probe are the scalar and vector quantities currently loaded into the function registers for the current dataset only. For example, if the user creates one surface displaying Pressure and a second surface displaying Temperature, and then probes the first surface, the value of Temperature will be returned, not Pressure. This is because Temperature is currently loaded into the function register. If no functions are loaded in a particular register, this section can be blank.

Face Data Note: When Face Data are present, probing on any surface, including a Boundary surface, will return only the normal nodal "volume" register values as well as face data results.

Frequently Asked Questions

Why should I use the Point Probe panel?
The Point Probe panel will return the exact scalar and vector quantities at any location within the *current* dataset. This will enable you to examine boundaries for proper boundary conditions, find exact values of temperature and pressure on the leading edge of a wing, or within a vortex, or verify the accuracy of a solution by comparing experimental data to the computational data at the same location.

How do I Probe in the Graphics Window?
To probe a location in the graphics window, simply move the mouse to position the cursor in the graphics window over the location you wish to probe, and press the control key and left-most mouse button (CTRL-M1). The X, Y, and Z coordinates will be displayed on the Point Probe panel, as well as the I, J, K coordinates (for structured datasets) and the values of the variables currently loaded into the function registers, at the specified point. You can probe in the graphics window as long as the cursor is positioned over an object in the graphics window, such as a computational surface, iso-surface, streamline, etc. If you click on empty space or an invalid object (such as a legend or title), a "beep" is sounded and the point is not probed.

How do I Probe a Specific X, Y, Z Location?
To probe a specific X, Y, Z location, you can type in the desired location on the Point Probe panel, and press the Calculate button. At this point, the values for the desired point will be displayed on the Point Probe panel. If the point you selected is not contained within the computational grid, an error message will be displayed.

How do I see the location of the point I have probed?
When a point is probed interactively, the Display Point option on the Point Probe panel is on by default. This will leave a marker in the window until Display Point is turned off or the panel is exited.

What is the difference between the Point Probe and the Surface Probe on the Computation Surface panel?
The Surface Probe, on the Computational Surface panel, is used to probe specific I, J, and K coordinates. As a result, it is only available for structured grids, and will only return the scalar value at a specific I, J, K location. The Point Probe lets you probe anywhere in a structured or unstructured dataset, and will return up to three scalar values and one vector value at the probe location.

How can I create a file based on the point probe output?

If the Output To File option is turned on, you will be prompted for the name of an output file for the results of the Point Probe. As long as the Output to File option remains on, the results of all subsequent point probes will be written to the specified file. The output will consist of the X, Y, Z coordinates, and up to 3 scalar and 1 vector function as shown on the panel itself. This file may be read into the 2D Plot Controls panel using the Import option. In that case, the variable that was in the Scalar register at the time of the probe will be the vertical axis of the plot (see **Chapter 12**). The exact format of this file is shown in **Appendix G** of the **Reference Manual**. Output to File is forced off and the file is also closed if the panel is exited.

If I already know the X, Y, Z coordinates, can I use them as input to the point probe?

Using the Point Probe Input option, a file of X, Y, Z coordinates may be input. **FieldView** will then output these X, Y, Z coordinates as well as the currently loaded registers at these locations. There are **3 input formats,** textual (.txt), comma separated values (.csv) and MAT file format (.mat), where the latter two are also used for exporting FieldView visualization objects, as described in **Appendix J** and **Appendix K** of the **Reference Manual**. The MAT file format is most computationally efficient. Note that the textual format is a simple list of the three coordinate values per line, with no commas, as shown in the following figure. The output file will contain the input values, followed by scalars/vectors which are returned for each value currently loaded in the *function registers*. Your Input file can be entered either with the *File > Open Restart* pull-down menu on the Main Menu, or by using ***Tools.. Point Query...***, see **Chapter 14** of **Working with FieldView.**

0.000	0.000	0.000
1.123	0.000	2.345
3.000	8.7	0.2

X Coordinates   Y Coordinates   Z Coordinates

Figure 228 Point Probe Sample Input File

Variable List (all on one line)

VARIABLES = "X", "Y", "Z", "V-VELOCITY", "STATIC PRESSURE", "U-VELOCITY",
"V-VELOCITY", "W-VELOCITY", "DENSITY"
ZONE J = 1, T = "Point Probe"

0.000	0.000	0.000	0.123	2.133	-3.3219	0.123	1.3294	14.4327
1.123	0.000	2.345	1.175	1.573	-3.12342	1.175	1.8374	10.3878
3.000	8.7	0.2	3.811	6.2318	1.32088	3.811	0.93284	5.03238

Values of Vector Register

Zone and Title Information   Value of Scalar Register   Value of Threshold Register   Value of Iso-Surface Register

Figure 229 Point Probe Sample *Textual* Output File

Cylindrical Note: When cylindrical coordinates are specified using an FVREG file (Region definition), the Point Probe input values are assumed to be in RTZ (Radius, Theta, Z) coordinates. See **Chapter 3** for more information.

Possible Problems

*Why does **FieldView** "beep" when I try to probe?*
When using the interactive probe, you must click (CTRL-M1) on an object in the graphics window that belongs to the current dataset. If you click on empty space or an invalid object (such as a legend or title), a "beep" is sounded and the point is not probed.

Why do I sometimes I see report NO FUNCTIONS DEFINED *as I interactively probe my data using the Point Probe Panel?*
The values returned by your interactive probes are those stored on the visualization object anytime after displaying those scalar/vector quantities on the **surface**, otherwise FieldView will report NO FUNCTIONS DEFINED. However, FieldView *will* return **volume** quantities as set on the Function Specification Panel if pressing the Calculate button. To help avoid confusion, the names of the variables are displayed as well as their values

Using the Point Probe panel to return Surface vs. Volume data.

Interactive Point probing allows you to return values stored on the visualization object rather than from the volume, making it possible for FieldView to return absolute information for [BNDRY] variables, important for *Boundary Only* data (without volume), including XDB exports which have been read back into FieldView. Probing interactively can only return functions which have been stored on the visualization object. If you have just created an object and attempt to probe on it, the Point Probe panel will appear as shown at left in **Figure 230**. In this case, the probe will only return information on the grid and the XYZ location. If you have applied functions to that object, or have pressed Calculate (to query *volume* variables), you will be presented with image at right.
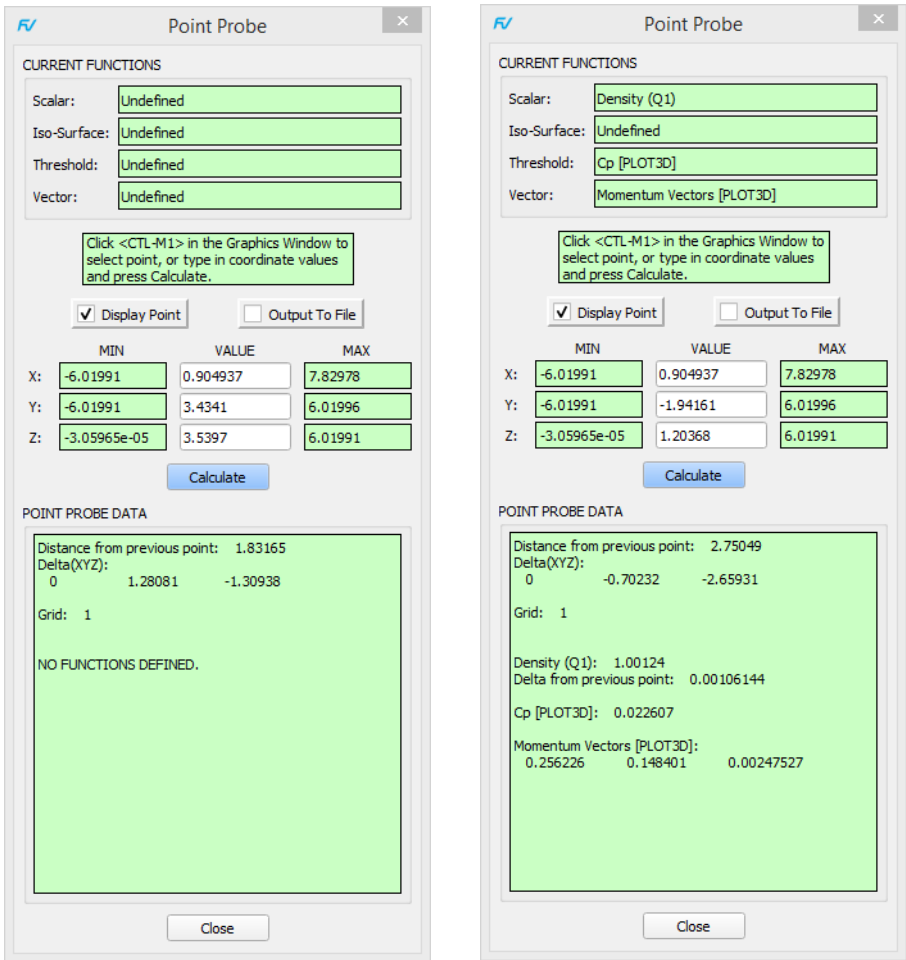
Figure 230 Interactive Point Probe result for a Visualization Object *prior* to applying functions (at left), and *after* applying functions to that object (at right.)

If a surface has a scalar (or threshold or vector) function defined prior to being probed, the result for that function will be displayed in the point probe panel. Iso surface functions are only returned by inter-

active probing if the surface is an iso surface. The probe will return the exact (constant) value of the iso function on that surface (which will be the same anywhere on the surface).

Note: If you use manually entered coordinate values and press the Calculate button, FieldView will return values for functions stored in the volume register (which is how earlier versions of FieldView behaved interactively, as well). If you want to the Point Probe Panel to report *only* functions as returned from the volume, rather than the Visualization Object, matching what was done in older Field-View releases, set the environment variable FV_VOLUME_PROBE (to any value) prior to launching FieldView, or use *Tools.. Point Query.*

Additional Point Probe features of note with current FieldView versions

Point probing on a streamline returns Duration for the location you picked. It also returns the value of the scalar function that was in the scalar register when the streamlines were calculated. Point probing on particle paths returns Duration or Time if they are present. It also returns Emission Time if there is a particle path variable named Time (unless Emission Time is already a path variable in the file). Point probing on particle paths also returns a scalar, but only if the particle paths are scalar-colored. The scalar can be a "path" scalar (from the file), or a volume scalar (from the scalar register).

Output to the interactive Point Probe has been modified to display the distance from the previous point probed, the differences in the X, Y and Z coordinates (or either the R, T, Z or R,T, X coordinates depending on your reference frame), and the difference in the scalar function. Differences can be reported across time steps as well.

Chapter 14

Tools

14

The Tools menu (**Figure 231**) contains features which allow you to Export surface and object information, Unify the display characteristics of all surfaces of a given type, create **FieldView** Structured Boundary Files based on the grid file read in, define custom colors with the Color Mixer, and use preset graphics window sizes for images and animations.
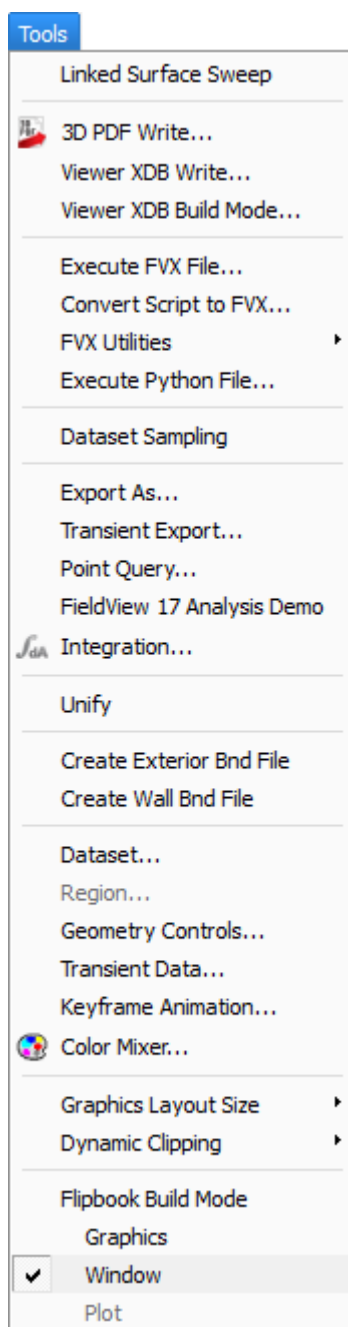
Figure 231 Tools Pull-Down Menu

Linked Surface Sweep

This feature makes the task of sweeping many surfaces at the same time easier. An additional benefit is that animations of multiple swept surfaces can be created with a single mouse button click. Sweep actions of a current coordinate surface will affect all other coordinate surfaces, including all datasets in a **FieldView** session. Similarly, sweep actions of a current iso-surface will affect all other iso-surfaces, and sweep actions of a current computational surface will affect all other comp surfaces.

When the Linked Surface mode is turned on from the Tools menu, control of all like surfaces is from the Sweep Controls section of the current surface:

Clicking Step Down will set the sweep direction to Down, broadcast this to all other surfaces of the same type, and advance all surfaces by one step.

Clicking Bounce will broadcast this to all other surfaces of the same type, and advance all surfaces by one step in the current sweep direction.

Clicking Step Up will set the sweep direction to Up, broadcast this to all other surfaces of the same type, and advance all surfaces by one step.

Picking the Sweep toggle will continually broadcast this to all other surfaces of the current type, sweeping all at the same time.

COORD PLANE: ● X ○ Y ○ Z
Min: Current: Max:
-6.01991 0.904937 7.82978
☐ Sweep ◀ ◀▶ ▶ Steps: 25 +/−

The range over which a surface is swept can also be set independently for each surface. This setting along with the Steps specified will determine the size of the step that a surface will take.

The number of steps can be independently set for each surface.

Figure 232 Explanation of Sweep Control for Linked Surface mode

Changing any other values apart from the sweep controls noted above, for the current surface, will not affect other surfaces. In the linked surface mode, the current surface will control the sweeping characteristics for all other like surfaces. So, if you change to a new current surface, and hit the sweep button your surfaces will be moved on the basis of the new current surface panel settings.

If you turn Flipbook Controls ON while the Linked Surface Sweep mode is ON, then you will be able to directly create a flipbook animation by pressing the sweep button on the current surface.

To disable this mode, you must explicitly turn it off from the Tools pulldown menu.

The state of Linked Surface Sweep (either ON or OFF) will be saved and restored by data file restarts. Restarts saved pre-dating the Linked Surface Sweep capability will force Linked Surface Sweep off.

A script command permits the automated operation of turning this mode on or off. The command is:

```
LINKED_SURFACE_SWEEP ON/OFF
```

Note that when using the Keyframe Animation Control Panel, this mode will be completely ignored. This will permit you to make a Keyframe Animation consistent with previous versions of **FieldView**, and your Keyframe Animation Restarts will work as they have previously. The Keyframe Animation panel will leave the state of the Linked Surface Sweep mode untouched.
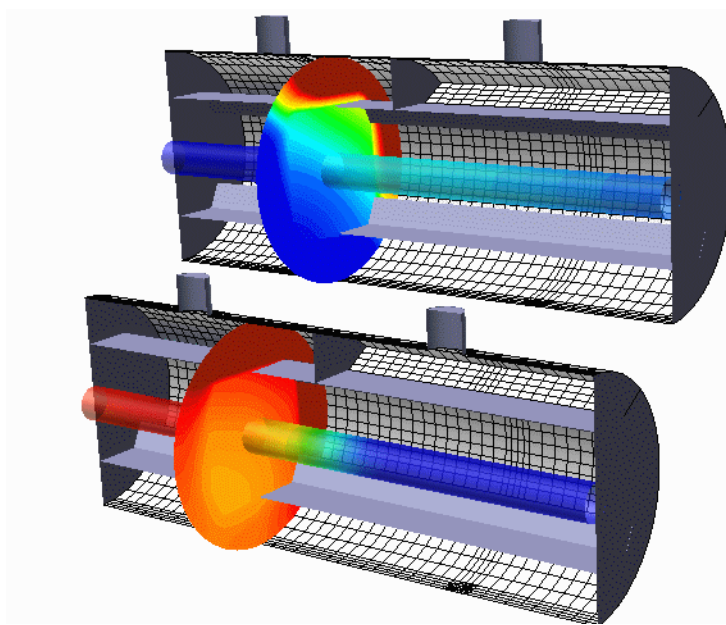
Four coordinate surfaces, two in each dataset, controlled by one R coordinate surface (see Heat Exchanger Tutorial Datasets).
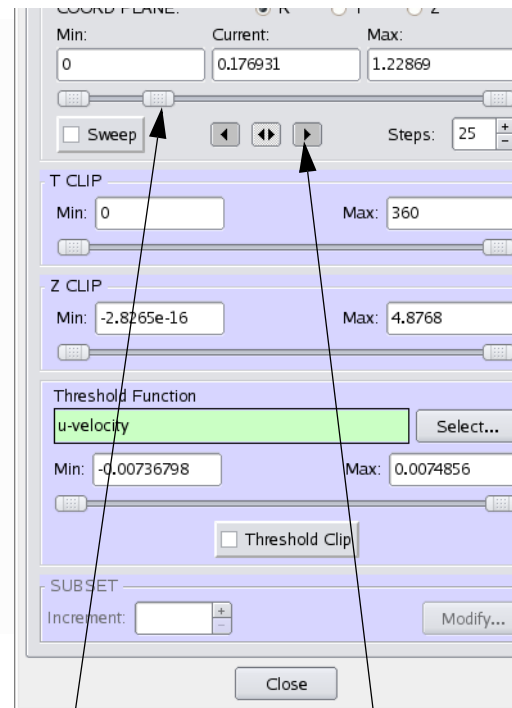
Starting point for sweeping all surfaces.

Single step advance to sweep surfaces

Figure 233 Starting point for Linked Surface example
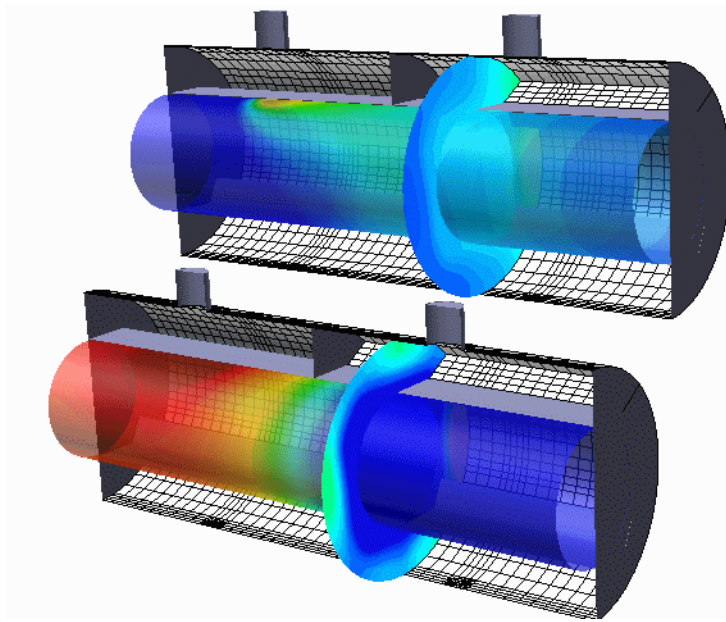
Finishing position of all four surfaces, controlled by R coordinate surface with Linked Surface Sweep ON (see Heat Exchanger Tutorial Datasets).
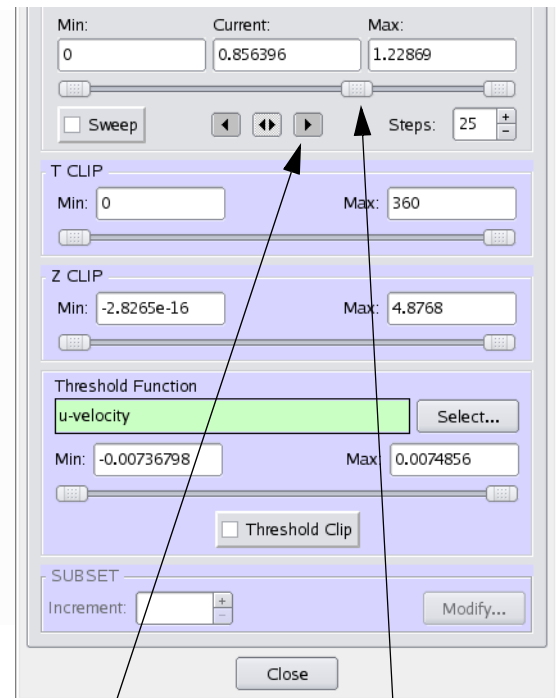
Single step advance to sweep surfaces

Finishing point for sweeping all surfaces.

Figure 234 Finishing point for Linked Surface example

Linked Surface Sweep with Multi-Windows

It is possible to sweep multiple surfaces of the same type (computational, coordinate, or iso-surface) in one or more datasets simultaneously by turning linked surface sweeping on via Tools → Linked Surface Sweep.

By default, Tools → Linked Surface Sweep is off. When the Sweep button on the current surface panel is pressed, only that surface sweeps. This is the behavior for **FieldView** for a single window.

It is also possible to perform linked surface sweeping across multiple windows. Control for this operation is accessible from the link icon, third from left on the window toolbar. By default, the "linked" sweep state is on (icon colored green) and can be toggled off (icon colored gray) by clicking directly on the icon itself.

The default operation for linked surface sweeping is shown at left in **Figure 235**. To sweep all surfaces in all windows, enable Linked Surface Sweep from the Tools pulldown menu, select any one of the surfaces in either window and hit the sweep button.
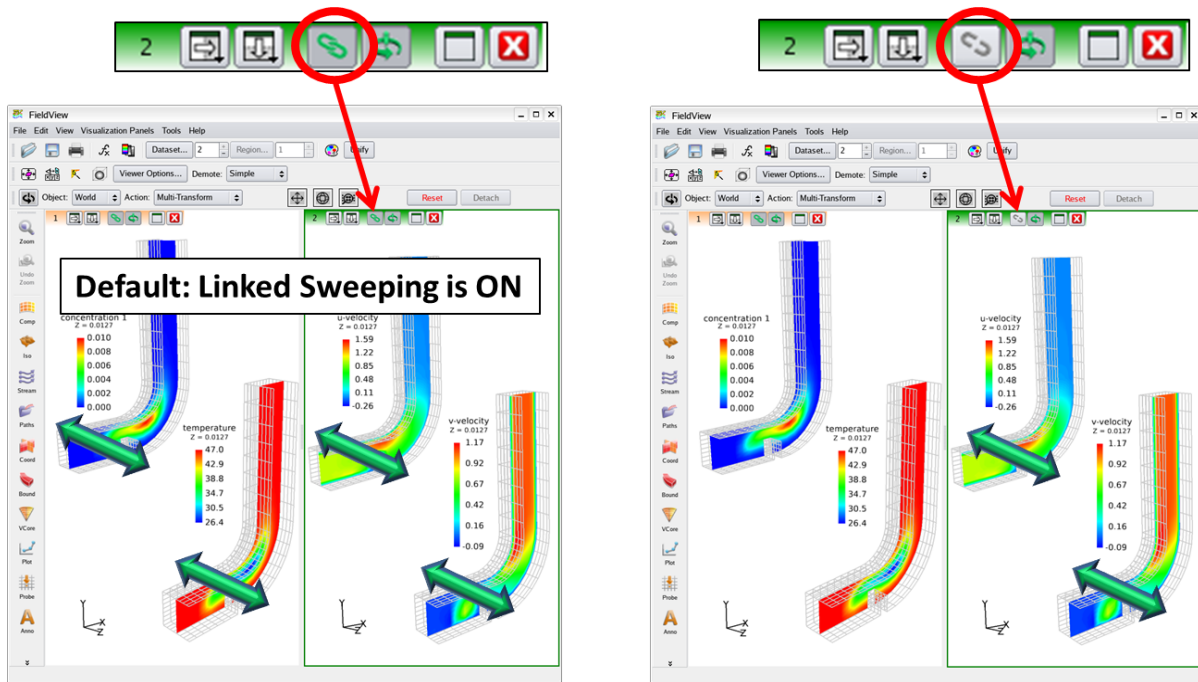
Figure 235 Linked Sweep Behavior

To simultaneously sweep the surfaces in just window #2, disable the linked sweep setting in the window toolbar by pressing the link icon. With Linked Surface Sweep set from the Tools pulldown menu, select any one of the surfaces in window #2 and hit the Sweep button for that surface to sweep only the surfaces in window #2. With window #1 and window #2 unlinked, you can sweep just the surfaces in window #1 in the same way.

To sweep only one of the surfaces, turn Linked Surface Sweep off from the Tools pulldown menu.

3D PDF Write

FieldView is able to export 3D PDF format files. Such files are viewable with Adobe Acrobat Reader versions 10, 11 and DC on Windows systems, and with various reader applications on mobile devices.

The current window will be exported to a 3D PDF format file. The GUI to manage the export process shown in **Figure 236** is invoked from the Tools menu entry "3D PDF Write..." (see **Figure 231**) or alternatively from the 3D PDF icon on the Main Toolbar (see **Figure 21**).
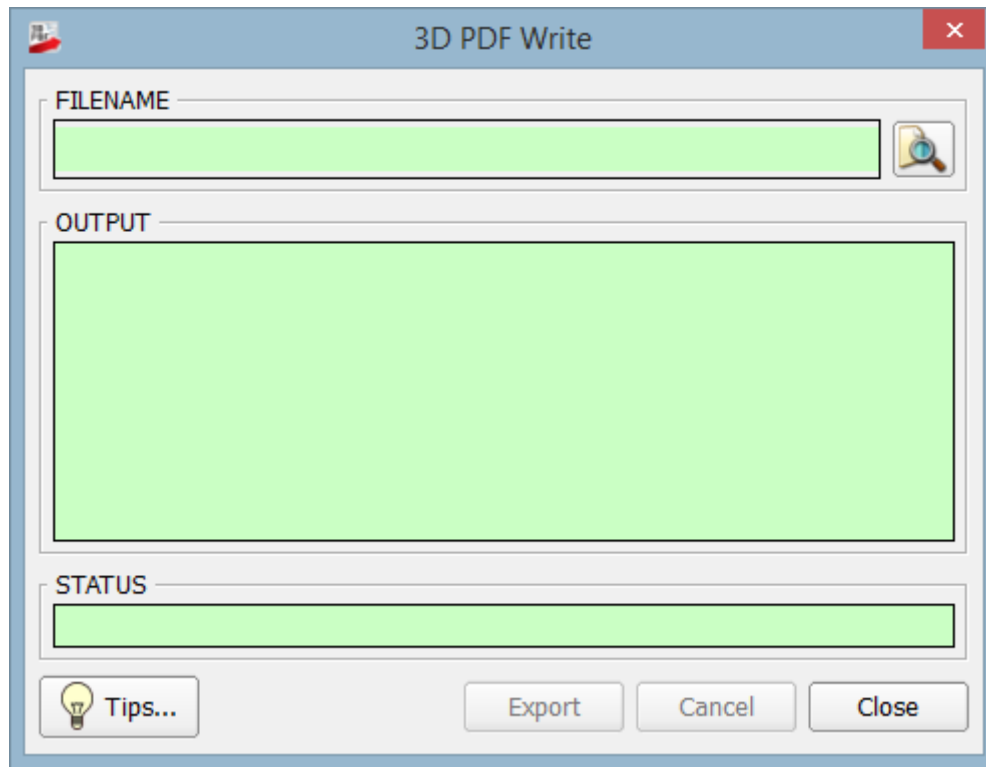
Figure 236 3D PDF Write GUI

All fields in the 3D PDF Write GUI are read only. To specify a file name for the export, click the icon to the right of the FILENAME field to bring up the Save Window to 3D PDF file selector shown in **Figure 237**.

When file name is entered and Save is clicked, an exporter process is launched; if the desired file name is already set in the GUI, the Export button will launch the exporter process. While data is being exported the Export button will be grayed out to block further exports. The OUTPUT window verifies the progress and the STATUS field reports the success of the 3D PDF conversion. A Cancel button is provided to interrupt a pending 3D PDF export, which is useful when such an export is exceeding time or memory constraints. When data has been provided to the exporter process, the Export button returns to normal and standard **FieldView** operations may proceed as usual. The Tips... button will open a window with additional information (see **Figure 240**). The Close button will close the panel. Note that closing the panel while the export is in progress allows you to resume your work in **Field-View**. If you'd like to see the export status again, just bring back the 3D PDF Write panel.
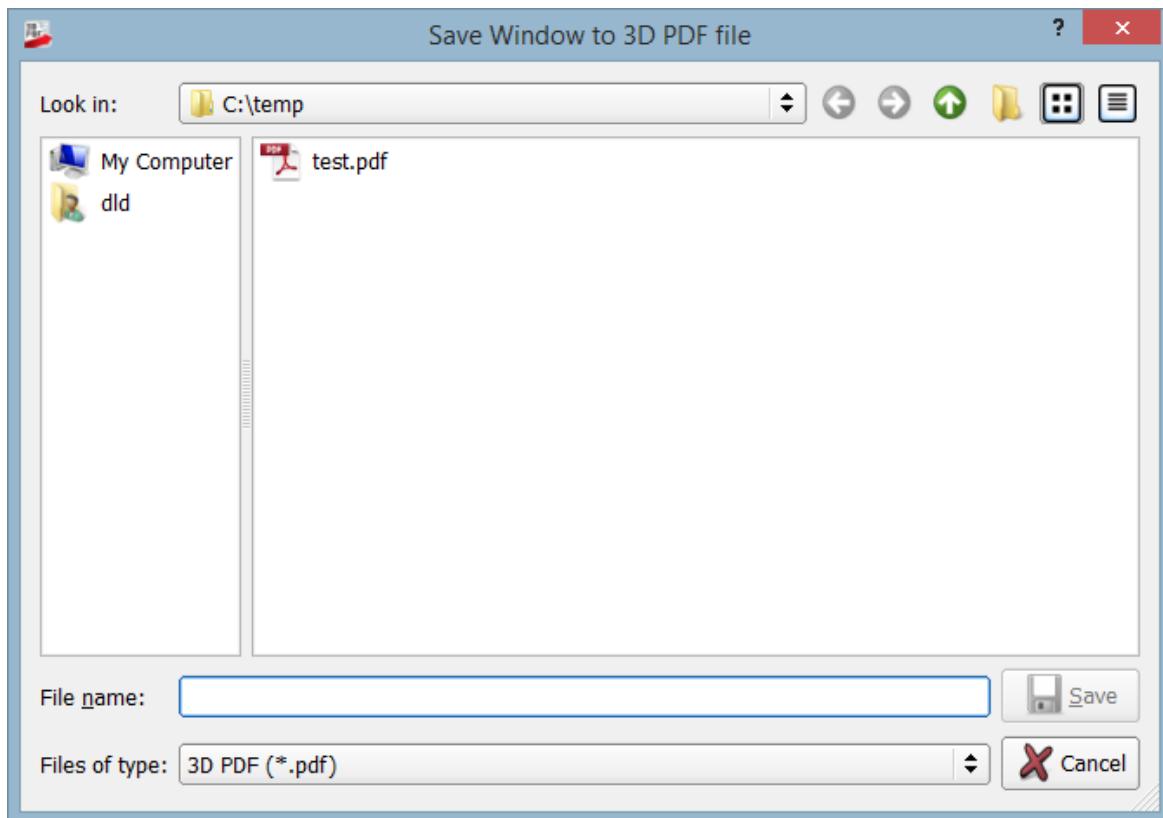
Figure 237 Save Window to 3D PDF file Selector

The resulting file contains 3D geometry which may be variously viewed and rendering altered with the provided controls, in 3D PDF viewers. All surfaces and rakes and geometries for all datasets that appear in the current window are exported to the 3D PDF file with several small exceptions. **Field-View** annotation titles, arrows and legends are exported as data to be rendered on top of 3D objects, for high quality readability. Dataset outlines and axes markers are not exported as the 3D PDF viewer will have its own version of these.

A complete restart with the form:

```
<output file base name>_pdf.<extension>
```

is written for each 3D PDF file export. (See the XDB export doc for a similar feature.) This will allow you to easily retrieve the state of your **FieldView** window, at the time of the export.

A script command `3DPDF_WRITE filename` provides 3D PDF file creation from **FieldView**'s scripting language. This script command can be called from **FVX** or Python by using the `fv_script()` function:

```
fv_script("3DPDF_WRITE filename")
```

Scripted exports must wait until the file export completes or fails before continuing to process more script commands.

The 3D PDF export will generate a document using the fixed A4 landscape format. (A4 is the standard European paper size.) As a consequence, objects may appear in a different position as in your **FieldView** graphics window, if it uses a different aspect ratio.
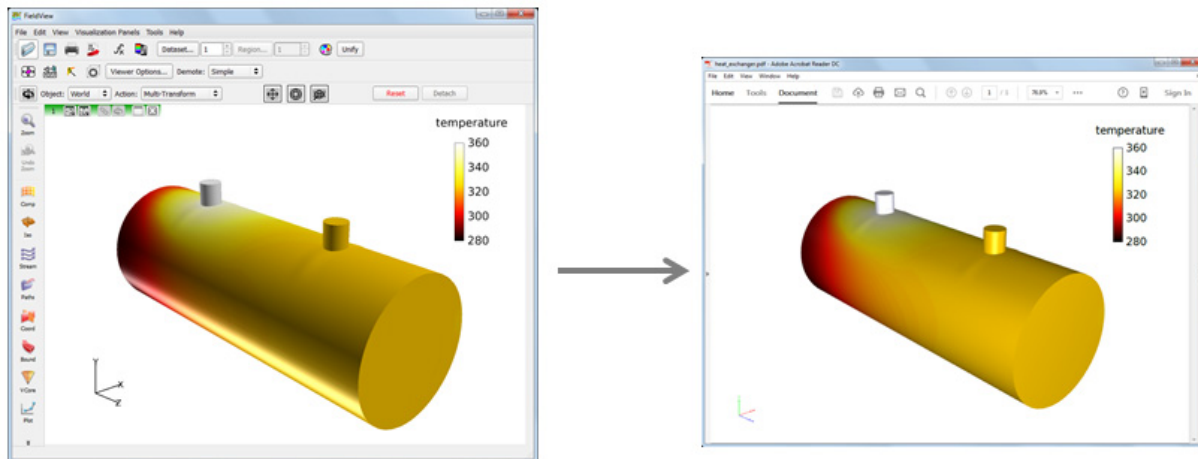
Figure 238 With a matching aspect ratio, FieldView objects fall into place in the PDF document exported to A4 Landscape format.
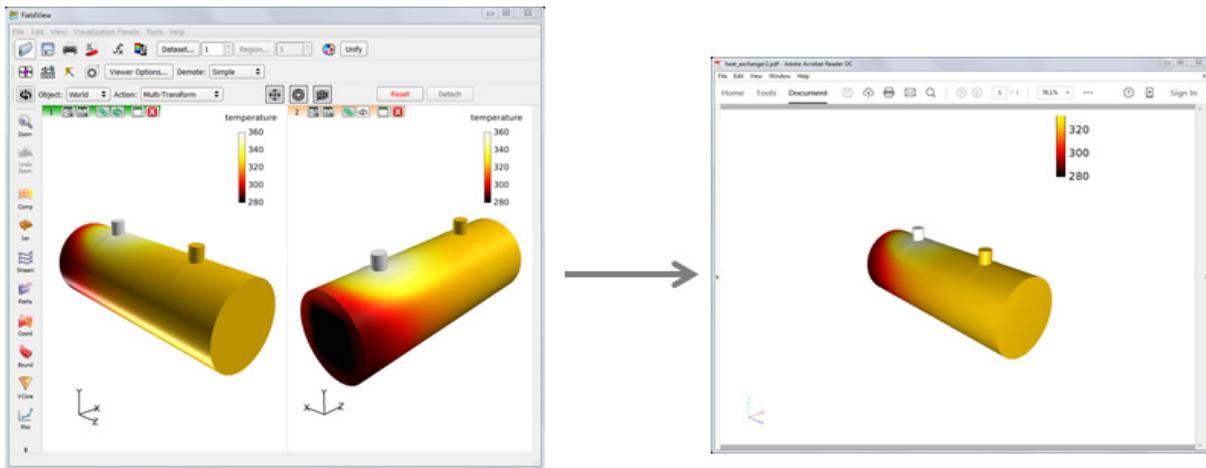
Figure 239 With a big difference in the aspect ratio of the FieldView current window, which is only half of the full graphics window in this case, FieldView objects can be misplaced or even out of frame in the exported PDF document.

To avoid this problem, we recommend that you resize your current **FieldView** window to an aspect ratio that matches or is close to the one of the A4 landscape format (1.414). For instance, among the layouts available from the Tools Menu > Graphics Layout Size, the 720x486 option will be very close (with an aspect ratio of 1.481). It is also possible to resize the graphics window manually by grabbing the bottom right corner of the **FieldView** main panel. Finally, you can accurately define the size of the graphics window in pixels with:

- The command line option: `-size=mmmxnnn`, where `mmm` and `nnn` are the width and height in pixels.

- The SCRIPT command: `SIZE width height`

Important Notes and Limitations

There are a number of known issues with the 3D PDF format and with PDF readers.

FieldView must transmit scene data to the 3D PDF file conversion process. If there is a large amount of data this may take some time during which no other **FieldView** operations are possible. Once this data has been transmitted by an interactive **FieldView**, the export process may continue to run and other **FieldView** operations are permitted, but the 'Export' button will be disabled (grayed out) until the 3D PDF file export completes. A scripted **FieldView** 3D PDF file export must wait for the export to complete advancing in the script. Only a single 3D PDF file export process may be run at any given time.

The implementation of transparency in 3D PDF viewers is very rudimentary and the quality is low.

3D PDF format files are viewable with Adobe Acrobat Reader versions 10, 11 and DC on Windows systems, and with various reader applications on mobile devices. There is no version of Adobe Reader to support 3D PDF files on Linux systems.

For more information on 3D PDF readers' limitations, please refer to the Tips... window, available from the 3D PDF Write panel. This window is reproduced in the figure below.
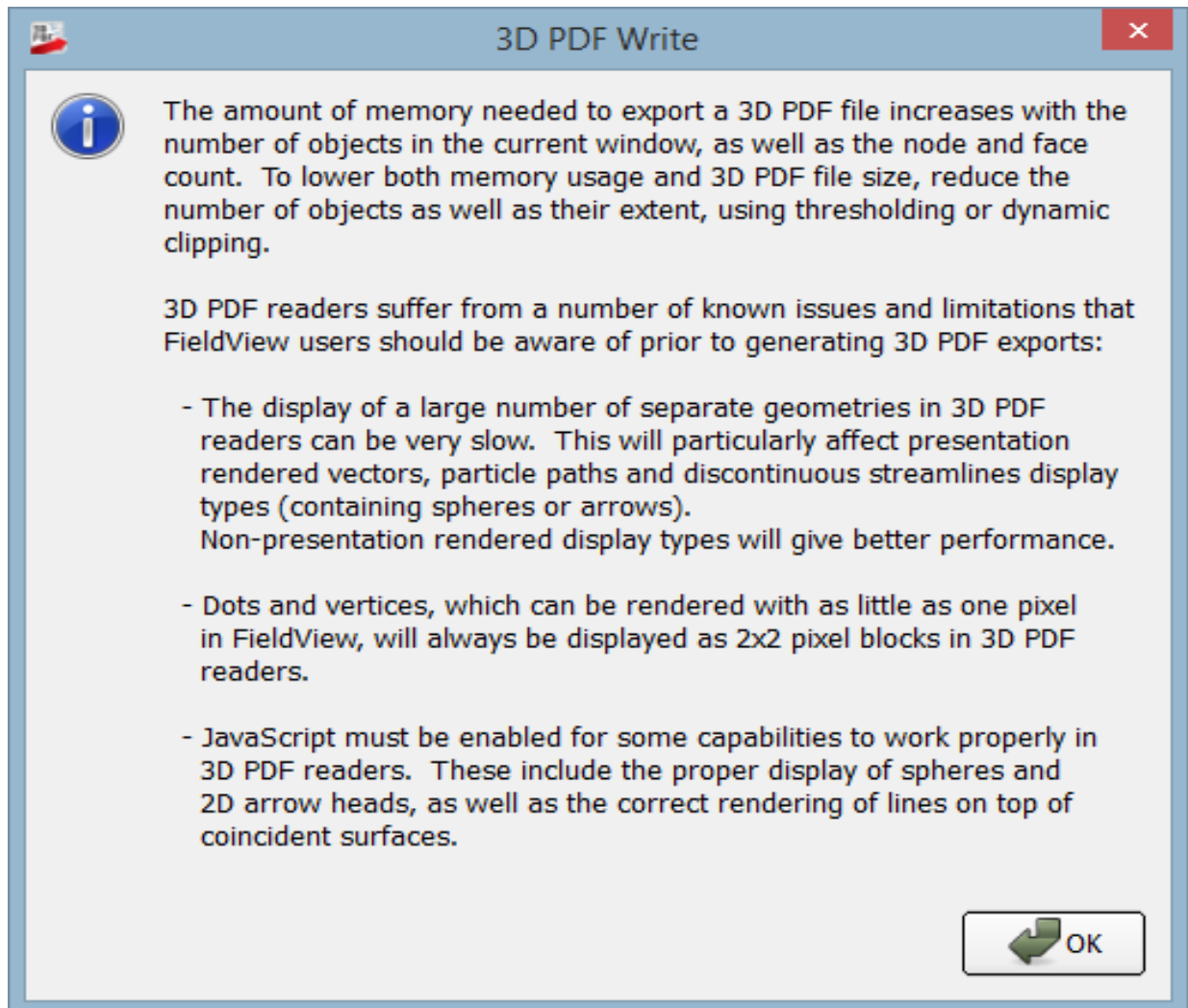
Figure 240 3D PDF Write Tips...

Writing XDB Files

The extraction process used to prepare data for import to **FieldView** or viewing with **XDBview** involves four steps. Step 4 is further detailed below.

1. Load a dataset into **FieldView**.

2. Create any surfaces (computational, coordinate, iso- and boundary) and/or rakes (streamlines, streaklines and particle paths) to be included in the XDB file extract database.

3. Unless you prefer to interactively select functions to be saved when exporting the XDB, prepare the `xdb_vars` file to include the names of all variables that will be written to the extract database file.

4. Create the extract database file by writing the viewer XDB file in either the normal mode or the build mode.

Viewer XDB Write

The first mode is a typical Viewer XDB export, which adds every graphical object that is created and presently visible in the graphics window to the extract file. To write a Viewer XDB file in this manner, click on the Tools pull-down menu from the Main Toolbar and select Viewer XDB Write.
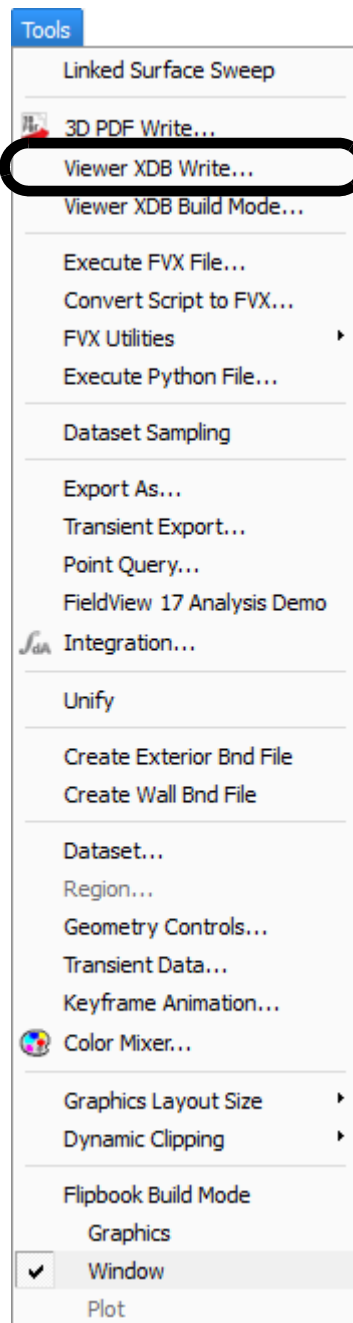
Figure 241 Viewer XDB Write

When prompted, choose the directory in which the extract file will be saved, name the extract file, and press OK. A panel should now be on the screen providing space to insert a title and comments. When finished, click OK to write the extract file.

Viewer XDB Build Mode

The second mode allows the user to include a surface or transient sweep in the extract file. Bring up the surface or transient data panel with the controls of the sweep that is to be included in the extract file. From the Tools pull-down menu, select Viewer XDB Build Mode.
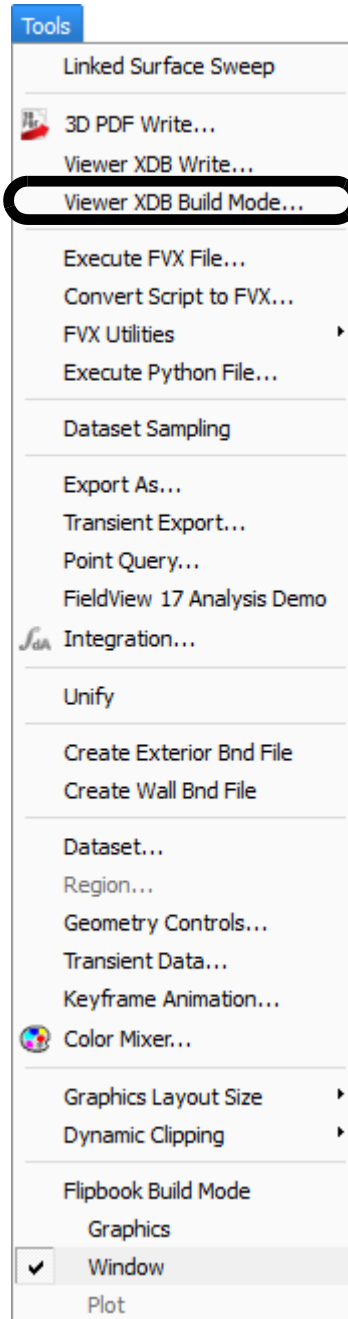
Figure 242 Viewer XDB Build Mode

Similar to the previous case, save the extract file and insert a title and comments, if desired. After pressing OK on both of these panels, note that the Sweep button on the surface/dataset panel has

changed to Xbuild. Pressing Xbuild will temporarily turn off graphic updating, sweep the surface/time steps, write the extract database file, and restore the graphics window.

Specifying the scalars and vectors for the XDB file

Interactive Function Selection for XDB Export

To simplify the task of selecting the functions to be saved when exporting an XDB, a GUI is provided, as shown in **Figure 243**.
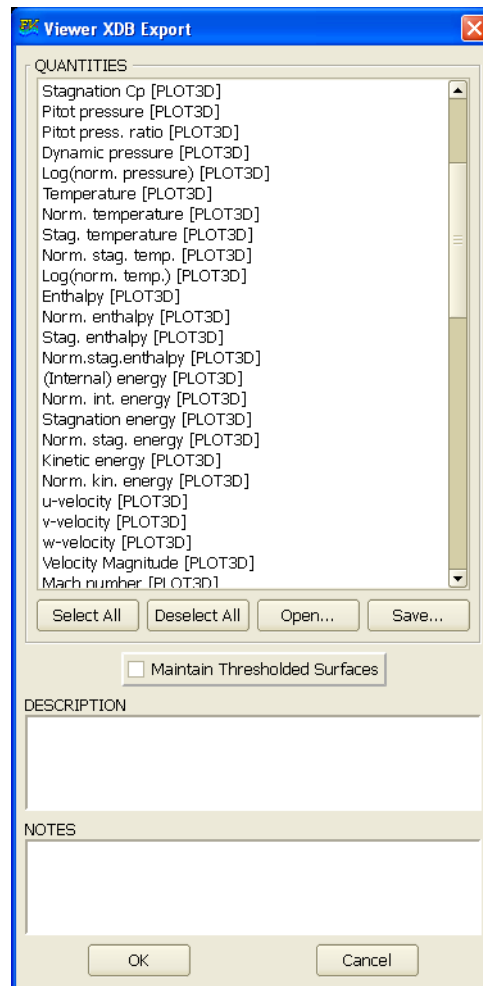
Figure 243 Viewer XDB Export Panel

The QUANTITIES section of the Viewer XDB Export panel in Viewer XDB Write and Viewer XDB Build modes allows interactive function selection for XDB export. It lists all scalar and vector functions and formulas valid for the current dataset, excluding geometric functions (such as "X", "Y", "Theta" and "Rcyl"). Scalar functions are listed first, followed by vector functions; scalar and vector functions are further broken up by categories.

To simplify function selection when exporting an XDB file, functions in the QUANTITIES section of this GUI are pre-selected according to the following rules:

• They have been used for a visualization object whose visibility is ON. In other words, if a function is in active use on any surface or rake for the current dataset, it will be pre-selected for XDB export.
• The function is included in the `xdb_vars` file.

For certain visualization objects, functions in use are not pre-selected:

• Scalar functions in use by 2D Plots are not pre-selected.
• Vector functions that are only used for streamline or vortex core calculations are not pre-selected.

There are four buttons located immediately beneath the QUANTITIES section of this GUI. The Select All/Deselect All buttons allow you to change your current selected functions, selecting either all or none, respectively.   If you choose the Select All button, you will see the following XDB Write Warning message. In general, the intent for XDB exports is that fewer functions are saved, containing only the functions of immediate interest.

Figure 244 XDB Write Warning

Pressing the Open... button lets you browse to and open a previously created `.xfn` file. When this is done, any functions in the QUANTITIES list matching those in the `.xfn` file are pre-selected. This action will de-select anything that was already selected.

Pressing the Save... button opens a browser and lets you save a `.xfn` file, independently from the XDB export. This saved `.xfn` file can be re-used in a later session.

Functions can be selected interactively, overriding the pre-selection rules. Selecting a function by left-clicking on it will select only that function and clear any pre-existing selections. To select more than one function, hold the [Ctrl] key down and left-click on the desired functions. To select a range, left-click on the function at the top of the range, hold the [Shift] key and left-click on the function at the bottom of the range. You can also select a range of functions by left-clicking and dragging.

The Viewer XDB Export panel (**Figure 243**) has a toggle button to "Maintain Thresholded Surfaces" in Viewer XDB Write and Viewer XDB Build modes. When enabled, only the surface geometry visible

after threshold clipping is output, potentially reducing file size substantially. This option can make extracts even smaller, allowing faster transfer and read operations for these files. It also reduces the memory required for loading them into **FieldView** or **XDBview**, pushing further the benefits of XDBs.

Thresholding includes:
- clipping by the selected threshold function, and
- min/max coordinate axis limiting (coordinate surfaces only).

Thresholded surfaces are exported with the tag `[THR]` embedded in the boundary surface name and include the following types:
- Coordinate surfaces
- Iso-surfaces
- Boundary surfaces (unstructured datasets only)

Thresholding by the selected threshold function will be disregarded for computational surfaces or boundary surfaces of structured datasets when exported. However, they will continue to always clip by IJK coordinate axis limiting. If either surface types are present, visible and thresholded, complete unthresholded surfaces are exported and the following warning is issued:

XDB Write Warning: Computational or Boundary Surfaces
of structured datasets have been detected and will not
maintain thresholding on export.

Dynamic clipping has no effect on the exported XDB files.

To export the XDB file, press the OK button at the bottom left corner of the panel. When you export an XDB file, a `.xfn` file with the same name as the XDB file is written. If a `.xfn` file with the same name already exists, you will see the following overwrite confirmation:

Figure 245 XDB Function List Overwrite Confirmation

In this way, you can choose to preserve or overwrite your existing `.xfn` file. After a selection has been made here, the XDB is exported.

If you choose not to export the XDB, you can push the Cancel button at the lower right corner of the Viewer XDB Export panel.

Important notes and limitations

Surface normals are always exported for all surfaces and can be used in subsequent calculations. The surface normal function, named "N", and its components (N.vector.x, N.vector.y, N.vector.z) are reserved names.

The limit on the functions exported to an XDB is 100. If you attempt to export more than 100 functions, you will be presented with a console warning advising you that you have exceeded this limit. At this point, an XDB file will be exported containing only the first 100 of all the functions selected.

XDB exports saved via SCRIPT commands will use either the `xdb_vars` or the `.xfn` method to specify the functions to export. Note that Boundary variables [BNDRY] are not exported to XDBs for Structured datasets.

Preparing the `xdb_vars` *File*

To specify the scalars and vectors that will be written to the XDB file, there are two options. The first is to create a file with the special extension `.xfn`, placing it in the same location where you will be saving the XDB file. When saving an XDB file, **FieldView** will first look for this `.xfn` file by matching it to the rootname for your XDB file. For example, if the name of the XDB file is `my_boundaries.xdb`, you would place the names of the scalars and vectors, written one per line, in the file called `my_boundaries.xfn`. The second option is to create a file with the specific name `xdb_vars`. In either the `.xfn` or `xdb_vars` file, the names of each of the scalars and vectors are written one per line. Comments can be added to this file by starting the line with an exclamation mark (`!`).

Creating an `.xfn` file offers the benefit of allowing you to associate the scalars and vectors in your XDB file with the XDB file itself and the restart files that get created when the XDB file is written. There is some flexibility in where you place either the `.xfn` or `xdb_vars` file. The search path that **FieldView** uses is to check in:

- the location where you are writing your XDB file

- "My Documents", for WINDOWS Users only

- `$HOME` (your home directory)

- `$FV_HOME/data` (the **FieldView** installation `/data` directory)

An example of the `xdb_vars` viewer extract file is as follows:

```
! xdb_vars example for PLOT3D Turbomachinery Case
!
Density (Q1)
u-velocity [PLOT3D]
v-velocity [PLOT3D]
w-velocity [PLOT3D]
```

The variables listed within the `xdb_vars` file do not need to include all of those supported by the solver. Note that the more variables added to the `xdb_vars` file, the larger the subsequent extract file will be.

There are two modes **FieldView** uses to create extract databases. Each mode will create an extract database file with the extension `.xdb`. A complete restart is automatically saved based on the root filename.

Creating an extract database using either method provides the user with the opportunity to title the viewer export as well as provide comments, both of which can be displayed later in the **FieldView** Advanced Technology Viewer.

It is possible to read an XDB file (steady or transient) and treat it in the same way as any dataset loaded into **FieldView**. This offers functionality well beyond what is available with our **FieldView** Advanced Technology Viewer. Please refer to **Chapter 1** for additional details.

XDB Format

It is possible to write face-based results to XDB files for unstructured datasets only. This further enables quantitative postprocessing via XDB workflows - integrations performed on surfaces with face-based results in XDB files will match those reported by the finite-volume based CFD solver codes.

For transient cases, XDBs are saved using one file per time step, offering a 5X to 7X improvement to read a transient XDB export. Exported filenames contain the time step number embedded in the base name, as specified by the user, according to the following rules:

- The `.xdb` extension (case insensitive) will be automatically appended to all XDB exports. If you have already specified the `.xdb` extension in the base name, it will not be redundantly added.
- The time step number matches the integer for the TIME STEP as shown in the Transient Data Controls panel for the current dataset.
- The time step number will be inserted in front of the first `"."` in the non-directory part of the specified filename. If the base name is specified as `/opt/data/my.transient.export.xdb`, then the name for the XDB for time step "10" will be `/opt/data/my10.transient.export.xdb`.
- If the last character before the `"."` is a digit, we will insert a `"_"` in between this digit and the time step number. For example, if your specified XDB filename is `F18.xdb`, the name of the XDB for time step "12" will be `F18_12.xdb`.

When you specify an XDB filename for a transient case, either interactively or from a SCRIPT containing the `XDB_ENABLE xdb_file_name` and `SWEEP TIME` commands, you will see the following console printout:

```
XDB Extract data will be exported to xdb_file_name

XDB Extract filename is modified to include the time step number.
```

A complete restart is always written when an XDB is exported to allow the easy re-creation of the original set of visualization objects stored in the XDB. For transient XDBs, this complete restart will only be written once using the base name specified for the XDB and will not have the time step number included.

Transient XDBs include the solver solution time, enabling time synchronization between XDB files and particle path data. The PARTICLE SET format is used for transient streakline data to take advantage of the substantial performance benefits for this feature. Due to these format implementations, **FieldView** releases prior to 14 are unable to read the newer XDB files.

If particle path data is imported into **FieldView** along with steady or transient volume data, it will not be redundantly stored within XDB exports.

It is still possible to create XDB files using the older XDB file format, which is compatible with the **FieldView** Advanced Technology Viewer. To do this, you must set this environment variable:

```
FV_USE_FV13_XDB_FORMAT
```

Script Support for Viewer XDB Files
There are two **FieldView** script commands that automate the generation of Viewer XDB Files. The first matches the Viewer XDB Write option from the Tools Menu GUI. It has the syntax:

```
XDB_WRITE xdb-filename [THRESHOLD/NOTHRESHOLD] [title-string [notes-filename]]
```

Here, `xdb-filename` refers to the output filename to be used. The title, which would be interactively entered in the Viewer XDB Notes, would be specified as a string value. The accompanying notes section can be included by indicating an additional file, with appropriate comments. A typical example would have **FieldView** call a Restart via a script command, and then subsequently, write out the Viewer XDB file as follows:

```
! A simple SCRIPT to create a Viewer XDB File
RESTART ALL my-full-scene
XDB_WRITE my-viewer-scene "Simple illustration" see-my-notes
```

In the case where you would like to sweep through a transient dataset, or simply sweep a coordinate surface for instance, the script command to automate this procedure changes slightly:

```
XDB_ENABLE xdb-filename [THRESHOLD/NOTHRESHOLD][title-string [notes-filename]]
```

This methodology can be applied to a keyframe animation as well, since this can also be swept. A few simple examples follow below:

```
! A simple SCRIPT to create a swept Viewer XDB File
```

```
RESTART ALL my-full-scene
XDB_ENABLE my-viewer-sweep "Sweep illustration" see-my-notes
SWEEP BOUNCE 1
```

In the script above, the `SWEEP` command will sweep the current surface, and would follow the GUI actions of first turning on the Viewer XDB Build Mode, and then pressing the Sweep button on a surface panel. To sweep a transient case, you could use something like:

```
! A simple SCRIPT to create a transient Viewer XDB File
RESTART ALL my-full-scene
XDB_ENABLE my-viewer-transient "Transient illustration" see-my-notes
SWEEP TIME 1
```

This script could be used to sweep through a transient dataset and write the Viewer XDB file at the same time. Please recall that these types of scripts can also be run in batch from the command line where you start **FieldView** by using the `-batch` switch. The `-s` switch specifies the name of the script that you will run. For more information on script commands, see **Chapter 5**.

Frequently Asked Questions

What happens if I forget a variable in my viewer XDB file?
You will need to edit the `xdb_vars` file and add the appropriate scalar and/or vector functions to the list of variables. You can then use the XDB restart to re-create the graphical scene of interest and re-write the Viewer XDB file.

What does the error message, "Problem writing extracts file" mean?
In this case, you have a scalar or vector defined in the `xdb_vars` file that does not match what is in the dataset. **FieldView** detects this problem, issues the error message, and stops the process of writing the Viewer XDB file.

What happens if I include a variable in either my `.xfn` *or* `xdb_vars` *file which is not present in my dataset?*
Having this variable specified will not cause your XDB save operation to fail; the variable will simply be ignored. For example, if your variable name is `missing_scalar`, you will see a message in your console window similar to what is shown below:

```
XDB Extract data will be exported to my_boundaries.xdb
Functions stored in XDB are listed in my_boundaries.xfn
Function "missing_scalar" does not exist on this dataset.
```

Known Limitations of Writing Viewer XDB Files
1. It is possible that the particle paths that are visible in **FieldView** may not be present in the viewer.

2. Particle path variables that are not present in the flow volume are not exported. The only exception to this is emission time, which is exported.

3. Sweeps (both time and parameter) always output in the sequential order that they are traversed in **FieldView** via Xbuild. This has implications in the Sweep GUI panel if you started your Xbuild at any position other than the minimum value of time or parameter.

4. User-defined color maps are not exported.

5. The initial view of your model in input will not match the view in **FieldView** that was current when you exported.

6. Boundary types are grouped into boundary surfaces for export. They cannot be separated later in the viewer.

7. There is no support for 2-D plot export.

8. Dynamic clipping is not supported in the export.

9. All dataset, region, and surface transforms are ignored on export.

10. Keyframe animation settings are ignored.

11. The light source position will not match that of **FieldView**.

XDBview

XDBview is a stand-alone program based on state-of-the-art graphics technology. The viewer provides a user-friendly environment capable of visualizing and manipulating data extracted by **FieldView**. **XDBview** permits high performance display of large data and transient cases. To get data into the viewer, it must be first extracted by **FieldView**.

Execute FVX File

The Execute **FVX** File action will read and execute a **FieldView** Extension language (**FVX**) program. When this menu option is selected, the user will be presented with a file selector so that an **FVX** file may be selected.

Upon filename selection, the program is read in and execution begins immediately. If there are any errors, the debugger output will display the details on the error. More detailed information may be found on **FVX** in **Chapter 4** of the **Reference Manual**.

Convert Script to FVX

The Convert Script to **FVX** tool will allow a **FieldView** script to be read-in and the corresponding **FieldView** Extension language (**FVX**) version to be created, in one step. When this menu option is selected, the user will be presented with a file selector so that a script (`.scr`) file may be selected. The con-

verted output file will be saved to the same directory as the input file. More detailed information may be found on **FVX** in **Chapter 4** of the **Reference Manual**.

FVX Utilities

Some **FVX** utilities are provided with **FieldView**. These utilities are placed in the `fvx_and_restarts` subdirectory of the standard installation. These **FVX** utilities are directly accessible from within the **FieldView** User interface from the Tools Menu (see **Figure 246**).
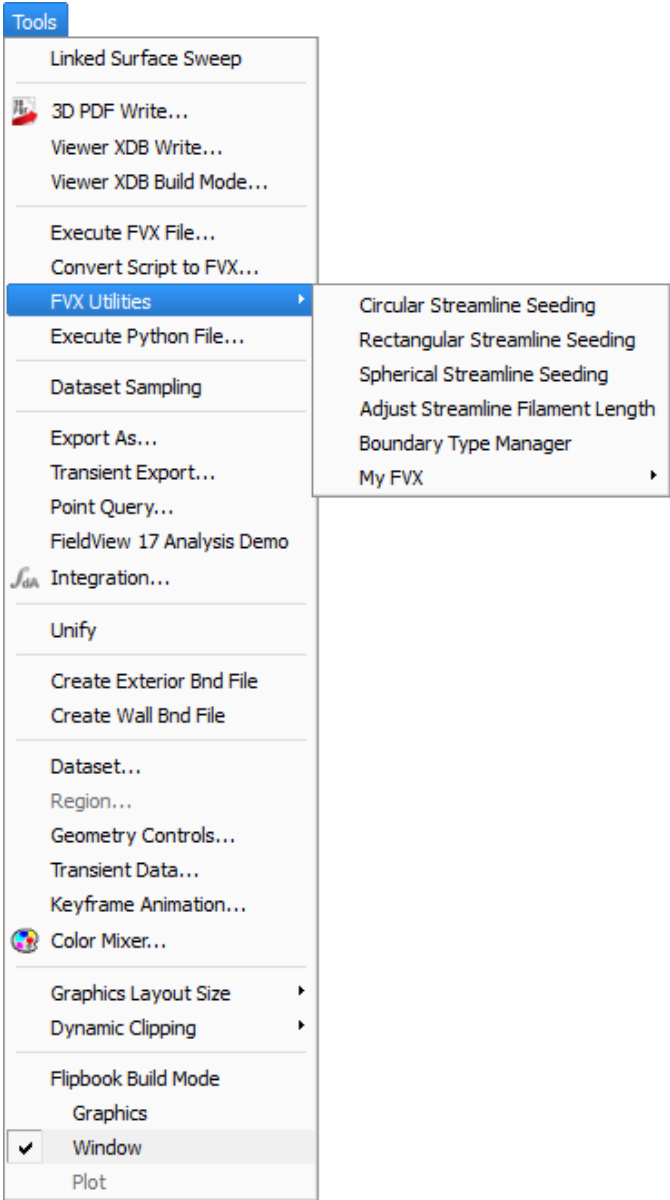
Figure 246 FVX Utilities provided with FieldView

Streamline Seeding Tools

Streamline Seeding Tools allow you to add circular, rectangular, and spherical seeding rakes to datasets. The three geometric seed distribution patterns utilize graphical user interfaces (GUIs) to provide quick, easy streamline seeding. Input options on the GUI panels determine the placement, orientation, size, and seed quantity of the seeding geometries and allow the user to create new rakes as well as delete old rakes. These **FVX** based tools are integrated into **FieldView** as part of the standard release. From the **Tools** menu, navigate to **FVX Utilities**. From the fly-out menu, you can choose one the three seeding tools, as illustrated in **Figure 246 FVX Utilities provided with FieldView**.

Detailed instructions on how to apply each of the seeding tools to a dataset are located in **Chapter 15** of the **FieldView User's Guide**. The chapter also includes useful example tutorials for each seeding geometry utilizing datasets from previous tutorials. It is suggested that first-time users work the basic tutorials before attempting the Geometric Streamline Seeding tutorials.

A complete discussion on the 'Adjust Streamline Filament Length' **FVX** program is provided here:

Boundary Type Manager

This **FVX** utility automates the creation of Boundary Surfaces for datasets with hundreds or thousands of Boundary Types. After specifying substrings, all Boundary Type Names are searched, and matching lists are generated using logical operators.

Note: This feature is currently not supported on the Mac.
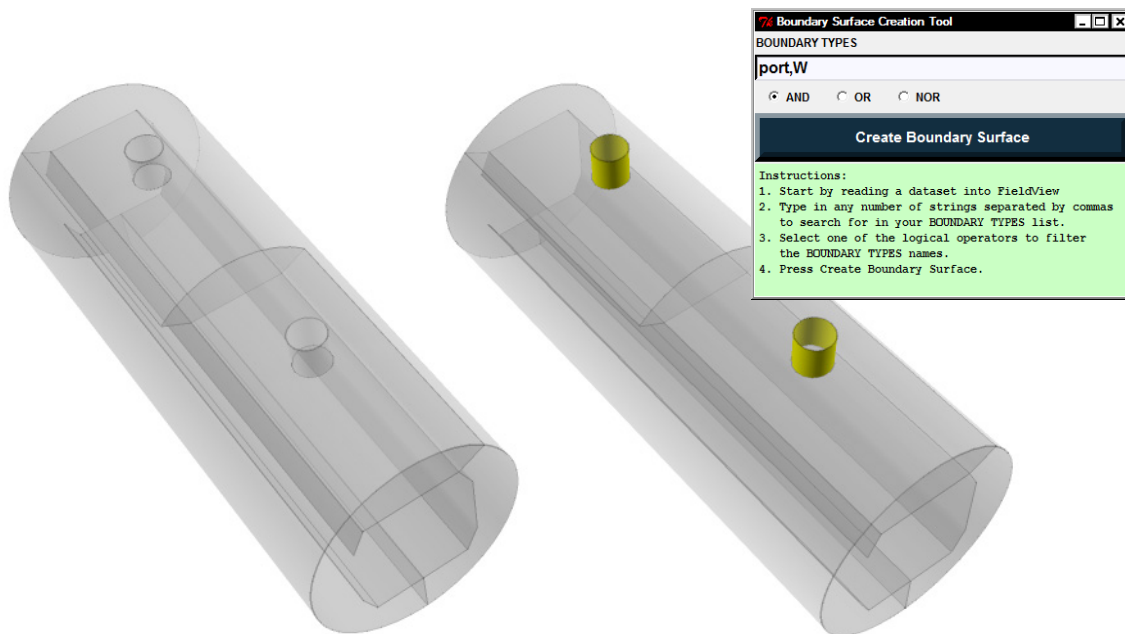
Figure 247 Using AND in the Boundary Type Manager

As an example, we will read the cocurrent heat exchanger dataset (`cocurrent.uns`) which can be found in your **FieldView** installation in the `\examples\heat_exchanger` sub-directory. Type in the strings **port** and **W**, separated by a comma. Select the AND filter and press the '**Create Boundary Surface**' button. When this is done, a new boundary surface having the types called **port1W** and **port2W** will be created. This boundary surface is colored yellow in **Figure 247 Using AND in the Boundary Type Manager**.

In the next example, type the strings **slip** and **distributor**, again separated by a comma. Select the OR filter and press the '**Create Boundary Surface**' button. In this case, a Boundary Surface containing the Boundary Types called **no_slip1**, **no_slip2**, **no_slip3**, **no_slip4**, **no_slip5** and **distributor** is created (see **Figure 248 Using OR in the Boundary Type Manager**).
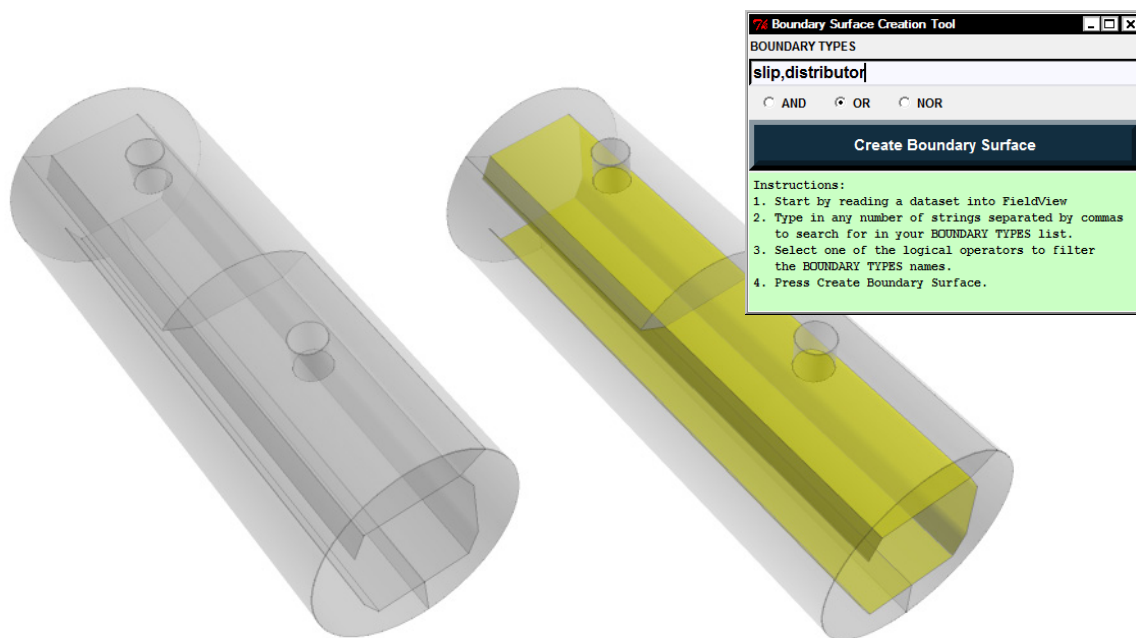
Figure 248 Using OR in the Boundary Type Manager

In the final example, leave the strings **slip** and **distributor** in place and select the NOR filter. When you create the boundary surface this time, it will contain all of the Boundary Types whose names do not contain either of the strings specified (see **Figure 249 Using NOR in the Boundary Type Manager**).
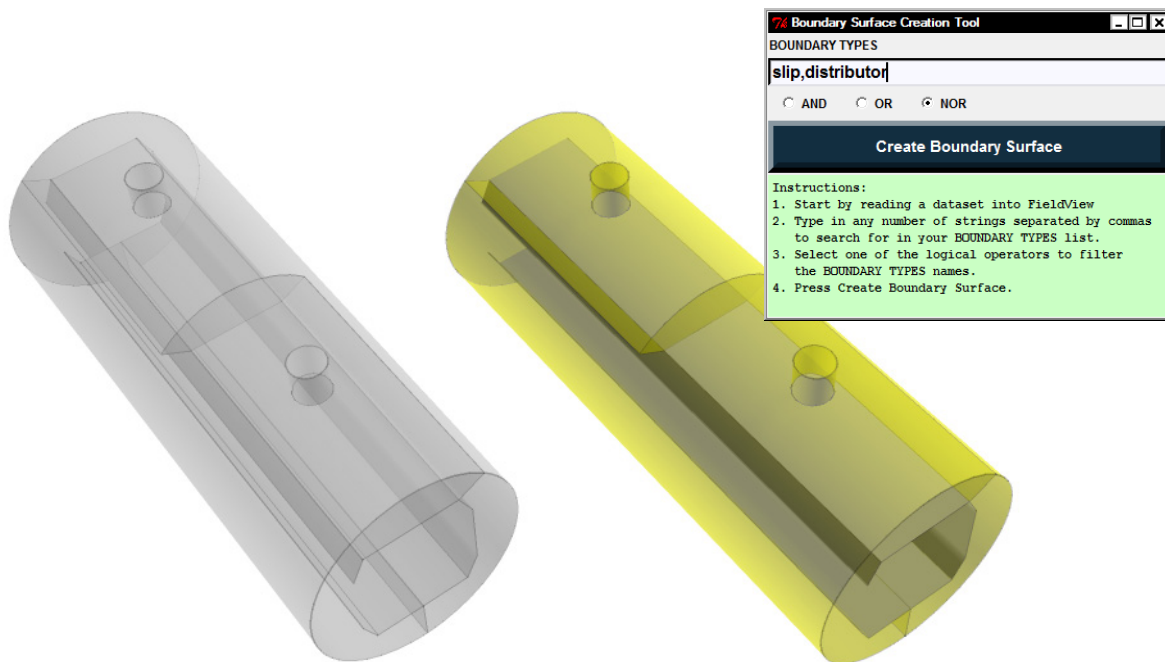
Figure 249 Using NOR in the Boundary Type Manager

My FVX

Users can customize an additional menu, accessed through My **FVX**, to point to their own **FVX** programs. An example of how this might look is illustrated in **Figure 250 FVX Utilities defined by User**.
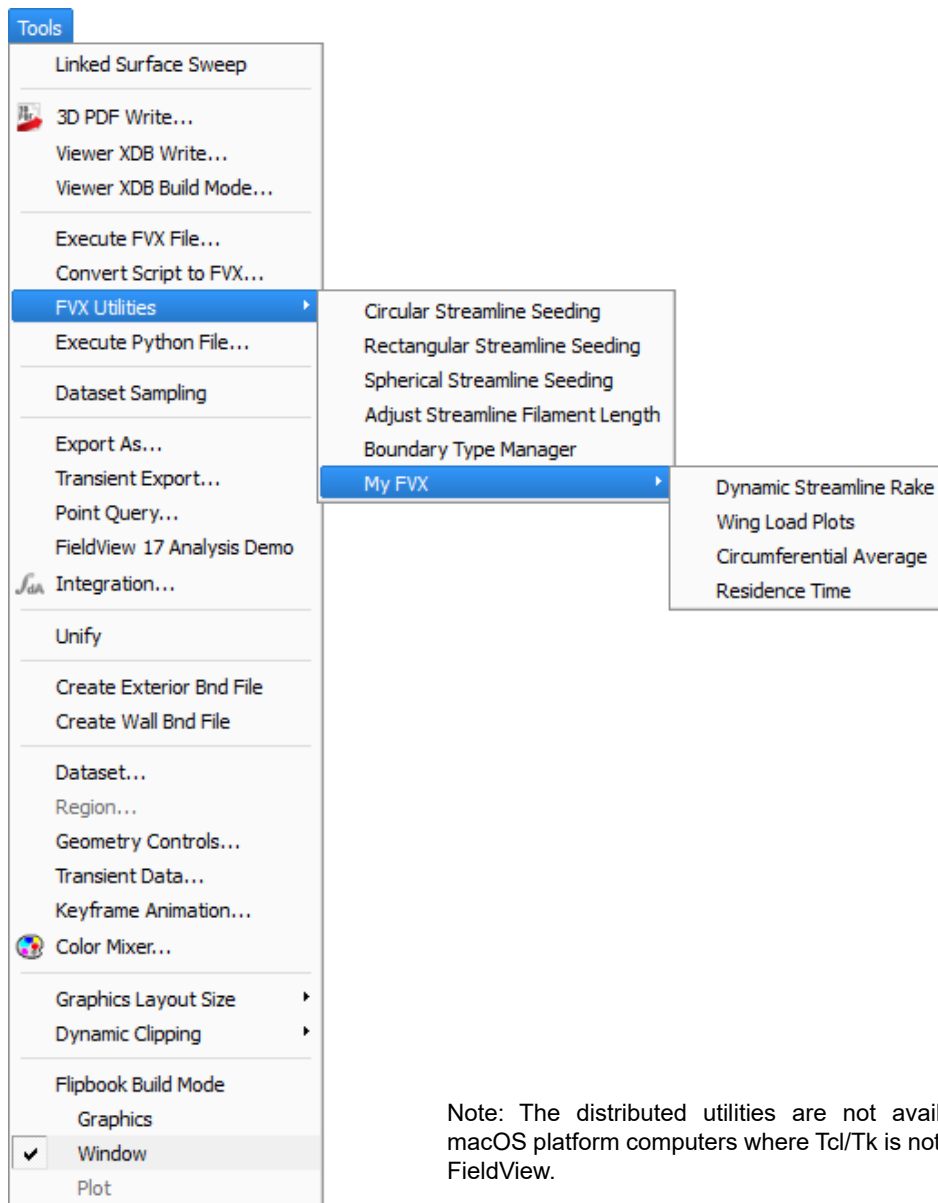
Note: The distributed utilities are not available on the macOS platform computers where Tcl/Tk is not supported in FieldView.

Figure 250 FVX Utilities defined by User

The contents of these menus are controlled using two files, `ReleaseFVX.fvxt` and `UserFVX.fvxt`, both located in the `/fvx_and_restarts` subdirectory. The syntax of both of these files is straightforward. For each menu entry, there must be two lines in the `.fvxt` file. The first entry contains the exact name of how this entry will look on the menu interface. The second line contains the complete path to the location, and name, of the **FVX** file that will be executed when the menu entry is pressed. For example, the `ReleaseFVX.fvxt` file may look something like this:

```
Circular Streamline Seeding
seed_circ_GUI.fvx
```

```
Rectangular Streamline Seeding
"seed_rect_GUI.fvx"

Spherical Streamline Seeding
seed_sphere_GUI.fvx

Adjust Streamline Filament Length
length_fraction.fvx

Plot Streamline vs Distance
plot_along_streamline.fvx

Plot Streamline vs Time
plot_along_streamline.fvx
```

This feature provides for up to 12 **FVX** utilities associated with the **FieldView** Release, and up to 20 **FVX** utilities associated with My **FVX**.

Execute Python File

The Execute **Python** File action will read and execute a **python** program. When this menu option is selected, the user will be presented with a file selector so that a **python** file may be selected. Upon filename selection, the program is read in and execution begins immediately. If there are any errors, the debugger output will display the details on the error. More detailed information may be found on **FVX** in **Chapter 4** of the **Reference Manual**.

Dataset Sampling

Sampling is a prerequisite for performing quantitative comparison between two simulation results based on different grids. It brings both results onto the same grid, allowing the creation of user-defined formulas containing differences between datasets. Dataset Sampling performs sampling of a results target dataset using all of the nodes of a grid target dataset. This operation can be time consuming and has various restrictions (you must use Server Append in client-server mode, parallel servers are not supported, and you must use Append Sampled Data to read the output of the sampling).

Surface Sampling allows sampling of a results target surface using all of the nodes of a grid target surface. This is many times faster than Dataset Sampling for looking at comparisons on one surface, and none of the above restrictions apply. Surface Sampling is done with an **FVX** command using **FVX** surface handles for the grid and results surfaces. For further information, please refer to **Surface to Surface Sampling for Dataset Comparison** in the **FieldView Reference Manual**.

Dataset Sampling is based on the Dataset Comparison methodology in which the function calculator is used to create formulas which span datasets. An underlying requirement which makes this possible is that the grids for the datasets being compared are identical. To enable this for datasets with different grids, the approach is to create a new 'Results ONLY' or Sampled Dataset. This choice offers the ben-

efit of full interactivity on the sampled dataset, permitting real-time sweeping of surfaces. However, the time to sample one 3D dataset against another can be large and prohibitive for real-world cases. To maximize interactivity and facilitate real-time surface sweeping, we recommend using XDB extracts. XDB extraction involves loading a dataset into **FieldView**; creating surfaces, rakes and/or feature detection objects to be extracted; selecting functions for exportation; and saving the XDB file(s) in Write or Build mode. For additional details, see **Writing XDB Files** or **Chapter 1 XDB Workflows for CFD**.

To create a Sampled Dataset, the following methodology is suggested:

- Read Dataset "A", a *RESULTS* Target

- Read Dataset "B", a *GRID* Target

- Select Dataset Sampling from the Tools Menu

- Create the Sampled Dataset "C"

- Append the Sampled Dataset from the Data Files Menu

The output created by interpolating the results from Dataset "A", the RESULTS Target, onto the grid from Dataset "B", the GRID Target, is the Sampled Dataset "C". It is a dataset which contains only the interpolated results. To read this data back into **FieldView**, it needs to already have the grid that was used for interpolation in memory. This special feature of Dataset "C" lets us append it, as sampled data to an active **FieldView** session.

This feature can be applied to study a series of different problems involving dataset comparisons. Some examples include:

Figure 251 Dataset Sample Methdology

Experimental versus CFD comparisons
(PIV or Wind Tunnel versus simulation results)

Solver versus Solver (Structured versus Unstructured, or tet versus hex)

Grid Refinement Studies (coarse grids versus finer ones)

Convergence over a solution volume

To create a Sampled Dataset, navigate to the Tools pulldown menu and select Dataset Sampling. This tool, when activated, will create a Sampled dataset which has the following properties:

1. It contains only RESULTS. There is no grid information at all.

2. It is based on the **FieldView** Unstructured File format for results.

3. It will be compatible with the grid of the dataset which was current when it was created.

4. It can only be read into **FieldView** using the 'Append Sampled Data' feature.

5. The GRID Target dataset must be read into **FieldView** immediately prior to appending the sampled dataset. Note: If the target grid already has results data, appending the Sampled Dataset will result in a newly appended dataset in **FieldView**, adding 1 to the Total: shown on the Tools > Dataset... panel, and to the Dataset selector on the main toolbar.

For the Dataset Sample Tool to be active, it is a requirement that there be at least two datasets read into **FieldView** at the same time.

The intent is that one will be the Results Target and the other will be the Grid Target. The Grid Target dataset is ALWAYS the current dataset in **FieldView**. For the case of a multiple dataset session, you will have the option to pick the Results Target from any of the non-current datasets. Depending on how many datasets you have loaded into your current **FieldView** session, the Tools menu will automatically update itself to provide you with a selection of candidate Results Targets.

In the GUI illustration at left, the Dataset Sampling menu shows that there is one dataset in the current **FieldView** session which can be chosen as the Results Target Dataset. The Grid Target is the current dataset, and in this case, it is Dataset #2. This is known since it does not show up as a choice in the Dataset Sampling Menu. Also, the Grid Target cannot be sampled onto itself.

Figure 252 Dataset Sampling Menu
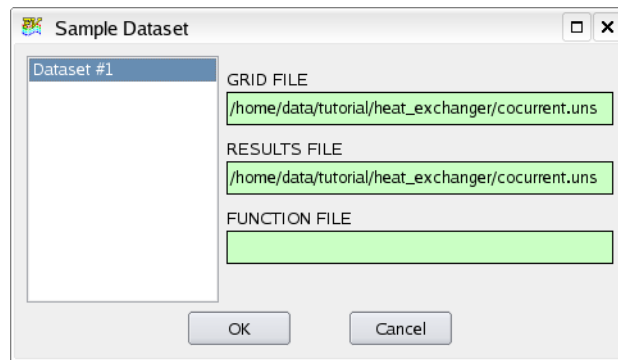
Figure 253 Sample Results Target Dataset Selector

An option to obtain filename information for any of the Results Target Datasets is available by selecting the Datasets... option from the pull-out menu. Making this choice brings up a GUI (see **Figure 253**) which will let you see the Target Dataset names, and let you make any choice from that list.
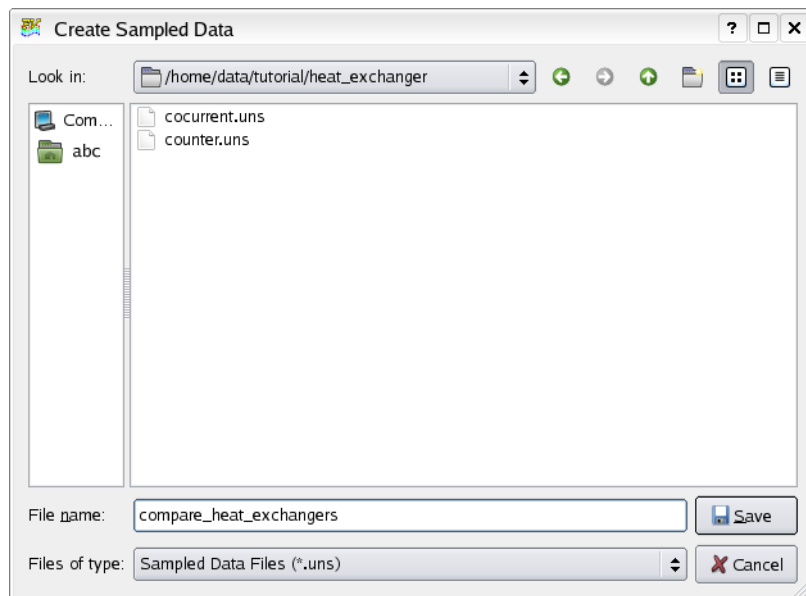
Figure 254 File Selector for Sampled Dataset

Once a Results Target Dataset has been selected, a file browser (see **Figure 254**) lets you specify the name and location for the Sampled Dataset.

There is no restriction on the type of data format for your Grid and Results Target choices. So, it is possible to sample a PLOT3D dataset onto **FieldView** Unstructured Dataset exported from the commercial solver, FLUENT.

There is a restriction on how the sampled dataset is read back into **FieldView**. The operation to read this dataset is to use Append Sampled Data... on the Data Input Menu (see **Figure 11** in the **FieldView Reference Manual**). In order for this dataset to be read correctly, the Grid Target Dataset must already be read into **FieldView**, and, it must be the last, or most recent dataset read into **FieldView**. Consider the following methodology:

- Read Dataset #1 into **FieldView** (the Results Target)

- Read Dataset #2 into **FieldView** (the Grid Target)

- Select Dataset #1 from the Dataset Sampling Menu
    (this implies that Dataset #2 is the current dataset)

- Create a Sampled Dataset

- Read the Sampled Dataset into **FieldView**

This stepwise approach will work since the Grid Target Dataset is the most recent dataset read into **FieldView**. Now consider the alternate methodology:

- Read Dataset #1 into **FieldView** (the Grid Target)

- Read Dataset #2 into **FieldView** (the Results Target)

- Make Dataset #1 the current dataset

- Select Dataset #2 from the Dataset Sampling Menu
    (this implies that Dataset #1 is the current dataset)

- Create a Sampled Dataset

- Read the Sampled Dataset into **FieldView**

When you attempt the last step above, you will receive the following error message:
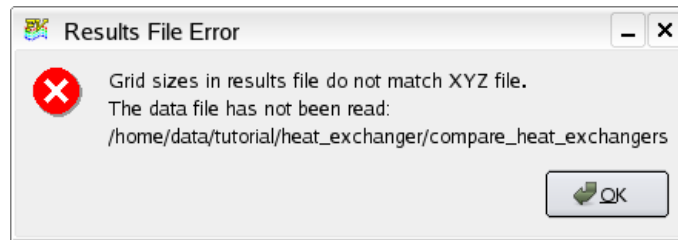
Figure 255 Append Sampled Dataset File Error

The reason for this error message is that the Grid Target, which the sampled datasets needs to map onto, is not the most recently read dataset.

Limitations

- Dataset Sampling cannot retrieve [BNDRY] variables.
- Append Sampled Data is not supported for transient data, regardless of the data format.

Note: Creating and reading sampled datasets is an order dependent operation.

It is possible to automate the process of creating Sampled Datasets using the following **FieldView** script command:

```
DATASET_SAMPLING  results_target_dataset_number  sampled_dataset_name
```

```
     results_target_dataset_number = Number (order) of Results Target
     sampled_dataset_name           = name of Sampled Dataset to be created
```

A simple script example to read a RESTART and create a sampled dataset is shown below:

```
! Read a tet mesh dataset followed by a hex mesh
RESTART ALL tet_then_hex

! Choose the second [hex] dataset as the Grid Target
SELECT DATASET 2

! Choose the [tet] dataset as the Results Target, and
!  create a sampled dataset called tet_2_hex
DATASET_SAMPLING 1 tet_2_hex
```

The `DATASET_SAMPLING` script command may be executed in **FVX** using the `fv_script` command.

Dataset Sampling Example: Hex vs. Tet Mesh
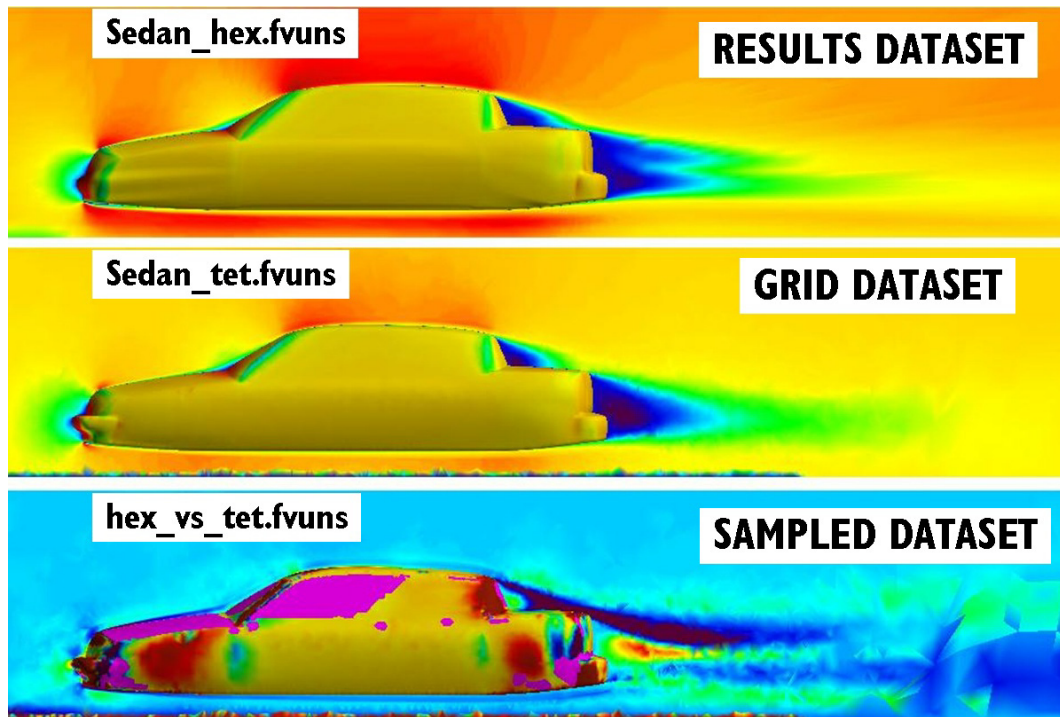
Figure 256 Dataset Sampling on Hex and Tet Grids

In the example illustrated in **Figure 256**, the Results Target dataset is a solution generated for a hex mesh. The Grid Target dataset contains a solution for the same geometry, performed on a tet mesh. The resulting Sampled Dataset has been appended to the same **FieldView** session, and the function calculator has been used to create formulas which represent the difference in axial velocity and pressure for the two cases. In all three illustrations, the coordinate plane is colored using the axial velocity; in the bottom picture, the axial velocity is shown as a difference between the hex case and the tet case. Differences, particularly in the area downstream from the car are quite evident.

Pressure is shown on the car body itself, again for all three illustrations, with the bottom one showing the difference between the two solutions. The magenta colored areas indicate locations where no results are available from the Target Results data for the Target Grid. It is notable that there are several locations on the body of the car itself where there are small differences between the two grids for the two cases.

Dataset Sampling Example: Static Mixer
In this example, we are comparing two different geometries. In the top dataset, a static mixer has been modelled to study six (6) mixer blades. It is being compared against a mixer with only four (4) blades. The geometry inlets are located at the same point, and are aligned to the left of this illustration.
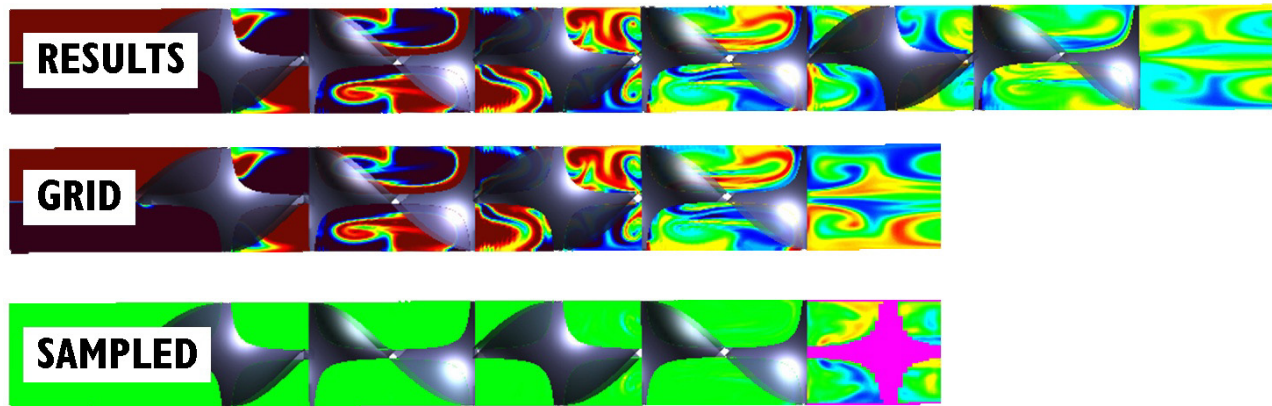
Figure 257 Dataset Sampling on Static Mixers, inlets matched

The sampled dataset is shown at the bottom of this illustration. Coordinate planes passing through the centerline of the model are contoured using a tracer species to show mixing. In the sampled dataset, the magenta region shows that there are no results available from the six (6) blade Results Target dataset at the position of the fifth blade (the grid dataset has only four (4) blades). Differences in tracer species are essentially zero until the final zone in the Grid Target dataset.

To compare flow at the exits of both models, we can use a simple region file to transform the grid dataset as follows:

```
FVREG 1
DATASET_COORD_TYPE cartesian
    ORIGIN        0   0  -0.144
    Z_AXIS        0   0   1
    THETA_AXIS    1   0   0
```
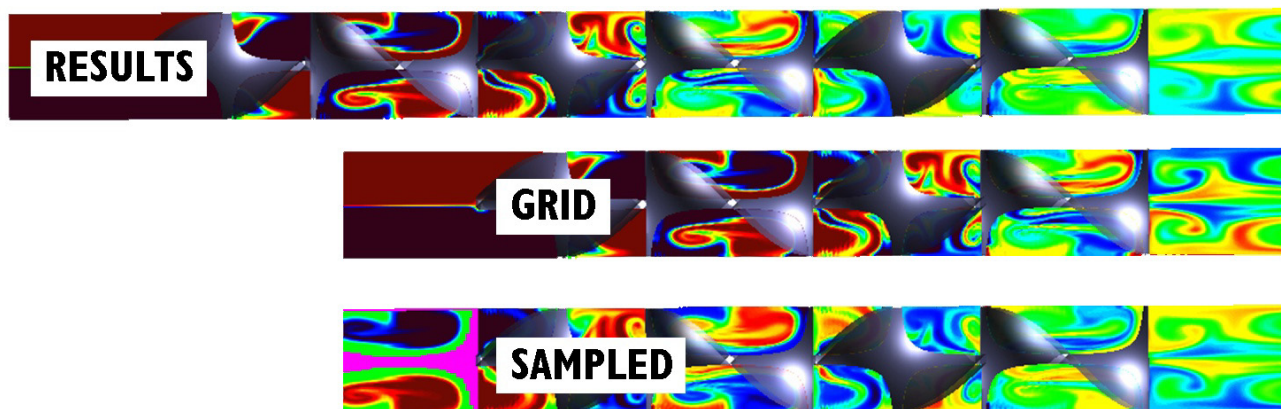

Figure 258 Dataset Sampling on Static Mixers, outlets matched

Here, the value of -0.144 is sufficient to shift the grid target dataset along its axis so that the outlet planes between the Results Target Dataset and the Grid Target Dataset match.

When the Sampled Dataset is calculated this time, we see a magenta zone at the entrance, corresponding to the presence of the second mixing blade in the six (6) blade Results Target dataset, which in not present in the translated grid. In this case, differences between the tracer species after six (6) blades versus four (4) blades are clearly seen.

Dataset Comparison for FieldView Client-Server
By default, there is no current ability to communicate between two different server processes.
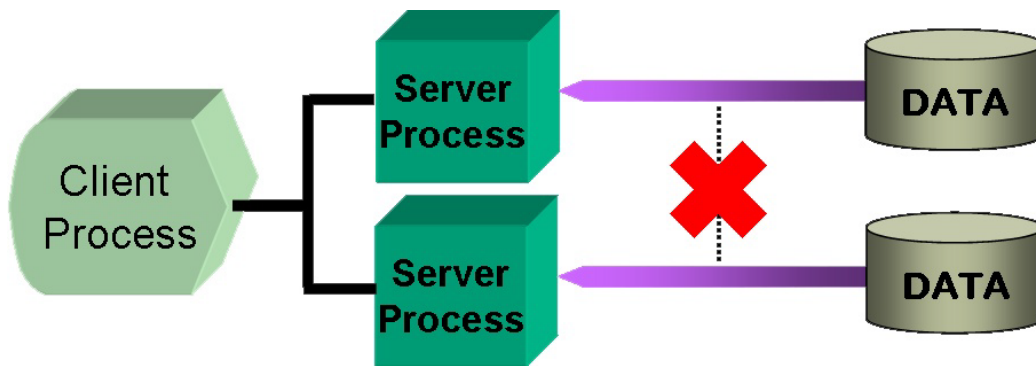
Figure 259 No Communication between Server Processes

To do Dataset Sampling in Client-Server mode you need to set the same SERVER process to read more than one dataset.
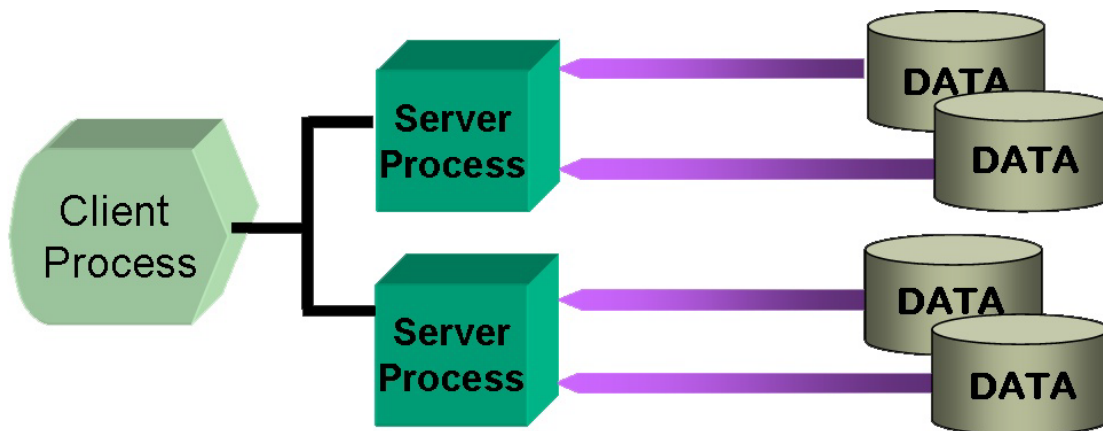
Figure 260 Appended Server Processes

To turn this feature on, first read a dataset into **FieldView** using the Client-Server feature. Then, to read another dataset to that same SERVER process, in the Data Files Menu pulldown (see **Figure 11**), turn the Server Append button ON. With the minimum of two datasets read into **FieldView** in this way, the Dataset Sampling and Dataset Comparison features will be fully enabled.

Export As...

Export As... will cause the information from the current surface or rake of only the current dataset to be output to a file. When Export As... is selected from the Tools menu, the user will be presented with a file selector so that an output file may be given. If the selected filename already exists, the user will be given the option of over-writing the file or canceling. Upon filename selection, the numerical information for the current surface or rake will be output to the file as described below. The variables that are output are the variables currently loaded in the Scalar, Vector, and Threshold function registers. The Iso-surface register will not be output.

The "Files of type:" options depend on the current panel and is initially set at the beginning of each **FieldView** session. If the setting is changed, **FieldView** will remember it for the next Export. All text file exports are identical in format to previous versions of **FieldView**, but text files exported interactively are appended with the extension `.txt` as of **FieldView 17**. As in previous versions of **FieldView**, Streamlines and Vortex Cores/Surface Flows are exported as **FieldView** Particle Paths, and interactively are appended with the extension `.fvp` as of **FieldView 17**.

The MAT-File format `(*.mat)` provides efficient input to and support by both MATLAB and GNU Octave. CSV (Comma Separated Values) `(*.csv)` is a file format that can be read by many programs such as Microsoft Excel and Scilab. These exports facilitate additional outside analysis outside of **FieldView** and were introduced for Coordinate and unstructured Boundary Surfaces in **FieldView 17**, with support for additional objects expected in future releases. Format details appear in **Appendix J** (MAT-File Export) and **Appendix K** (CSV Export) of the **Reference Manual**.

Note: Surface Plots cannot be exported. If two or more of these registers contain the same variable, that variable will only be written out once. Only Streamline rake and 2D Plot Export files can be read back into **FieldView** (using Particle Paths Import and 2D Plot Import, respectively).

Appropriate error messages will be generated if the current panel does not support Export As..., or if the current surface or rake on the current panel is not visible.

If the 2D Plot panel is the current panel when the user selects Export As... from the Tools menu, it will act as if the user had pressed Export... on the 2D Plot panel.

The text format of the surface or rake exported is given in **Appendix I** of the **Reference Manual**.

Transient Export...

A surface may be exported as a dataset is sweeping through time via the "Transient Export..." entry on the Tools menu (see **Figure 231**). When "Transient Export" is turned on, the Transient Data Controls panel (see **Figure 117**) comes up with the Sweep button changed to "Export As..." Turning this on brings up the file selector and "Files of type:" options described in **Export As...**

Appropriate error messages will be generated if the current panel does not support Transient Export..., or if the current surface or rake on the current panel is not visible.

Point Query...

This feature delivers high performance query of your simulation results with MAT-File and CSV I/O. It is a great way to extract information from your CFD results, from just a few points to be compared to EFD pressure probes to a full re-sampling on millions of nodes. Note that to obtain the right probing speed, the Grid Processing setting should be adjusted at dataset read time (for more details, see the **GRID PROCESSING** section of **Chapter 1** of the **Reference Manual**). With the setting "More", speeds on the order of 200,000 probes/sec can be obtained. For best performance, the MAT-File I/O should be used since it is a binary format.

Point Query is based on the same MAT-File, CSV, and plain text formats as described in **Export As... FieldView** will look at arrays in the MAT-File or columns in the CSV file named X Y and Z, or a simple plain text list of coordinate values.

- For MAT-File or CSV, if all three are found, **FieldView** will probe the current dataset at these locations and output the current scalar, vector and threshold variable to the same format.

- For MAT-File or CSV, if any two are found, the missing one will be assumed to be all zeros and **FieldView** will proceed with the probing.

- For plain text input, an error will result if the number of values in the file are not a multple of 3.

The current variables (scalar S, vector components U, V and W, threshold scalar T) will be probed and exported, in addition to the XYZ coordinates. These variables can be selected interactively through the Function Specification panel. In batch, the recommended approach is to select them for a surface and to make that surface the current object.

The Tools menu entry, "Point Query..." (see **Figure 231**), will bring up a file selector to open an input file of points for analysis. If the user does not Cancel, a second file selector will be brought up to specify the output file. The file type extension of the input file will be the default for the output file.

The Point Query feature is based on an interpolation of the CFD results from the mesh nodes onto the point locations provided as input. If a point lays outside of the current dataset, the result will be undefined (NaN) for all variables at that location.

Integration

A discussion on the operation and features of the Integration panel has been provided in **Integration Controls**.

Unify

The Unify button is used to make the display style of all surfaces of a particular type (Computational, Iso, Coordinate, etc.) the same. This applies to all surfaces/rakes of the current dataset only. The following features will be unified: Display Type, Color, Scalar Variable, Scalar min/max (Color map range), Vector display (including vector head type (2D or 3D) and scaling and vector scale factor), Line Thickness, Show Mesh, Transparency and Thresholding. The following items are not affected by Unify: Position/Iso-surface value, Visibility and Subsetting. See the **special note** about how Local Scalars behave with Unify.

For example, suppose you have 5 surfaces as shown below:

Surface Number	Type	Position/Value	Description
1	computational	I = 7	Pressure colored shaded surface
2	computational	K = 5	Geometry colored mesh surface
3	computational	J = 3	Vector display with Show Mesh on
4	iso	Vel. = 0	Geometry colored surface
5	iso	Y = 9	Temperature colored mesh surface

If the Computational Surface panel is up, and the current surface is 1, pressing the Unify button will result in the 5 surfaces shown below:

Surface Number	Type	Position/Value	Description
1	computational	I = 7	Pressure colored shaded surface
2	**computational**	**K = 5**	**Pressure colored shaded surface**
3	**computational**	**J = 3**	**Pressure colored shaded surface**
4	iso	Vel. = 0	Geometry colored surface
5	iso	Y = 9	Temperature colored mesh surface

The surfaces shown in **bold** have been updated. Note that only the computational surfaces have changed, because the current surface is a Computational Surface. Also note that the position (I, J, K, etc.) has not been changed on the updated surfaces.

Create Exterior Bnd File

This tool and the following provide a means to automatically create two different types of structured boundary files. A `.fvbnd` file will be generated for the current dataset. If a `.fvbnd` file for the current grid already exists, and the user chooses to overwrite it, the original file will be renamed `.fvbnd.old` and the specifications contained in it will be merged into the new `.fvbnd` file. The format of the `.fvbnd` file is described in **Appendix H** of the **Reference Manual**. Boundaries will be renumbered, but the specifications will otherwise be the same. No work will be done to ensure that the result of merging the generated and existing boundary specifications will not create duplicate boundaries or boundary names.

Note: This tool is not supported for parallel servers.

This particular tool will create a file named `gridfilename.fvbnd` with boundary types "`exterior-grid#`" (if the dataset is single grid the grid number is omitted from the boundary name). These boundaries will specify the exteriors of each grid.

Modifying or creating .fvbnd files for Exterior Boundaries can also be automated by using the **DataGuide™** command line option `-pN` along with `-create_exterior_fvbnd`, while also calling a Data Input restart with -f, or a script with `[ -s | -fvx ]`, provided that the script includes a Data File (or Complete) restart.

See **Chapter 1** of the **User's Guide** for more information about **DataGuide™** Processing. Note that this feature cannot be used if your data has been read with Parallel FieldView, unless read in Partitioned File (PFPR) format. If tried, FieldView will issue a corresponding message.

Note: When using the command line argument, boundary files are only created for the last (highest-numbered) dataset in a multi-dataset data restart file.

Please refer to **Appendix H** of the **Reference Manual** for complete details about the Structured Boundary file.

Create Wall Bnd File

This tool will create a file named `gridfilename.fvbnd` with a boundary type "`walls`" for all no slip walls, and create boundary types "`wall-grid#`" per grid (if the dataset is single grid the per grid boundary names are omitted). These boundaries will specify the exteriors of each grid where velocity is zero. When running interactively the user is prompted to confirm that velocity magnitude (in whatever form that exists) is currently loaded as the current Threshold function prior to calculating the walls. As with Create Exterior Bnd File, this tool only creates a wall boundary file for the current dataset. The format of the `.fvbnd` file is described in **Appendix H** of the **Reference Manual**.

Note: This tool is not supported for parallel servers

Modifying or creating .fvbnd files for Wall Boundaries can also be automated by using the **DataGuide™** command line option `-pN` along with `-create_wall_fvbnd`, while also calling a Data Input restart with -f, or a script with `[ -s | -fvx ]`, provided that the script includes a Data File (or Complete) restart.

See **Chapter 1** of the **User's Guide** for more information about **DataGuide™** Processing. Note that this feature cannot be used if your data has been read with Parallel FieldView, unless read in Partitioned File (PFPR) format. If tried, FieldView will issue a corresponding message.

When running with the command line option, **FieldView** expects that the function "Velocity Magnitude [PLOT3D]", or at least one of the following user defined functions be present: "velocity magnitude", "mag("velocity")", or "velocity". If none of these functions are present a structured boundary file may still be written, but it may not correspond to anything meaningful.

> If a dataset's grid(s) are defined in such a way that `K=1` (for instance) represents the boundary surface of the model geometry, then the user can create a function using the Function Formula Specification panel ("CFD Calculator") defined as `"K"-1` and load this function into the Threshold register. Then use this tool, which will create a Structured Boundary file consisting of the necessary information to create the geometry surface as a Boundary Surface.
> When using the command line argument, boundary files are only created for the last (highest-numbered) dataset in a multi-dataset data restart file.

Dataset

A discussion on the operation and features of the Dataset Controls panel has been provided in **Dataset Controls**.

Region

A discussion on the operation and features of the Region Controls panel has been provided in **Region Controls**.

Setup

Transient Data

A discussion on the operation and features of the Transient Data Controls panel has been provided in **Transient Data Controls**.

Keyframe Animation

A discussion on the operation and features of the Keyframe Animation panel has been provided in **Keyframe Animation**.

Color Mixer

The Color Mixer (**Figure 261**) will allow the user to change 8 of the 10 colors that **FieldView** uses for Geometric and Background colors. Black and White are not editable. The set colors are global over all objects existing on the screen or created anew.

The panel will initially display of the Geometric colors as both the Old and New color chips, and its Red/Green/Blue (RGB) value is displayed in the RGB sliders and type-ins. To choose a different color chip to edit, press on the desired color chip to select it. As the RBG values are changed, the New color chip and the Select Color to Change chip update to reflect the new color. The Old color chip continues to display the old color (see panel description on next page).

The user may change more than one color at a time, before applying the changes. Pressing the Apply button commits all color changes made to any chip. The colors committed would be those displayed in the color chip selector in the upper portion of the panel.

Note: In 8-bit color, updates to surface coloring will occur simultaneously as the RGB values are changed.

Press the desired color chip to edit.

Values for the three components of each color are controlled by these sliders. Each of Red, Green, and Blue range in value from 0 to 255. Adjusting any slider position will simultaneously change the value field and also the New chip color.

Instead of using the sliders, these fields can be used to enter in RGB values. Press Enter to accept each value.

When Apply is pressed, the Old color chip will update to the new color value of the selected chip. When color changes are applied, they are propagated to all graphical objects and GUI components:
   Surfaces
   Legends, including contour legends
   Titles
   Streamlines
   Particle Paths

Press this button to revert to **FieldView**'s default color value for this chip.

Press this button to launch an advanced color mixer panel.

Pressing this button cancels all unapplied changes and restores the 8 color chips in the upper portion of the panel to their last applied RGB values, as well as the Old and New chips, and RGB sliders/type-ins.

Press this button to close the panel. If there are un-applied changes, you will be asked to confirm the exit, as it will cancel all un-applied changes.

Figure 261 Color Mixer

When the Advanced button is pressed on the Color Mixer panel, the Select Color panel, shown in **Figure 262**, is launched. This panel has additional controls for creating custom colors. It also features the ability to save custom created colors which are saved across **FieldView** sessions.
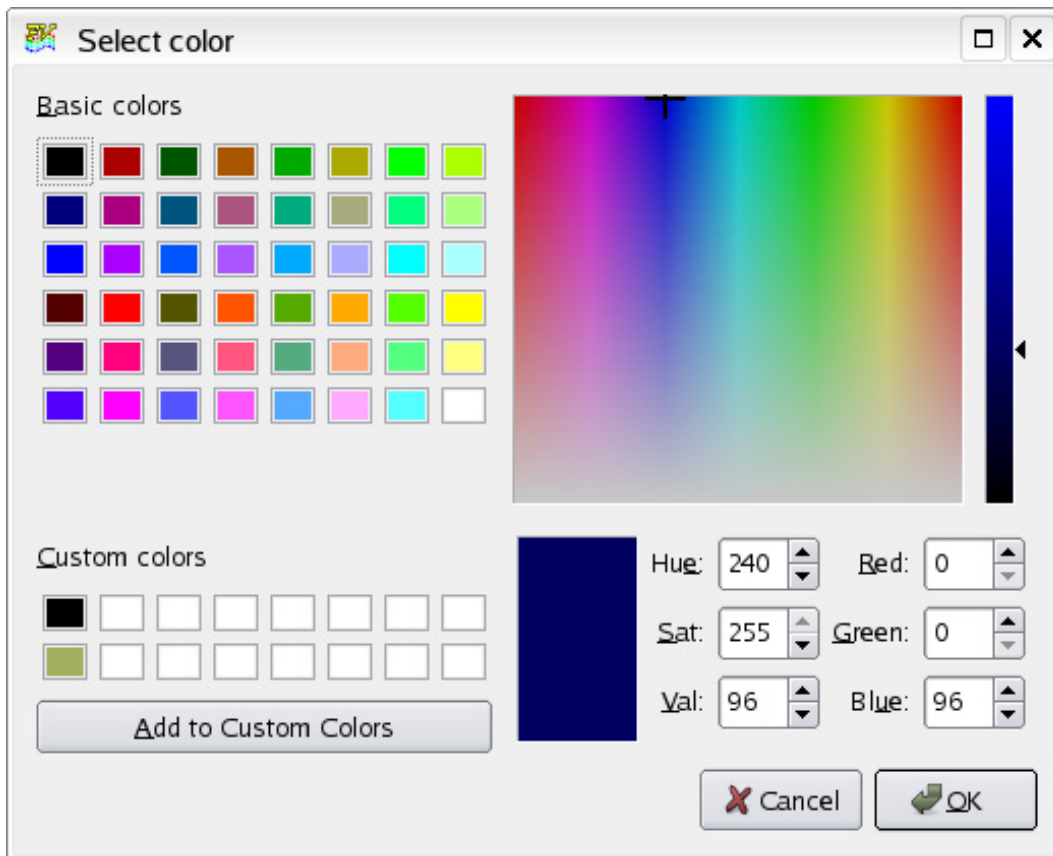
Figure 262 Select Color panel for saving custom colors

If the New color chip has been modified by adjusting any of the RGB values and the Exit button is pressed before the Apply or Cancel button, a confirmation pop-up will be displayed (**Figure 263**):
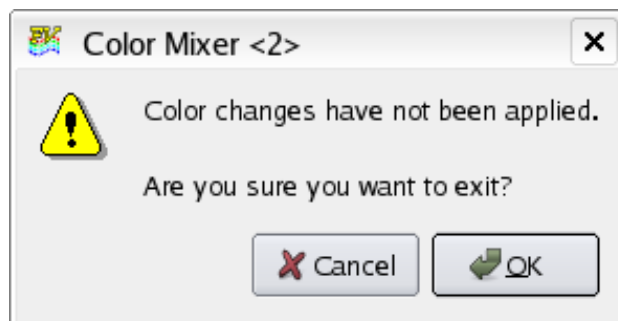
Figure 263 Color Mixer Pop-up

Most systems should have a file called `rgb.txt` which is usually found in `/usr/lib/X11` or `/etc/X11` (`/usr/openwin/lib` for SUN Solaris) which contains hundreds of colors already defined. This list will allow the user easy starting points to define such colors as brown, tan, pink, slate, etc.

Graphics Layout Size

This sub-menu (**Figure 264**) is redundantly accessible from the View menu and provides a way of easily setting the graphics window sizes for use during Flipbook and Keyframe animation creation. The pull-down list offers several standard image sizes. Choosing one of these predefined sizes will cause the graphics window to be resized. This will override the window size previously set using the command line switch `-size=widthxheight` if used during **FieldView** start-up. The specific window you select will be maintained when you resize the **FieldView** GUI manually by grabbing the window border with the left mouse button and dragging. If you want the graphics area to automatically update as you interactively resize (or maximize) the GUI, you must first select Tools → Graphics Layout Size → Default. The graphics area can also be changed using the `SIZE` script command (see **Chapter 5** of the **Reference Manual**).

<div align="center">

1080p: 1920x1080

720p: 1280x720

480p: 854x480

Default
640x480 NTSC
320x240 1/4 NTSC
160x120 1/8 NTSC
720x576 PAL
360x288 1/4 PAL
180x144 1/8 PAL
720x486 D1

</div>
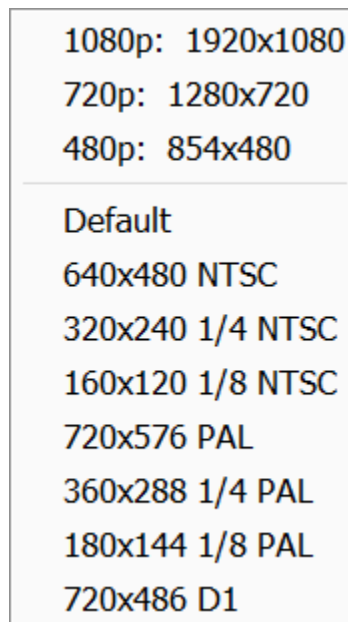
Figure 264 Graphics Layout Size Sub-menu

Selecting the Default choice will size the graphics window to the value stored automatically in the **FieldView** preference file (`FieldView.ini`) or as specified with the command line argument `-size`. If this is the first time you've launched **FieldView**, the default graphics area will be the largest which can be created on your maximized **FieldView** interface (unless overridden with the command line argument `-size`).

Note that when running with the `-batch` command line option, the default size is 640x480, unless launched with the command line argument `-size`. Note that for batch operation, your scripts can also contain a `SIZE` command.

Any images created using Print or Save Image on the File menu or animations created during Flipbook Build Mode or Keyframe Animation will be the same size as that of the graphics window.

Dynamic Clipping

FieldView's Dynamic Clipping feature allows users to visually "clip" (threshold) all current graphical objects, such as coordinate planes, boundary surfaces, streamlines, iso-surfaces, etc. This feature is available for use both interactively in **FieldView** as well as functionally through numerical specification in a user defined **FVX** script.

Currently, two types of clip definitions are available: clip lines and clip boxes. Clip lines are defined by a point and a normal vector, indicating the starting point and direction in which the current display is to be clipped. Similarly, clip boxes trim objects outside the bounds of a user-defined box.

Users can save several clip lines and/or clip boxes as a clip group. Once saved, a clip group can be toggled on and off, deleted, or replaced with another defined clip group. **FieldView** allows up to 12 dynamic clip groups to be specified at any given time, each defined by up to 16 clip lines, 4 clip boxes, or a combination of the two.

New clip groups can be applied to a dataset interactively in the **FieldView** graphics window or through the use of functions in **FVX**.

*Dynamic Clipping in **FieldView***
The following steps outline the process of creating, editing, and deleting a new clip group in **FieldView**.

1. Load a dataset into **FieldView** and create any desired streamlines, coordinate surfaces, boundary surfaces, iso-surfaces, etc.

2. From the Tools pull-down menu on the Main Menu bar, choose Dynamic Clipping -> Manage Clip Groups….

3. The Clip Group Definition panel (**Figure 265**) will appear on the screen. As no previously-defined clip groups exist, a clip group can only be created. Click Create.

4. The default name of the clip group `Clip Group #1` appears in the Name: text field. This name can be changed at any time.

5. Instructions on how to apply clip lines and clip boxes to the current view appear on the right side of the Clip Group Definition panel. Use the mouse controls to position the dataset for the first (or only) clipping.

Figure 265 Clip Group Definition Panel

6. To create a clip line, position the mouse at the cutting point. Press and hold CTRL-ALT-M1 (left mouse button) and drag to define the clip line. The small glyph located in the middle of the clip line indicates the side to be clipped. (Note: The clip line will cut completely across the current view no matter how far the line is dragged or which surfaces it does/does not cover. Thus, to clip the direction opposite to that which the glyph indicates, swivel the endpoint to the opposite side of the start point).

Menu: Dynamic Clipping menu is accessed from the Tools pull-down menu on the Main Menu bar. Click Manage Clip Groups... to create a new clip group.
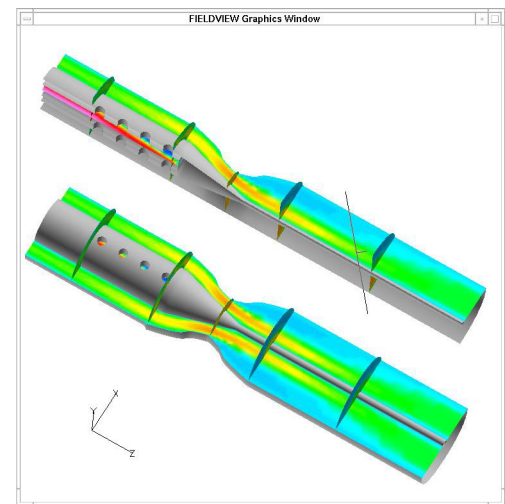
Before Clip: Clip line is applied to the first of two datasets. The small glyph in the middle of the clip line indicates the graphics to the right of the line will be removed.

Figure 266 Illustration of the Dynamic Clip Line
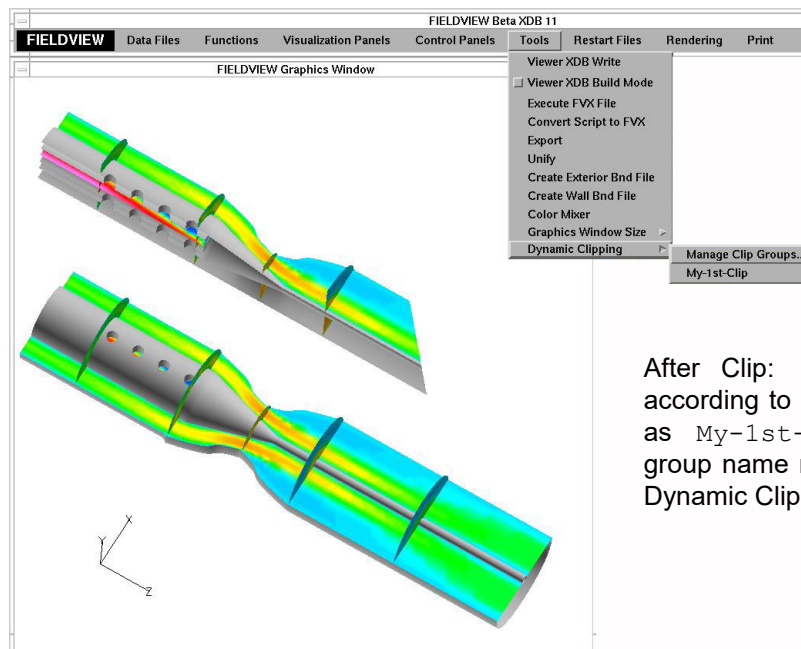
After Clip: Dataset 1 is cut according to specifications saved as My-1st-Clip. This clip group name now appears on the Dynamic Clipping submenu.

Figure 267 Dynamic Clip Line on a Dataset

- To create a clip box, position the mouse at the desired start point of the box. This point defines the first corner. Press and hold CTRL-ALT-M3 (right mouse button) and drag to define the opposite corner of the clip box. All graphics lying outside the boundary of the box will be clipped.
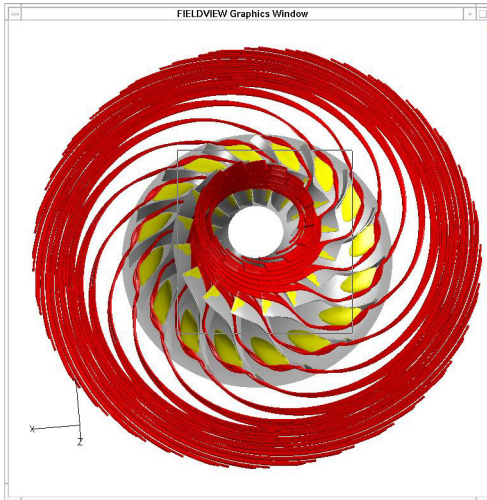
Before Clip: Clip box is dragged around the center region of the rotationally-duplicated impeller.
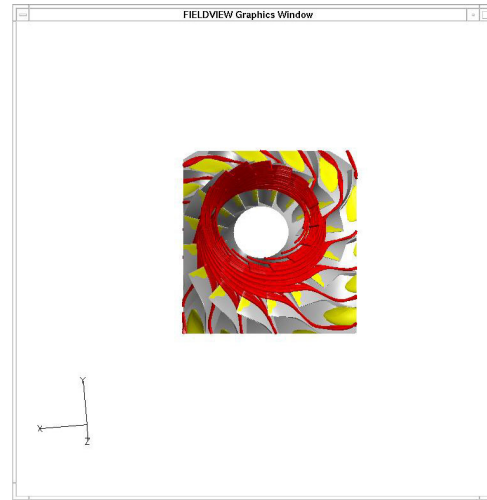
After Clip: Objects outside boundary of box are removed.

Figure 268 Illustration of the Dynamic Clipping Box
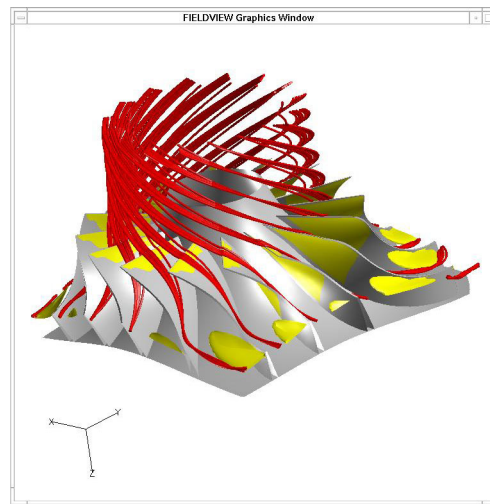
Clipped Image: Remaining portion of dataset is rotated to show clipped boundary surfaces, iso-surfaces, and streamlines.

Figure 269 Dynamic Clipping Box on a Dataset

7. While defining the current clip group, the view in the graphics window can be adjusted using the mouse controls to set up for more clip lines and/or clip boxes. At any time, the Undo (last) button can be used to remove the most recent clip line or clip box added to the clip group.

Note: Clip lines and clip boxes can only be removed while their clip group is being defined. Once the Clip Group Definition panel is closed, the clip group properties cannot be edited.

8. Once all clippings have been made, click Close. To toggle the visibility of a clip group on and off, click the Tools pull-down menu on the Main Menu bar and bring up the Dynamic Clipping sub-menu. In addition to the Manage Clip Groups… option, all saved clip groups are listed. Click the name of a clip group to toggle its visibility. When more than one clip groups are saved, this sub-menu can be used to change which clip group is currently visible, if any.

Note: No more than one clip group can be visualized at any given time.

9. To delete or rename a clip group, return to the Clip Group Definition panel by choosing Tools Dynamic Clipping Manage Clip Groups… from the Main Toolbar.
    • To rename a group, select a clip group from the list. Type a new group name in the **Name:** field and press **Enter**.
    • To delete a group, select a clip group from the list and press **Delete**.

Dynamic Clipping in FVX

A dynamic clip group can be created by specifying a series of clip definitions:

```
dynamic_clip_table={
    dataset = 1,
    name = "clipping_line",
    clip_definitions = {
        { point  = {3.27324,    5.43,     4.6723  },
          normal = {3.27324,  18.105,  22.874    }
            },
        { point  = {3.27324,    4.1625,  2.85213 },
          normal = {3.27324,    2.8950,  1.03196 }
            },
                        }, -- end of definitions
        active = 'on|off',
        }
create_dynamic_clip(dynamic_clip_table)
```

dynamic_clip_table			
Field	**Data Type**	**Comments**	**Default**
dataset	number	dataset number	current
name	string	must have a name (explicit definition required)	
clip_definitions	table	point table, normal table	
point	table	{point1, point2, point3}	
normal	table	{point1, point2, point3} ordered as X, Y, Z in dataset space	
active	string	"on" or "off"	"off"

FieldView automatically checks for active **FVX**-defined dynamic clips and GUI-defined dynamic clips. If an **FVX** defined dynamic clip is turned ON, then ALL the other **FVX**- and GUI-defined dynamic clips are turned off. Similar to other surfaces, the user can Create, Delete, Modify, and Query the `create_dynamic_clip` function via the handle. These **FVX** commands can only be applied to **FVX**-created dynamic clips, and not to GUI-created dynamic clips. The dynamically created GUI must also be maintained.

What happens when you delete an FVX created clip using the GUI?
The handle will become nil, and this clip group will no longer be available from the GUI.

What happens if you rename an FVX created clip using the GUI?
The **FVX** context for this clip group will be lost, and will no longer be accessible from **FVX**.

What happens if you create more than 12 dynamic clip groups using FVX?
Only the first 12 show up. As soon as a dynamic clip group is created, it will need to show up on the GUI.

What if you define more than 64 clip definitions within a clip group?
Only the first 64 are recognized. The others are ignored. On a query, only the first 64 (recognized) will be dumped out.

Guidelines of Using Dynamic Clipping

Datasets

There are no limitations on the type of dataset needed to utilize dynamic clipping in **FieldView**. Dynamic clipping can be applied to datasets with and without regions, mirroring capabilities, and the ability to be rotationally duplicated.

Dynamic clipping, however, is unique to a dataset. Clip group definitions can be turned on, toggled, and deleted for each dataset. If more than one datasets are loaded into **FieldView**, each dataset can show a different clip group without affecting the other datasets.
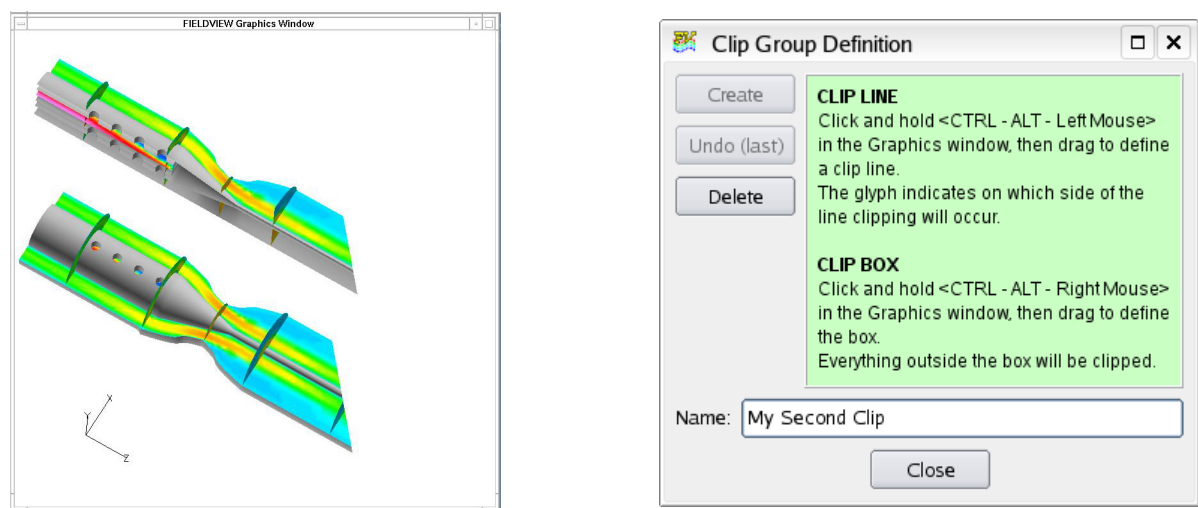
Double Dataset: Two datasets from the bio-medical tutorial each show clipping from clip group `My-1st-Clip`.

New Clip Group: A new clip group named `My Second Clip` is created and defined in the Clip Group Definition panel.

Figure 270 Dynamic Clipping Line on a Second Dataset
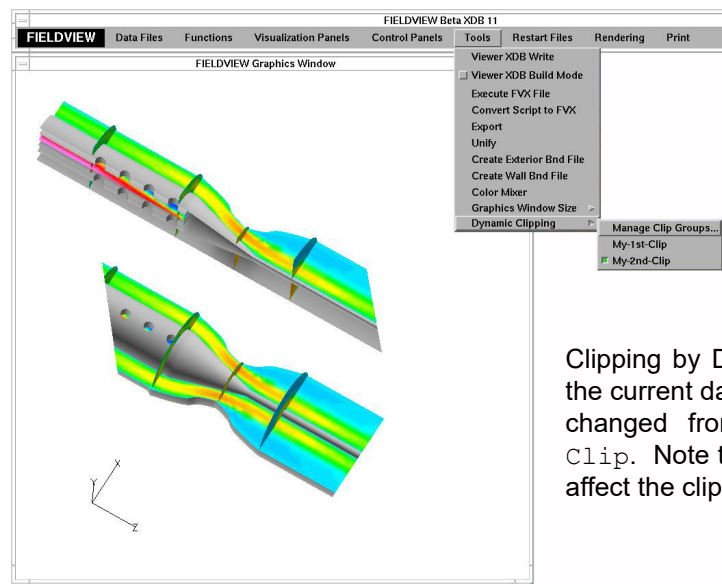
Clipping by Dataset: After making Dataset 2 the current dataset, the visualized clip group is changed from `My-1st-Clip` to `My-2nd-Clip`. Note that making this change does not affect the clipping of Dataset 1.

Figure 271 A Second Dynamic Clipping Group

Editing Clip Groups

As stated above, clip groups can only be edited when they are initially defined. Once the Clip Group Definition panel that was used to create a given clip group is closed, the group can no longer be edited (i.e. the Undo (last) feature cannot be accessed for the given clip group). It is important to note that

certain **FieldView** actions will cause the Clip Group Definition panel to close automatically, closing out editing options on any current clip group. These actions include:

- Changing the time step in a transient data file
- Loading a new dataset into **FieldView** in Replace mode
- Loading a new dataset into **FieldView** in Append mode

Creating Objects with Dynamic Clipping On

Boundary surfaces, coordinate surfaces, iso-surfaces, streamlines, computational surfaces, etc. created while a clip group is visible are clipped according to the clip lines and boxes of the current clip group. Toggling the visibility of the clip group off will correctly restore these objects.

Duplication, Mirroring, and Transformations

The dynamic clipping feature fully supports the use of duplication, mirroring, and transformations as applied to datasets, regions, and surfaces. For example, an angled clip line that crosses the symmetry line of a mirrored dataset is not mirrored on the reflected side. Instead, it continues straight as intended, unaffected by the dataset mirroring.

The sequence of operations becomes important when transforming surfaces and applying dynamic clipping. Note that a surface that is transformed before clipping a dataset may be vastly different than transforming the same surface after clipping a dataset. Similarly, use caution when performing transformations on surfaces with dynamic clipping turned off; turning on a clip group after transforming surfaces could produce undesired clipping effects.

Integrating with Dynamic Clipping On

Values obtained from integration are affected by dynamic clipping. For example, the area value of a boundary surface obtained from integration is altered after the surface is clipped and the integration is repeated.

Note: While Dynamic Clipping and Duplication are both active, all visible parts of duplicated data are included when obtaining integrations, but Int *(S * Nx), Int (S * Ny), Int (S * Nz)* and *Int (V dot N)* results will reflect values for the additional area in its original position with the original normal orientation. For this reason, the integration result will not be zero in the direction of mirroring.
FV prints *WARNING: Integral includes all copies*

Exporting Visualization Objects with Dynamic Clipping On

For visualization objects which can be exported using *Tools. Export..* on the main menu, clipping will be preserved in the output data for that object, with the exception of object types *streamlines* and *particle paths.*

Restart Files

FieldView allows users to save clip groups in the form of a restart file. Clip groups restart files have the extension `.dyn` and are included in complete restarts.

Flipbook Build Mode

A discussion on the operation and features of the Flipbook Build Mode has been provided in **Flipbook Animation**.

2D to 3D Flat Surface Extrusion

This feature will extrude any binary FV-UNS (FieldView Unstructured) 2D dataset, including (but not limited to) data produced by FLUENT, CFD++ and NPARC/WIND, in a direction normal to the plane of data. This will work for any such dataset which has been stored with results in either the XY, XZ or YZ planes.

To enable this extrusion behavior, set the environment variable:

```
FV_2D_TO_3D=1
```


Use the environment variable,
FV_2D_TO_3D=1
to extrude a 2D dataset.

This will let you
  Create 2D plots
  Point probe on the surface

**Extrusion thickness is exaggerated in
this picture using Dataset Scaling**
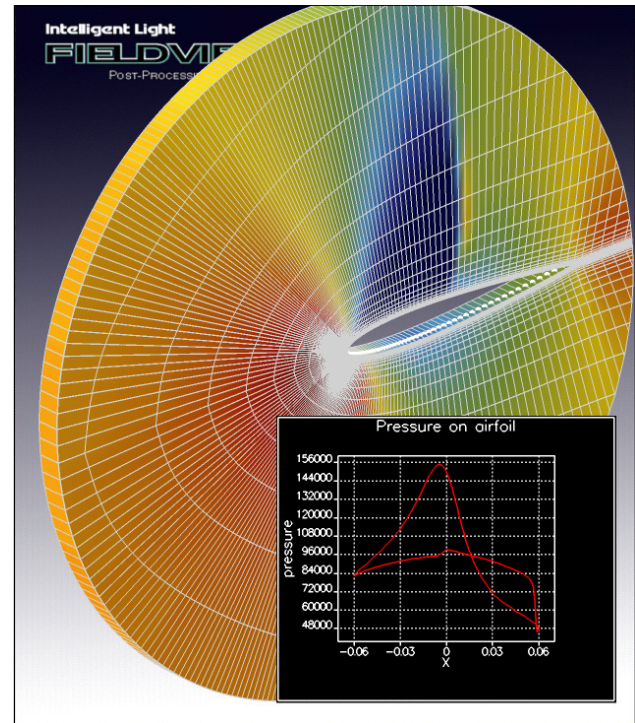
Figure 272 Extrusion of a 2D Dataset