NOĒSYS

Transform User's Guide and Reference Manual

RESEARCH SYSTEMS
Software ≡ Vision™

Transform Version 3.4
May, 1999 Edition

Restricted Rights Notice

The IDL® software program and the accompanying procedures, functions, and documentation described herein are sold under license agreement. Their use, duplication, and disclosure are subject to the restrictions stated in the license agreement. Research Systems, Inc. reserves the right to make changes to this document at any time and without notice.

Limitation of Warranty

Research Systems, Inc. makes no warranties, either express or implied, as to any matter not expressly set forth in the license agreement, including without limitation the condition of the software, merchantability, or fitness for any particular purpose.

Research Systems, Inc. shall not be liable for any direct, consequential, or other damages suffered by the Licensee or any others resulting from use of the IDL software package or its documentation.

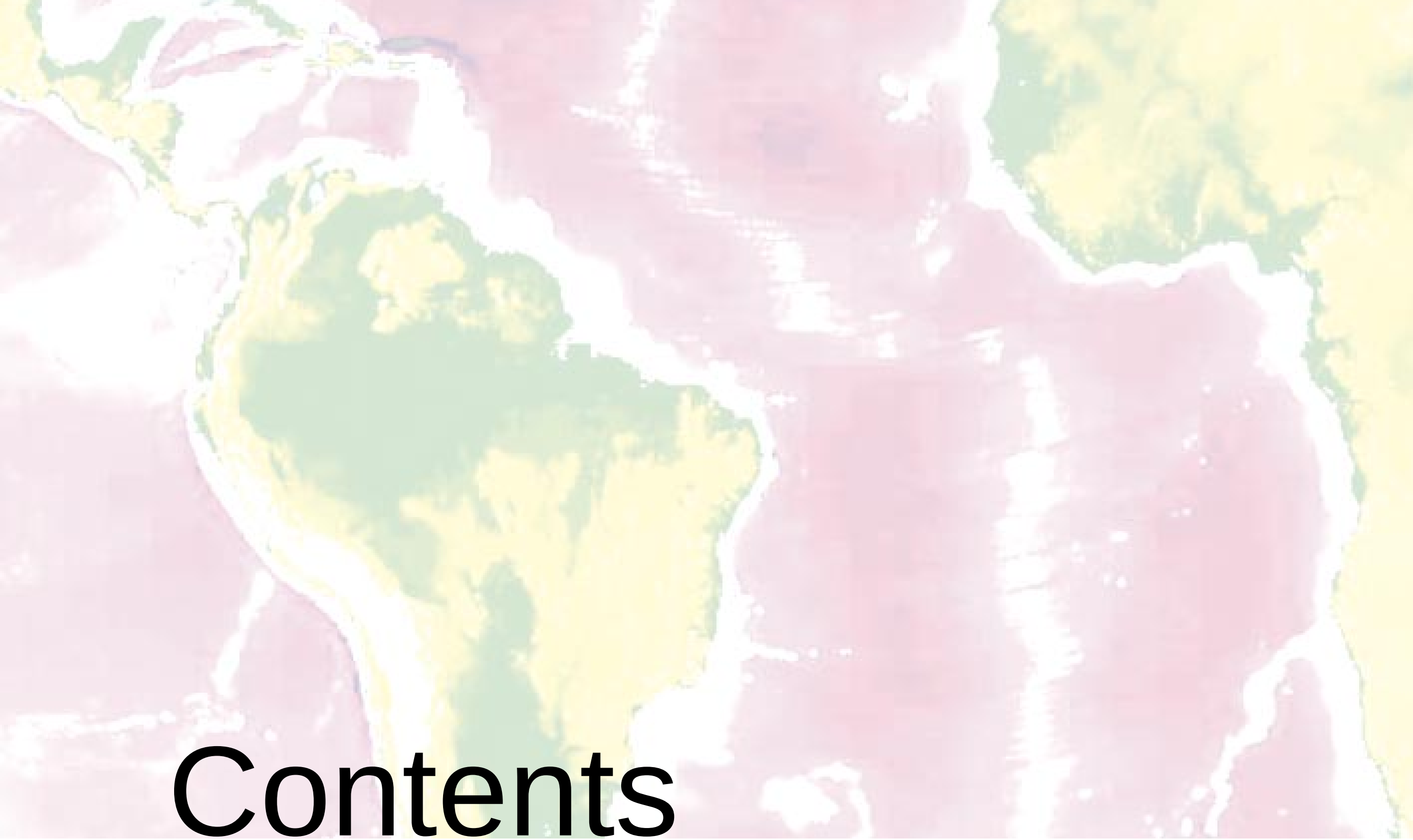
Permission to Reproduce this Manual

If you are a licensed user of this product, Research Systems, Inc. grants you a limited, nontransferable license to reproduce this particular document provided such copies are for your use only and are not sold or distributed to third parties. All such copies must contain the title page and this notice page in their entirety.

Acknowledgments

Fortner Software and its logo are trademarks of Fortner Software LLC

Transform, Noesys, Plot,T3D and the HDF Browser are trademarks of Fortner Software LLC



Contents

Part II: Tours

Part III: Reference Manual

Part I: Introduction

Chapter 1:
Welcome

Welcome to Transform, a powerful, scientific visualization software application that helps you graphically view and analyze large amounts of data. From color raster images to contour plots, Transform lets you interact with your data to gain insight into it. You can also use Transform to create publication-quality images directly from your data.

With Transform, you can analyze and present data in dataset windows or in any of seven graphical display types:

- Color Raster Images
- Interpolated Raster Images
- Line Graphs
- Contour Plots
- Surface Plots
- Vector Plots

- Histogram Plots (Windows only)
- Polar Plots (Macintosh only)

Transform maintains a constant relationship between all types of graphs on the screen so you get multiple, coordinated views of your data. Transform was designed to help you analyze two-dimensional data, that is, data with two independent variables (such as latitude and longitude) and one dependent variable (such as temperature or precipitation).

Transform is a highly interactive program with most features only a mouse click or keystroke away. Within Transform, however, is a powerful macro-scripting language that gives you explicit control over Transform functions, making it easy to automate repetitive tasks, such as importing data, manipulating arrays, creating plots, adding labels, and exporting images and data. Transform minimizes the amount of programming necessary to create custom macros by allowing you to convert existing images into macros that may be directly applied to other datasets.

All of Fortner Software's data manipulation and visualization applications use the Hierarchical Data Format (HDF), a public-domain, platform independent standard format for the storage of scientific data, images, and auxiliary information. It is an object-oriented, binary format capable of storing in one file many different kinds of data objects. Transform uses different HDF object types to store data arrays, notes about the data, images, and other plots. For more information about the HDF format, see Appendix D: HDF Reference.

Transform automatically reads two-dimensional HDF datasets, HDF images, ASCII Special, TIFF, FITS, GIF, PICT (Macintosh only) and BMP (Windows only). With only minor manipulation, it also reads MATLAB 4.0 files. In addition, Transform can import matrix and column data stored as ordinary ASCII text and as raw binary data.

Transform can open 2D or 3D datasets and images stored in HDF files that were created in Fortner Software's Noesys, a technical data processing application that lets you access, analyze, manipulate, visualize and organize large amounts of multidimensional data quickly and conveniently.

In addition, macro commands can be sent to Transform from Mathematica, making it possible to turn Transform into a visual data analysis post-processor. For more information about using Transform with Mathematica® see Appendix B: MathLink to Mathematica

Transform Lets You...

- Import data from virtually any source
- Display two-dimensional arrays in a spreadsheet-like format
- Scale and generate labeled color raster images from the data
- Generate labeled interpolated images from the data
- Create line graphs from any row or column of the data
- Generate labeled contour, surface, vector and histogram (Windows only) plots from the data
- Overlay one type of plot onto another type of plot
- Re-sample data to a different array size
- Fill missing data points with interpolated values
- Find the data values that correspond to points on an image
- Create and save formulas, comments, and notations with the data to which they apply
- Apply calculations from internal functions to data in order to generate derived datasets
- Synchronize points among multiple datasets that have the same dimensions
- Copy rows or columns of data, together with their axis labels
- Export data and graphics

Getting Started with Transform

This section describes how to install and upgrade Transform.

Installing Transform

Transform is installed with Noesys from the Noesys CD-ROM. For more information, see *Installation Guide for Windows and Macintosh* included with your Noesys CD-ROM.

Upgrading Transform

When upgrading Transform, an additional step is recommended. In addition to performing the normal installation described above, we recommend that you import the preferences settings and custom macros from your previous version of Transform. Importing the preferences settings and custom macros will enable you to upgrade Transform while retaining this information.

If you decide not to import the preferences settings and custom macros, the new version of Transform will use the factory default settings.

Windows - Upgrade Preferences from Fortner Software Transform

To import these settings, start Transform 3.4. Once Transform 3.4 is running, follow the procedures below:

1. Select **Preferences-Settings...** from the Edit menu.
2. Select **Import...** from the “Preferences Settings” window.
3. In the Import Preferences window, change the directory to where the previous version of Transform can be found.
4. If the previous version of Transform is from Fortner Software, change the Files of Type pop-up menu to display .frl files. Select the file “Prefs.frl”.
5. Select **Open**.

After step 5, you will be prompted if you would like to import your custom macros. By selecting “Yes”, all of your custom macros from the previous version of Transform will be transferred to the new version of Transform.

Windows - Upgrade Preferences from Spyglass Transform

The preference file for storing custom macros has changed with the upgraded versions of Transform. The old preferences file is called 'Prefs.spy' and it stores custom macros from Spyglass Transform. The new preference file is called 'Prefs.frl' and is located in the 'Trnsfrm' directory.

If you have custom macros in Spyglass Transform, open Spyglass Transform, select **Edit Macros...** from the Macros menu, highlight your macro and select the **Export...** command. Export your macros to text files with a .txt extension. These text files can then be imported into Fortner Software's Transform 3.4, by selecting **Edit Macros...** from the Macros menu, then choosing the **Import...** button from the Edit Macros dialog.

Power Macintosh - Upgrade Preferences from Fortner Software Transform

Transform 3.3 stores its preferences and macros in the "Transform Prefs" file located in the "Hard Drive:System:Preferences:Fortner Research" folder.

Transform 3.4 also stores its preferences and macros in the "Transform Prefs" file which is now located in the "Hard Drive:System:Preferences: Fortner Software" folder. To preserve your old preferences or custom macros you can simply copy the "Transform Prefs" file from the "Fortner Research" folder to the "Fortner Software" folder before you launch Transform 3.4. If you do not have a "Fortner Software" folder you can create one or you can launch Transform 3.4, then copy the old file into the folder, replacing the one that is there.

Power Macintosh - Upgrade Preferences from Spyglass Transform

The preference file for storing custom macros has changed with the upgraded versions of Transform. The old preferences file is called ‘Spyglass Settings’ and it stores custom macros from Spyglass Transform. The new preference file is called ‘Transform Prefs’ and is located in the 'Fortner Software' folder inside the 'System:Preferences' folder.

If you have custom macros in Spyglass Transform, open Spyglass Transform, select **Edit Macros...** from the Macros menu, highlight your macro and select the **Export...** command. Export your macros to text files with a .text extension. These text files can then be imported into Fortner Software's Transform 3.4, by selecting **Edit Macros...** from the Macros menu, then clicking the **Import...** button from the Edit Macros dialog.

About this Manual

The *Transform User's Guide and Reference Manual* presents information in four parts:

Part 1, Introduction, Chapters 1-2

Part 2, Tours, Chapters 3-8

Part 3, Reference Manual, Chapters 9-24

Part 4, Appendices, A-H

Part 1, Introduction, describes Transform's capabilities, how to use the manual, how to install Transform, how to register, and how to contact technical support representatives.

Part 2, Tours, shows you how to use Transform. You should be able to complete the tour in an hour or two. Keep in mind that the tour provides only an introduction to how the program operates. If you would like more detailed information on specific features, please refer to the appropriate chapters in the reference and appendix chapters. After completing the tour, you should be able to proceed with your own data analysis on at least a basic level.

Part 3, Reference Manual, focuses on four main concepts:

- The data window and how to manipulate its settings
- Data formats and how to import files
- Image windows and how to create images from data
- Methods for preparing data and using macros

The Reference Manual details ways in which you can manipulate Transform to meet your needs. It also describes how to better analyze your data to take fullest advantage of Transform's capabilities.

Part 4, Appendices, further develop previously discussed ideas or introduce the interaction of Transform with other available software.

Chapter 2:

Using Transform

The Tour and Reference Manual use examples based on the following sample data files, which are provided on the CD. These files can be found in the ‘Transform\Samples’ folder.

Sample Data

This section describes the sample data used in the Tours. The files and directories described in this section are stored in the ‘Transform\Samples’ folder.

Monthly Temperatures

All the data in the ‘Monthly’ folder was obtained from the U.S. Weather Service. The files contain a 12-column by 40-row array listing average monthly temperatures in Fahrenheit from 1936 to 1975. Included are four files:

- **sprngfld.txt**. This array shows the average monthly temperatures for Springfield, Illinois.
- **Avg_S.txt**. This array also contains a 12-column by 40-row array, but every row is identical, containing the monthly temperatures for Springfield averaged over the 40-year timespan.
- **chicago.txt**. This array shows the average monthly temperatures for Chicago, Illinois.
- **Avg_C.txt**. This array also contains a 12-column by 40-row array, but every row is identical, containing the monthly temperatures for Chicago averaged over the 40-year timespan.

Thunderstorm

The ‘Tstorm’ folder includes four HDF files containing a slice of a three-dimensional, numerical simulation of a thunderstorm. The slice is a horizontal cut through the base of the storm (near ground level). The grid is 75 km on an edge. The xvel.hdf, yvel.hdf, and zvel.hdf files are the X, Y, and Z components, respectively, of wind speed on a horizontal slice from the thunderstorm simulation. The vort.hdf file represents the wind vorticity on the same slice. Authors: Dr. Louis J. Wicker and Prof. Robert B. Wilhelmson, National Center for Supercomputing Applications.

U.S. Weather

The ‘Weather’ folder contains an ASCII text file (weather.txt) of weather information from reporting stations across the United States on January 2, 1991. The first three lines consist of descriptive information including the names of the data columns. The data has 694 entries, one per line. Each line has the location of a reporting station in horizontal and vertical coordinates and 5 recorded values from that site. If no data of a particular type was available for a location, the value for that data type is entered as

-99. The columns are in order: horizontal map coordinate, vertical map coordinate, temperature (F), dewpoint (F), pressure (mb), East-West wind speed (m/s), and North-South wind speed (m/s).

- weather.txt. This is an ASCII Text column file of U.S. Weather.
- temp.hdf. A filled, interpolated version of the temperature data from weather.txt and includes contour and U.S. Map overlays.
- us_map.hdf. Consists of a blank dataset with an overlay of a continental US map.

Hydrogen Orbital (Macintosh only)

The file 'hydrogen.hdf' is in the 'Misc' folder. It contains a dataset and a polar image of a hydrogen orbital. The data represents a two dimensional slice through a particular wave function of an electron in a hydrogen atom. The selected wave function is one of the 3D orbitals, where (n=3, l=2, m=0). The general equation used is

```
r^2 sin( ) cos( ) exp(-r/c)
```

where r is a radius, is the angle in radians, and c is a constant. This translates to the following Transform expression:

```
Orbital=y(a)*y(a)*sin(x(a))*cos(x(a))*exp(-y(a)/20)
```

where a is a two dimensional array in transform with x (column scales) going from 0 to 2 , and y (row scales) going from 1 to 100.

M83 Radio Map

The file 'm83.hdf' is in the 'Astro' folder. This example, developed by S. Sukumar and R.J. Allen at the University of Illinois at Urbana-Champaign and the Space Telescope Science Institute, is the visualization of a map showing the distribution of polarized radio continuum emission around the spiral galaxy M83. The emissions were observed with the very large array radio telescope at a wave length of 20 cm. The 'Purple Haze' color table is used to bring out features in this dataset.

Neutron Star

The 'neutron.hdf' file is in the 'Astro' folder. This file shows the results of a one-dimensional time-dependent gas dynamic simulation of the accretion of gas onto a neutron star. The variable displayed here is density (in dimensionless units) as a function of radius (in units of the neutron star radius) and time (in seconds). At any particular time in the simulation, the radial density profile would be a vertical slice through the dataset.

Note the initial enhancement in density at large radii (this would be in the upper left hand corner of the image). Notice how this enhancement moves down over time, towards the neutron star surface. When this enhancement reaches the bottom of the grid the neutron star luminosity will increase. When the luminosity reduces to it's normal value, the density enhancement will again move towards the neutron star surface, starting the cycle over. The simulations were done on a Cray XMP at the National Center for Supercomputing Applications at the University of Illinois. Authors: Brand Fortner and Frederick Lamb, Department of Physics, University of Illinois at Urbana-Champaign; Guy Miller, Los Alamos National Laboratories.

San Andreas Fault

The ‘fault.hdf’ file is in the ‘Misc’ folder. This is an image from a dataset representing the calculated horizontal shear stress on a vertical slice across the San Andreas fault zone. The calculations are from a finite element model, which simulates the accumulation and release of strain on a strike slip fault that cuts through an elastic viscoelastic earth model.

The vertical scale is somewhat exaggerated in the displayed image; the vertical range of depths is 0 - 15 km, while the lateral width of the image is 200 km. The range of shear stress magnitudes displayed by the image palette is 0 -130 MPa (0 - 1300 bars). The original model data were generated using a geophysical finite element code called ‘visco’, running on the Ames Research Center/NASA Cray Y-MP supercomputer. Authors: Gregory A. Lyzenga, Jet Propulsion Laboratory; Arthur Raefsky, Stanford University.

Understanding Window Organization

Transform consists of several types of windows and tools. The figure below shows the menu bar and four types of windows that are commonly used in Transform: Data window, Notebook, and two image windows (interpolated image and a surface plot). How the Transform application and its windows are organized is described below.

Title Bar

In Windows, the Title Bar bears the name of the application in the window, the Minimize, Maximize, and Close buttons, and the Transform Control-menu box. The application window can be reduced to an icon by clicking in the Minimize button and can be enlarged to fill the screen by clicking on the Maximize button. Single-clicking the Control-menu box will raise a menu. Double-clicking will quit the application.

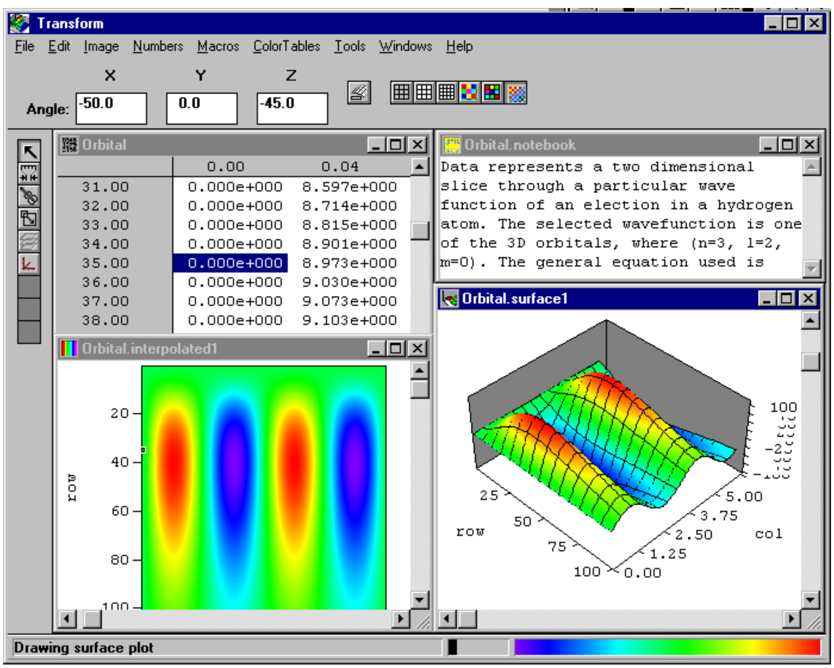


Figure 2-1: Transform Session

In Power Macintosh, the Title bar bears the name of the dataset in all windows, and contains zoom and close buttons.

Menu Bar

The Menu bar displays up to eight (Power Macintosh) or nine (Windows) menu names. The Color Tables menu is unavailable with certain types of graphs and plots. Clicking a menu name displays commands on the menu. Selecting a command carries out an action.

Tool Bar

The Tool bar displays information and/or provides text fields for changing information about the data, and lists buttons that, when clicked, carry out an action. Toolbar buttons perform the same functions as commands by the same name in the Tools menu (Windows only). When clicked, a button's function is specified in the text portion of the Status bar (Windows only). The type of buttons available in the Toolbar is controlled by the tool selected from the Tool Palette and the active window type.

Tool Palette

When a file is open, the tool palette provides a number of buttons. Clicking on a button invokes a list of buttons in the toolbar and allows actions to be carried out in the dataset or image windows. The tool palette in Windows can be horizontal, vertical, floating, or hidden.

Image Window

The image window displays one of the several graphical display types that are generated from the Image menu.

Dataset Window

The dataset window is a spreadsheet-like display of the data.

Notebook

The Notebook is used to store comments or enter and execute macros. Both comments and macros can be saved with the HDF file.

Status Bar (Windows)

The Status bar includes a Text Bar, Thermometer, and Color Bar. The Text Bar provides a brief description for an action about to be invoked. It also displays program status during long computations. The Thermometer is a graphical display of the progress of a long computation. For windows that use color, the Color Bar shows the current window's color table.

Icon (Windows)

When you click an image or dataset window's minimize button, the window is reduced to an icon and placed at the bottom of Transform's frame window. Each window type has a unique icon. Double click on the icon to enlarge the window again.

Part II: Tours

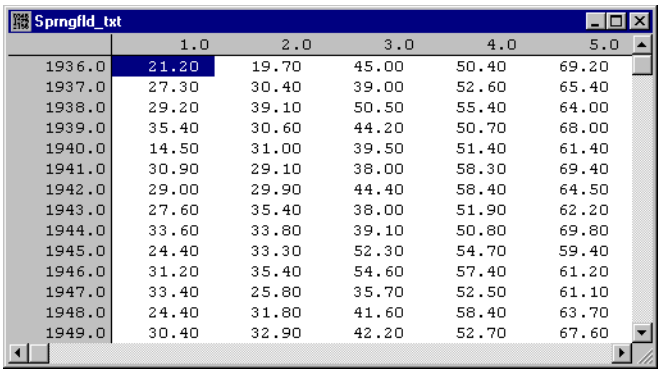
Chapter 3:

Viewing Your Data

To start the Transform program, double-click the Transform icon in the Transform program group or select it off the Start menu (Windows). If this is your first time in Transform, the start-up screen will prompt you for your name, organization, and registration number. This information is stored in the About Transform dialog.

Opening a Dataset

Choose **Open...** from the File menu and open the file ‘sprngfld.txt’ in the ‘Samples\Monthly’ folder. This ASCII file is in a format called ASCII Special that is automatically recognized by Transform. The sprngfld.txt file contains a 12-column by 40-row array of the average monthly temperatures in Springfield, Illinois between the years 1936 and 1975.


Sprngfld_txt

	1.0	2.0	3.0	4.0	5.0
1936.0	21.20	19.70	45.00	50.40	69.20
1937.0	27.30	30.40	39.00	52.60	65.40
1938.0	29.20	39.10	50.50	55.40	64.00
1939.0	35.40	30.60	44.20	50.70	68.00
1940.0	14.50	31.00	39.50	51.40	61.40
1941.0	30.90	29.10	38.00	58.30	69.40
1942.0	29.00	29.90	44.40	58.40	64.50
1943.0	27.60	35.40	38.00	51.90	62.20
1944.0	33.60	33.80	39.10	50.80	69.80
1945.0	24.40	33.30	52.30	54.70	59.40
1946.0	31.20	35.40	54.60	57.40	61.20
1947.0	33.40	25.80	35.70	52.50	61.10
1948.0	24.40	31.80	41.60	58.40	63.70
1949.0	30.40	32.90	42.20	52.70	67.60

Figure 3-1: sprngfld_txt Data Window

The Dataset Window

The data window that opens, ‘sprngfld_txt’, has a name slightly different from the text file name, ‘sprngfld.txt’. In Transform, all non-alphanumeric characters in file names are replaced with underscores. In the data window, numerical row labels show years and numerical column labels show months. If some of the data is not visible, scroll around the window using the scroll bars.

Changing Numeric Display Fonts

Transform for Windows displays values in floating-point format (e.g., 54.23) by default. Power Macintosh displays values in exponential format (e.g., 2.120e+01). The display format is controlled by the **Attributes...** command from the Numbers menu. Choose **Attributes...** and the Attributes dialog shown in Figure 3-2 will appear.

Note

In Power Macintosh, you can also double-click a data value to open a similar Edit Data dialog.

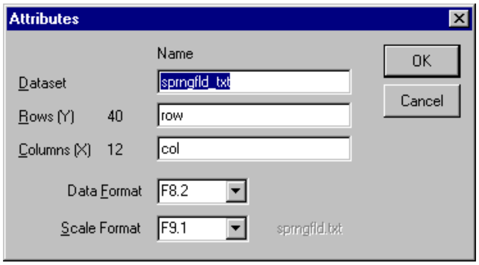


Figure 3-2: Attributes Dialog

The Attributes dialog has five components: the dataset name, row name, column name, data format, and the scale format. The Edit Data dialog lets you change a data value or change the format of the data display.

The Data Format (Array Data) pop-up menu specifies the format of numbers displayed in the dataset window. The Scale Format (Row/Col Labels) pop-up specifies the format of numbers for row and column data scales in the dataset window. Fortran 77 format specifications are used for numbers.

In the Attributes/Edit Data dialog, change the Data Format to **F8.2** and the Scale Format to **F5.0**. Click **OK** to close the dialog.

The dataset window displays in floating-point format and scales as integers.

Generating Images

It is difficult to gain insight into the data simply by looking at numbers. Transform helps you understand your data using images, line graphs, contour plots, vector plots, surface plots, polar plots (Macintosh only) and histograms (Windows only). Choose **Generate Image** from the Image menu to generate a simple image of the Springfield dataset. If necessary, expand the window to see the entire image. The default color table in Transform is 'Rainbow'. Your image should look similar to Figure 3-3.

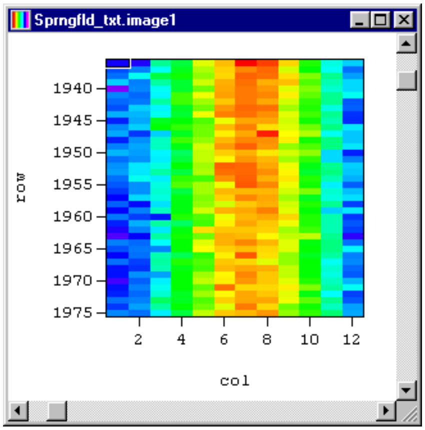


Figure 3-3: Raster Image of sprngfld_txt

Interactive Selection

Now compare your dataset window with the image window. Each number in the sample dataset is displayed in the new image window as a rectangular block of color. Click on any part of the image, and note that a data value in the dataset window is selected. The pixel in the image corresponds to the value highlighted in the data window.

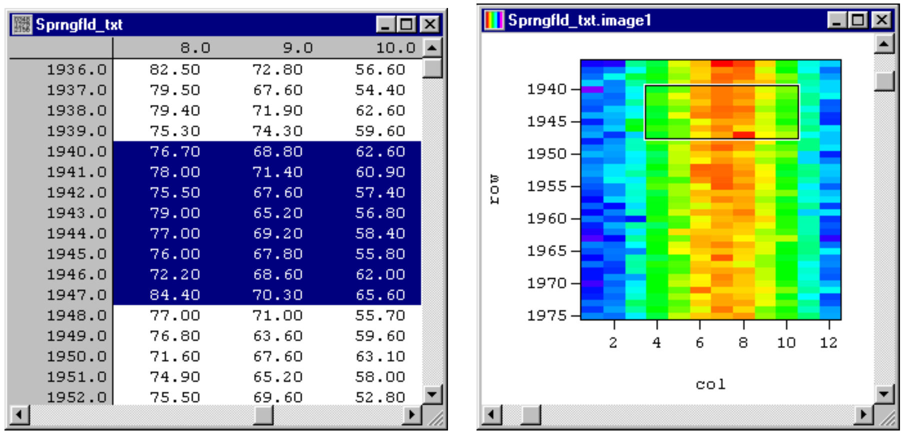


Figure 3-4: Data window and image window

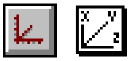
Note that high values in the image are shown as red pixel blocks, and low values are shown as blue pixel blocks. You can see the relationship more clearly by looking at the color bar at the bottom of the image. This shows the mapping of data values to colors.

Image Tools

Transform has nine primary operating modes, represented by tools in the tool palette: Select, MinMax, Fiddle, Resize, Overlay, Axis, Label, Contour and Vector. Each tool provides a different selection of buttons, dialogs and controls.

You can select a tool by clicking on its button in the tool palette or by selecting it by name from the Tools menu (Windows only). Each tool has a toolbar configuration that becomes active when the tool is selected. In addition, the Windows status bar can display a small message about each button to help you determine what that button does. Click on a button and hold the mouse down while reading the message from the status bar. If you do not want to perform the button's action, drag the mouse away from the button before releasing the mouse.

Axes and Labels



By default, raster images are generated with labels, which can be modified using the Axis tool. To do so, first click on the **Axis** tool in the tool palette. The toolbar now displays options that let you configure how the column, row, color bar axes and image title are displayed. Power Macintosh also provides the ability to change Axes font and size from the toolbar.

Click the Row Axis button (**Row...)** button to open the Row Axis/Axis Labels dialog.

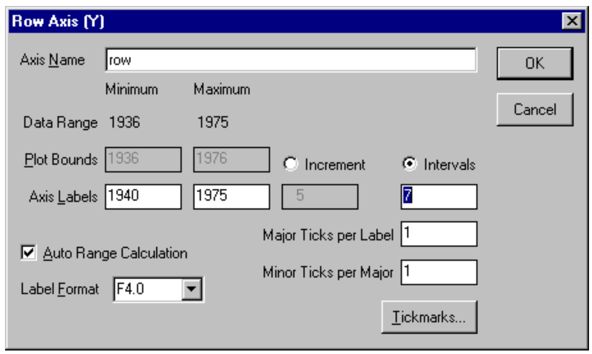
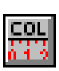


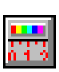
Figure 3-5: Row Axis Dialog

In the dialog, type 'Year' as the axis name, then type in numbers to specify axis label boundaries: **1940** for the minimum and **1970** for the maximum. Next click the **Increment** radio button (in Power Macintosh, click the left side of the double-arrow) and, in the box below it, change the value to 10. Now click in any text field and note that the value for Intervals changes. (Alternatively, you can change the Intervals value and the Increment value will adjust accordingly.)

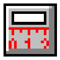
In the Minor Ticks per Major field, change the value from 1 to **2**. Scroll through the choices for Label Format, and select **F5.0**. Click **OK** to see the changes you made to your image axis labels.



Now click the Column Axis (**Col...)** button and type 'Month' as the axis name. Also type in Axis Labels of **1** and **12**, Increments of **1**, Label Format of **F5.0**, and Minor Ticks per Major of **1**. Click **OK** to return.



Next click on the Color Bar (**Bar...)** button and type 'Temperature' as the axis name. Type in Axis Labels of **20** and **80**, Increments of **10**, Label Format as **F2.0**, and Minor Ticks per Major of **1**. Click **OK** to return.



Finally, click the Image Title (**Title**) button and in the text field type 'Springfield, IL Temperatures'. Click **OK** to return. Your image should look similar to that in Figure 3-6.

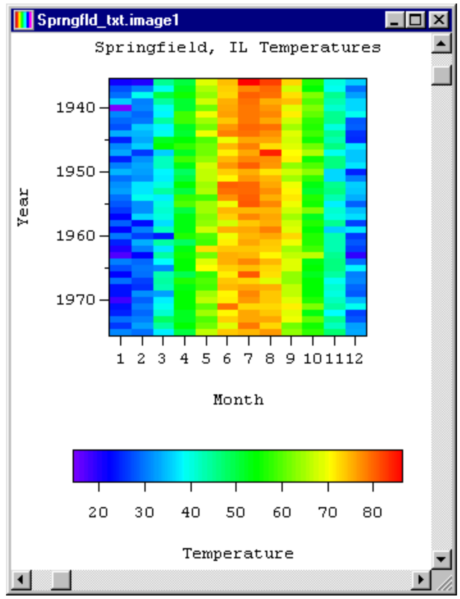


Figure 3-6: Finished Raster Image with Axis Labels Changed

Generating Interpolated Images

You can generate an interpolated image from the same dataset by selecting **Interpolated Image** from the Image menu. Bilinear interpolation is used to calculate the transition between the blocks of pixels, creating a smoother image.

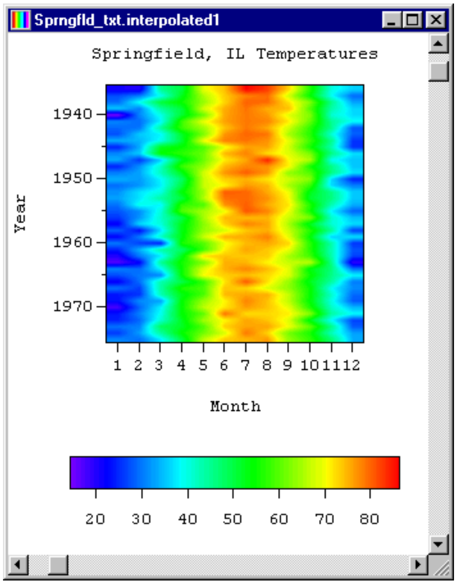


Figure 3-7: Interpolated Image using sprngfld_txt

Manipulating Color

Many of Transform's images and plots use color to represent data values, bringing out detail in the data.

In raster images, each pixel in the image is a color mapped from a data value in the array. Surface plots can use color to show the data values from one array combined with a surface formed from the data in another array. Contour plots can be drawn in color to reinforce the data value represented by each contour line. Dataset windows, notebooks, line plots, and vector plots contain no color.

Color Tables Menu

The easiest way to change the mapping of data values to colors is to select a new color table. Each table in the Color Tables menu is designed to bring out different types of features from the image. Select one of the image windows that contain color, and choose **Hot Metal** from the Color Tables menu. Select different color tables from the Color Tables menu to see how data values map to color changes. When done, select the **Rainbow** color table so your images match those shown in this chapter.

Fiddle Tool

Fiddle is so named because it lets you play with color mapping while looking for interesting characteristics in your image. The Fiddle tool compresses and shifts the color range, allowing obscured data to become more visible. With an image window active, click the **Fiddle** button in the tool palette. Manipulate the color table by holding down the mouse button and moving the cursor around in the image window. Expand and compress the color table by moving the mouse up and down. Move the region of color compression left and right by moving the mouse left and right.

The figure below shows a pair of images before and after applying Fiddle. The image on the left was generated using the 'Rainbow' color table. The color table for the image on the right was compressed using the Fiddle tool.

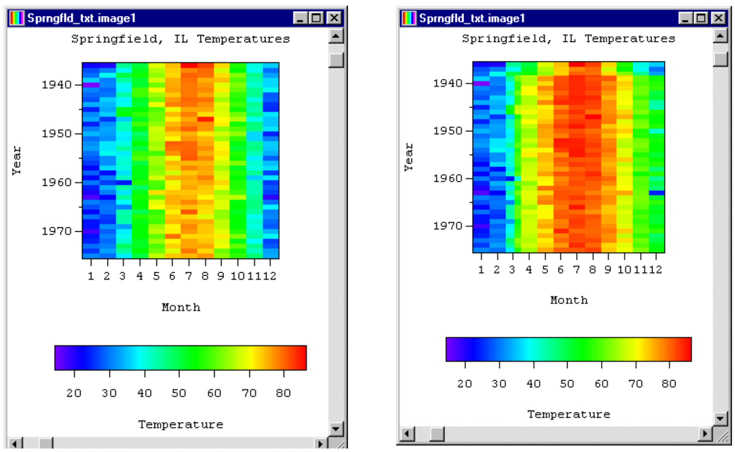


Figure 3-8: Using the Fiddle tool

In the Fiddle toolbar, you can change to any of the fiddle modes or restore the color table to its previous unfiddled state. The Rotate mode shifts the colors left and right without compressing the color range. It does not provide additional contrast, but it can reveal important details in an image. Click the **Rotate** button (or select **Rotate Colors** from the Power Macintosh pop-up menu) button in the toolbar and use left and right mouse movements to move the color table left and right. When you are done, click the Restore (**Restore)** button or select **Rainbow** from the Color Tables menu to return to the default color table.

You can also change the mapping of data values to colors using the Min/Max tool. This is described in the next section.

Choosing a Data Range

In Windows, minimize, but do not close, your 'sprngfld_txt' windows. Now open the sample file, 'chicago.txt' from the ‘Samples\Monthly’ folder and the 'chicago_txt' dataset will appear. Use the **Attributes**... command (Windows) or double-click a value (Power Macintosh) to change the data format to F8.2 and the scale format to F6.0. Select **Generate Image** to generate a raster image of the data, then select **GrayScale** from the Color Tables menu. You will see an image similar to the one shown below.

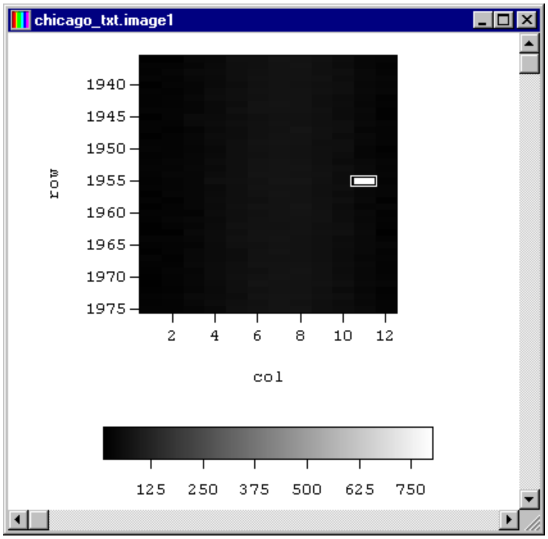


Figure 3-9: Raster Image of Chicago Temperatures

What went wrong? The entire image is black except for one white pixel block. Select the white pixel and note that the corresponding data value for Column 11, Row 1955 is 800.

Most likely, the temperature in Chicago was not 800 degrees in November. The correct value is 36.0. This problem can be solved in several ways.

Create a Histogram (Windows)

You can get a visual impression of the data scaling problem with a Histogram plot. Choose **Histogram** from the Image menu to create a new window with the new plot. Your plot should look like Figure 3-10.

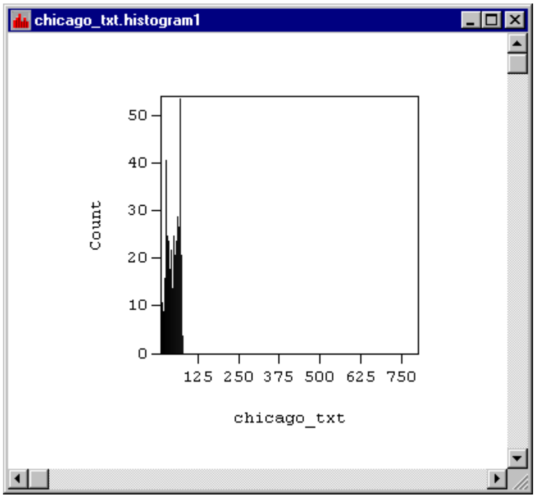


Figure 3-10: Histogram of Chicago Temperatures

Transform has picked the data range to include all of the data. It goes from the smallest number, 13.3, to the largest number, 800. The problem is, all of the data values that you want to see are clustered among the lowest 20 colors in the data range, and they are all black!

Later we will change the data value from 800. First, let us see how to change the color scaling without changing any of the data.

Changing the Data Min/Max (Windows)

Activate the **MinMax** tool for your histogram window. In the toolbar enter **14.5** for the minimum value and **86.2** for the maximum value. This range was chosen to match the min/max range used to generate the Springfield image. When you press Enter, the histogram redraws, showing you the temperature distribution in detail. To see this distribution in color, select **Rainbow** from the Color Tables menu.

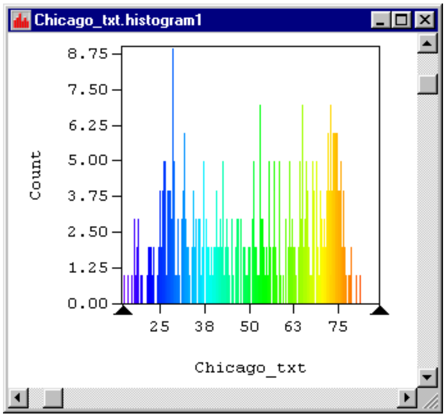
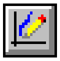


Figure 3-11: Histogram with New Data Range

In histograms, the outliers (data values outside the selected range, such as the 800 value) are omitted. They fall off of the edges of the plot. In the toolbar, you have a count of the number of these outliers so that you know they are there. You should see a "1" in the box for outliers that are higher than the given data range. In images, outliers are assigned their own special colors, described below.

Updating the Rest of the Images (Windows)

After finding the correct data range using a histogram, you need to transfer these values to other windows, especially the raster image displaying most of its colors in black. You could switch to that window and use the MinMax tool again to enter the same data range and then repeat the process for each window. Instead, click the



Update All Windows button to change the data range for every window associated with the dataset—all in one step. Now select the **Rainbow** color table from the Color Tables menu. The Chicago image, shown in Figure 3-14, should look similar to the Springfield image. In particular, since they were generated with the same min/max values, you can compare the colors in the two images directly.

Change the Data Min/Max (Power Macintosh)

Click the **Min/Max** icon in the image window and click the **Color Bar...** button in the image toolbar to open the following colorbar window.

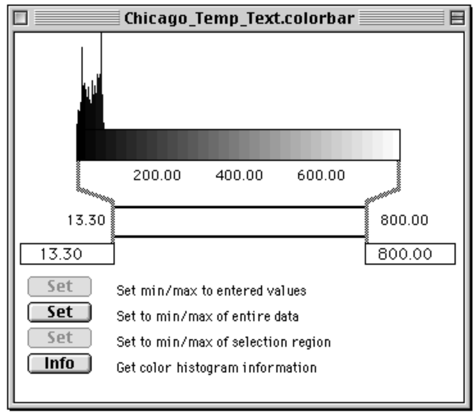


Figure 3-12: Colorbar Window

Replace the minimum value (13.3) with 14.5 for the minimum value and the maximum value (800) with 86.2. then click the **Set** button next to 'Set min/max to entered values' button,

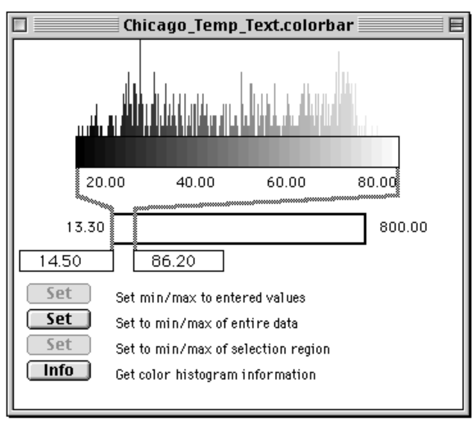


Figure 3-13: Colorbar Window with Modified Range

When ready, click the close box. Make the Chicago image active then click the **Replot** button in the toolbar. Now select **Rainbow** from the Color Tables menu. The Chicago image should look similar to the Springfield image as shown in the figure below.

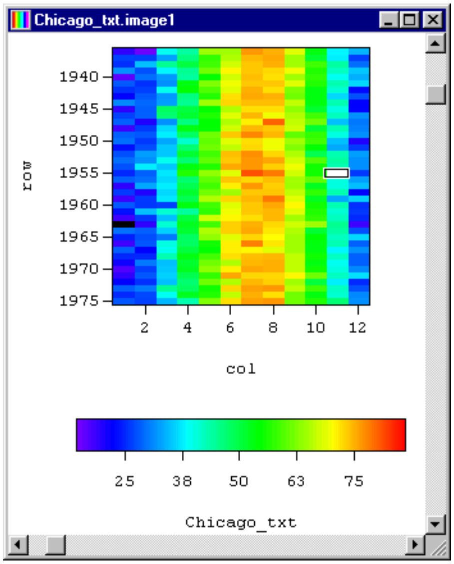


Figure 3-14: Chicago Temperatures with New Data Range

Edit Outlier Colors (Windows)

The incorrect temperature value of 800 is now shown as a white block in the image. No matter which color table you select from the Color Tables menu, this value always displays in white. We can make it appear less obvious by changing the color used to display outliers. Select the Chicago image and, under the **MinMax** tool, click on the **Edit Outlier Colors** button to bring up the dialog shown in Figure 3-15.

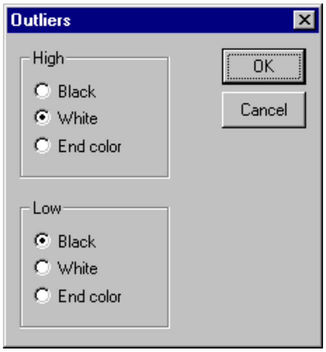


Figure 3-15: Edit Outlier Colors dialog

The 800 value is a High outlier because it is larger than the maximum value of the data range. As we saw in the image, it is set to White. Change the setting to **End Color** and click **OK**. If necessary, click the **Update All Windows** button to transfer the change to the other image windows for this dataset. Now the bad data value is displayed in the highest color of the current color table. Select several color tables to see the effect.

Change the Data Value

Of course, we could just replace the bad data value. This works whenever your dataset is small enough for you to edit one number at a time and you know the correct value. Click on the image at the spot representing the bad data value. The dataset window highlights the corresponding data location.

In Windows, activate the dataset window and enter the correct value, **36.0**, in the Data Value box in the toolbar and press Enter. In Power Macintosh, double-click the **800** value in the spreadsheet to open the Edit Data dialog. Replace the 800 value with **36** and click **OK**. When you generate new images or cause the old ones to redraw, the new value is used.

Note

Using the Fiddle tool is not a good method for dealing with data outliers, because Transform assigns 254 colors uniformly between the data minimum and data maximum. In the example above, most of the colors would have been assigned to the useless region between the true data maximum (86.2), and the bad 800 value.

Managing Multiple Windows

Select **Open...** from the File menu to automatically import the files 'avg_ c.txt' and 'avg_s.txt'. In each data window, select the **Attributes...** command from the Numbers menu (or in Power Macintosh, double-click a value to open the Edit Data dialog) and change the Data Format to F8.2 and Scale Format to F6.0.

Next, choose **Tile Windows** from the Windows menu. Then select **Synchronize** from the Edit menu.

Your screen should now look like the figure shown below, depending on the size of the screen.

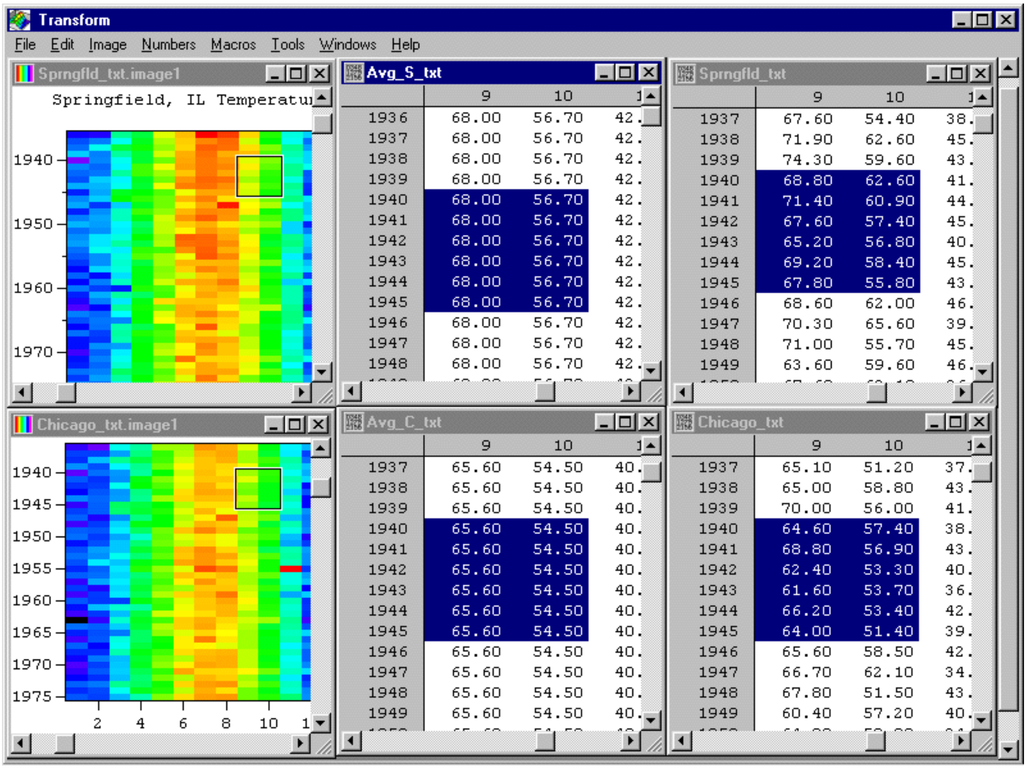


Figure 3-16: Images Synchronized

Selecting **Synchronize** means that every time you click on an image or a data window, the corresponding region of every data window and image from the different datasets will be selected. Note that although temperatures in Chicago and Springfield are similar, Springfield is usually slightly warmer. We will verify this in the next chapter.

Note

Synchronize only works if the datasets have the same array size and the images have the Select tool active.

Another way to manage windows is to click the Minimize button in each window (Windows only). The minimized windows reduce to icons. Double-clicking an icon with the mouse will cause that window to be enlarged. In Power Macintosh, hidden windows are not displayed, but their names are listed in italics under the Windows menu. Selecting a hidden window from the Windows menu will cause that window to be displayed.

Chapter 4:

The Notebook and Calculations

The Notebook window allows you to enter text notes and store them with your data. You can also enter algebraic and macro expressions.

Text in the Notebook

With the 'sprngfld_txt' dataset open, activate the dataset window and select **See Notebook** from the Numbers menu. Then enter some descriptive text such as that shown in Figure 4-1.

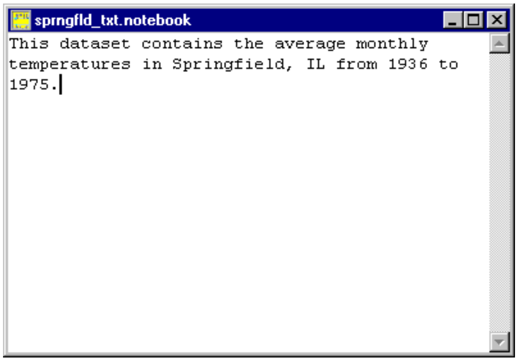


Figure 4-1: Notebook Window with Text Entered

The text is stored in the Notebook, and will appear whenever you reopen the Notebook window with **See Notebook**.

Before proceeding, close the Notebook.

Calculations in the Notebook

You can also execute calculations and macros in the notebook.

Example 1 - Difference Calculation

With the 'sprngfld_txt' and 'chicago_txt' datasets open, and the 'sprngfld_txt' dataset window active, again select **See Notebook** from the Numbers menu. Note that the text you entered in the Notebook window earlier is still there. On a new line, enter the following expression to calculate the difference between temperatures in Springfield and Chicago on corresponding dates:

```
difference = sprngfld_txt - chicago_txt
```

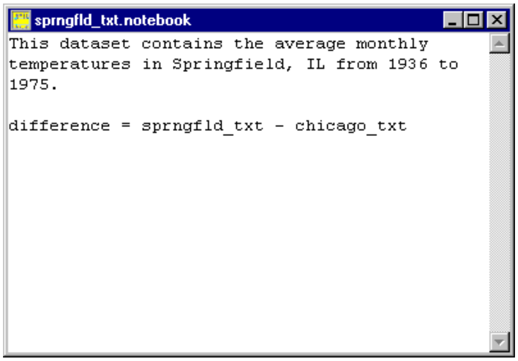


Figure 4-2: Notebook Window with New Text Entered

Note

This example assumes the bad data value in the Chicago dataset for November, 1955 has been changed from 800 to 36. See Chapter 3 for more information on changing this value.

Make sure the cursor is in the same line as the expression, then select **Calculate From Notes** from the Numbers menu. A new dataset window will appear. Now select **Generate Image** from the Image menu. Your new window should look similar to that shown in Figure 4-3.

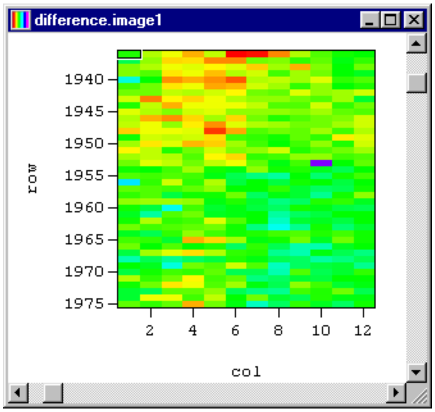


Figure 4-3: New image generated

The new dataset called ‘difference’ is equal to the difference between the temperatures measured at Springfield and those at Chicago for the same months and years.

Note that Springfield was warmer than Chicago in the 30’s through the 50’s, but is nearly equal to Chicago in the 60’s and 70’s.

Example 2 - Deviation Calculation

Return to the Notebook window and enter this on a new line:

```
Sdif = Sprngfld_txt - Avg_S_txt
```

Note

To avoid typing errors, select the dataset name(s) from the Datasets pop-up menu in the Toolbar (Windows only) or in the Notebook (Power Macintosh only). To place the dataset name in the Notebook at the location of the cursor, in Windows select the dataset name that you want, then press Enter; in Power Macintosh, simply select it off the pop-up menu.

Make sure the cursor stays on the same line as the expression and select **Calculate From Notes** from the Numbers menu.

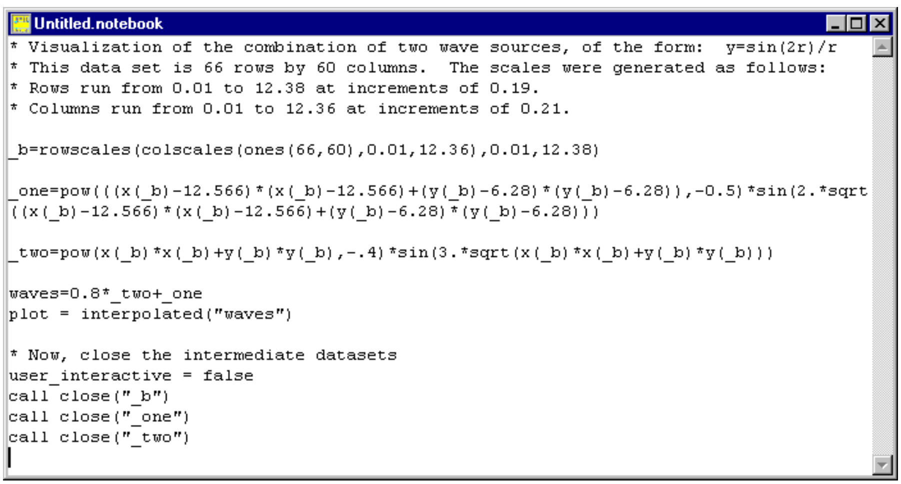
The dataset ‘avg_s.txt’ consists of a 12-element row of numbers copied 40 times. The numbers correspond to the monthly temperatures for Springfield, IL averaged over the entire time period. The dataset ‘Sdif’ is therefore equal to the monthly deviation from this average.

Other Possibilities

You could do the same calculation for Chicago and compare the deviations. It might also be interesting to subtract the two deviations to see the difference in deviations between the two sites. Or perhaps compare the two averaged datasets, or convert the data to an absolute temperature scale.

Example 3 - Wave Function (Windows only)

This advanced example visualizes an analytical function, a combination of two wave sources, of the form y=sin(2r)/r. To try this example, type the lines listed below into your Notebook window, exactly as they appear. You may omit, however, the comment lines, which are preceded by an '*'. Because Transform is shipped with this macro, you may also simply select **MakeWaves** from the Macros menu.



Untitled.notebook

```
* Visualization of the combination of two wave sources, of the form:  y=sin(2r)/r
* This data set is 66 rows by 60 columns.  The scales were generated as follows:
* Rows run from 0.01 to 12.38 at increments of 0.19.
* Columns run from 0.01 to 12.36 at increments of 0.21.

_b=rowscales(colscales(ones(66,60),0.01,12.36),0.01,12.38)

_one=pow(((x(_b)-12.566)*(x(_b)-12.566)+(y(_b)-6.28)*(y(_b)-6.28)),-0.5)*sin(2.*sqrt
((x(_b)-12.566)*(x(_b)-12.566)+(y(_b)-6.28)*(y(_b)-6.28)))

_two=pow(x(_b)*x(_b)+y(_b)*y(_b),-.4)*sin(3.*sqrt(x(_b)*x(_b)+y(_b)*y(_b)))

waves=0.8*_two+_one
plot = interpolated("waves")

* Now, close the intermediate datasets
user_interactive = false
call close("_b")
call close("_one")
call close("_two")
```

Figure 4-4: Notebook with Wave Function Macro Entered

Once you've entered these lines, place the cursor at the beginning of line one of the expression, then drag to the remainder. Then select **Calculate From Notes** from the Numbers menu.

This creates a 66 x 60 dataset, where row scales run from 0.01 to 12.38 in steps of 0.19, and column scales run from 0.01 to 12.36 in steps of 0.21. Several intermediate datasets are used to create the final 'waves' dataset. An interpolated image is created from this dataset, and the intermediate datasets are closed.

The 'waves' dataset window and corresponding interpolated image window, similar to those shown in Figure 4.5, should appear on your screen.

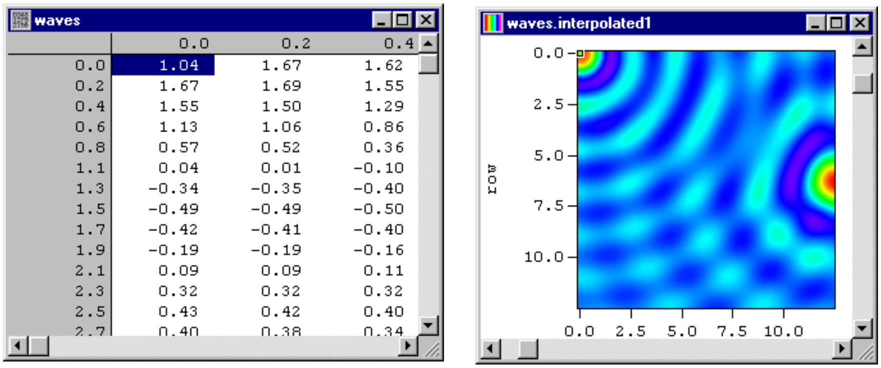


Figure 4-5: New dataset and image created from macro expression

You can enter just about any algebraic expression in the notebook window, using the dataset names as your variables. There are also numerous functions such as sin(q) and exp(q) available. See Chapters 22 and 23 for more detail.

Before continuing to the next chapter, close all windows except the Springfield dataset and image window.

Chapter 5:

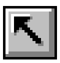
Line Graphs

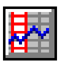
In Transform, line graphs are useful for displaying profiles through a two-dimensional array. For this chapter, we again use the 'sprngfld.txt' file.

Generating Line Graphs

Minimize the interpolated image window, but keep the simple image and the dataset window visible. Select **Line Graph** from the Image menu to create a line graph. Now select **Line Graph** again to generate a second line graph. Then, resize or move your windows so that all four are clearly visible.

Horizontal and Vertical Line Graphs





You should now have two identical line graphs, both representing a row of data values from the dataset array. Click one of the line graphs and, with the **Select** tool active, click the **Vertical** (radio) button. This plots data along a column instead of along a row. Your line graphs should now look like those in Figure 5-1.

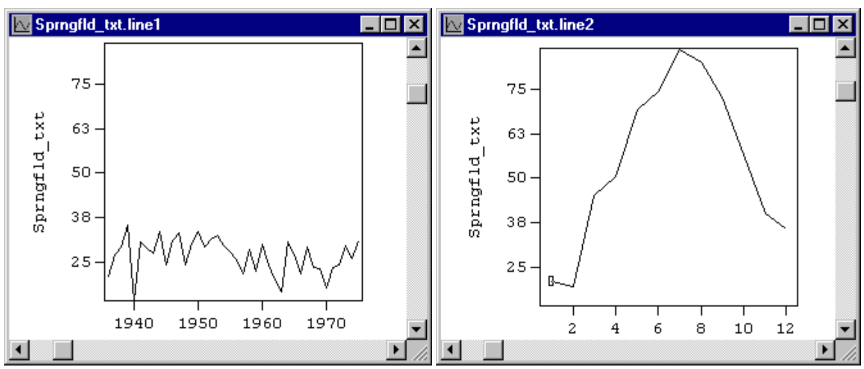
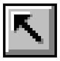


Figure 5-1: Vertical and horizontal line graphs

The line graphs in Figure 5-1 are for the row '1936' and the 'January' column. The vertical axis of both graphs represents the temperature. The horizontal axis shows years in the row plot (left) and months in the column plot (right). You can click on a value in the dataset window or a rectangle in the image window to plot different row and column graphs. To change the format of axis labels and titles, click on the Axis tool and proceed as you did in Chapter 3.

Synchronizing Windows



For each of the four windows, activate the **Select** tool (Macintosh only has tools available in the image windows). Click anywhere in the data window, and you will see a horizontal and vertical profile through the data at that point.

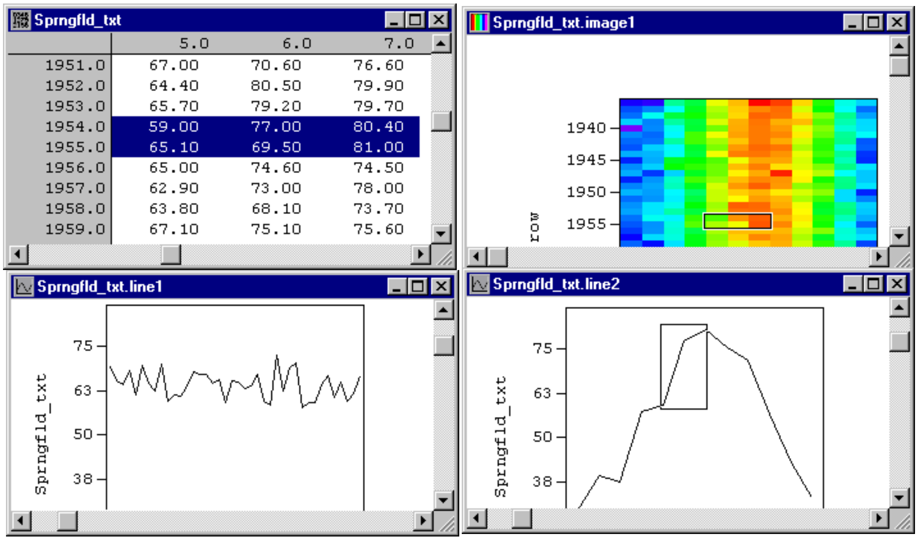
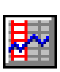


Figure 5-2: Example of Selected Areas Highlighted

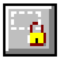
The corresponding region in both line graphs and the raster image is selected. You can also see the value of a particular point in a line graph by clicking on part of the line graph. The corresponding data value will be highlighted in the other windows.

Comparing Line Graphs

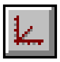
Note that when you select Column 7, the difference between the highest and lowest monthly temperature in July (Column 7) is shown in the Selection Min/Max to be about 15 degrees, but the difference between the highest and lowest monthly temperature in March (Column 3) is almost 30 degrees.

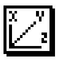


To see this comparison graphically, activate the horizontal line graph window, then click the **Vertical** (radio) button.

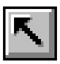


In Windows, click the padlock button on the toolbar. In Power Macintosh, click the **Locked** checkbox in one of the line graph windows to freeze that line graph. Now select Column 7 (July) using the arrow keys or the data window. You can compare line graph profiles by using **Copy** and **Paste Overlay**.





Select the unlocked line graph window and click the **Axes** tool. Deselect the **Show Axes** button (or checkbox) to turn off the axes, then select **Copy** from the Edit menu. Click the **Show Axes** button/checkbox again to turn the axes back on.



Now click the **Select** tool in that window, and use the arrow keys to select Column 3 again. Now select **Paste Overlay** from the Edit menu, and your window should look like the figure below.

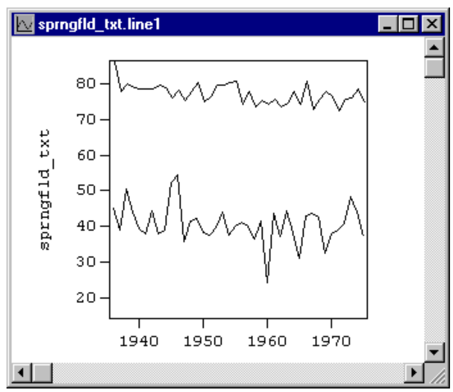


Figure 5-3: Overlaying July Temperature Line Graph on March Temperature Line Graph

The window consists of a live line graph, here displaying Column 3, with a graphic pasted on top of the line graph from Column 7. Notice that the spikes in the March and July line graphs only slightly correspond with one another.

The ability to compare line graphs as overlays is a good example of how Transform helps to lend insight into your data.

Chapter 6:

Contour and Vector Plots

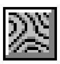
This chapter introduces the use of contour and vector plots through two examples. The first example shows you how to generate a contour plot of Xvel and overlay it on a color interpolated image of the same dataset. The second example explains how to generate a vector plot and to create a composite plot by overlaying vector and contour plots on a color interpolated image.

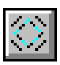
Contour Plots

Close all the files you have open by selecting the **Close All** command. Then open the following files in the 'Samples\Tstorm' directory: 'vort.hdf', 'xvel.hdf', 'yvel.hdf', and 'zvel.hdf'. Now select **Interpolated Image** for the Zvel and Vorticity datasets.

Activate the 'Vorticity' dataset window and select **Contour Plot** from the Image menu. In Windows, the contour plot will be generated automatically, in Power Macintosh the Contours dialog (the Windows equivalent of this dialog is called “Contour Levels”) will appear. Power Macintosh users may proceed to the next section “Contour Levels/Contours Dialog.”

How to Make Contour Plots in Transform for Windows





Maximize the contour plot that is displayed in the new image window. Now click on the **Contour tool** in the Tool Palette. Four toolbar options will be displayed: Dashes, Smoothed Lines, Color, and Contour Levels. Click on the **Dashes button** to display all contours below 0 as dashed lines. Your image should look like the figure shown below.

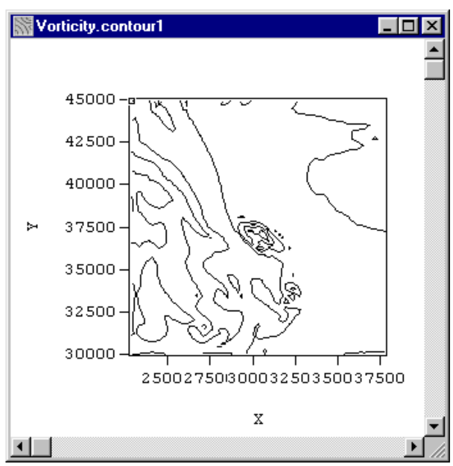


Figure 6-1: Contour Plot

Contour Levels/Contours Dialog

In Windows, click the **Contour Levels button** to activate the Contour Levels dialog, as shown on the left side in Figure 6-2. The Contours dialog shown on the right side of the figure is displayed when Contour Plot is selected from the Image menu in Power Macintosh.

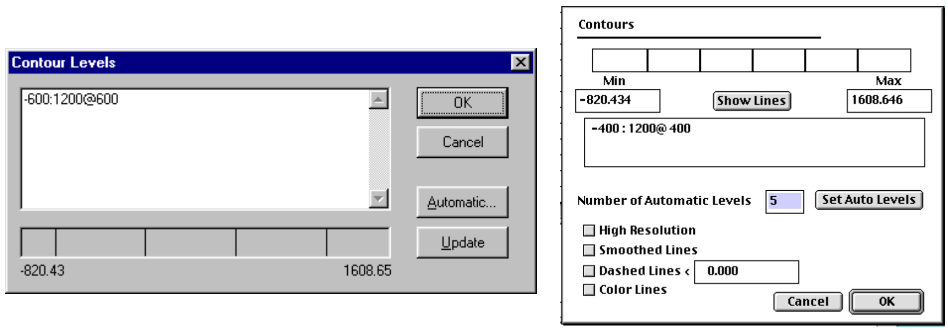
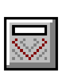


Figure 6-2: Contour Levels dialog

The Windows Contour Levels dialog opens with a default contour specification like -600:1200@600 which means "draw contour lines from -600 to 1200 in increments of 600". Across the bottom of the dialog is a line display of the specification. The line display indicates the minimum and maximum values in the dataset and where the contour lines will be drawn.

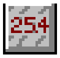
The Power Macintosh Contours dialog opens with a default contour specification of -400:1200@400 which means "draw contour lines from -400 to 1200 in increments of 400." The box at the top of the window shows where the contour lines will be drawn given the current minimum and maximum values.

Change the line in the Contour Levels/Contours text box to read **-850:1600@500**. Click the **Update button** (Windows) or **Show Lines button** (Power Macintosh) to see your changes to the text box recorded on the line display. In Power Macintosh, click **Dashed Lines** checkbox to display all contours below 0 as dashed lines. Click **OK** to draw the lines.



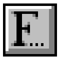
Note you can also draw contours interactively. Simply click on the **Contours button**, then click on the plot at the position you want the contours to be drawn. Add a few more contours by clicking on the plot, then click on the **Contour Levels button** (Windows) or select **Contour...** from the toolbar (Power Macintosh) to open the Contour Levels/Contours dialog again. You will see that the new contours you created are now specified in the text box, with each new contour level separated by commas. You can add new levels separated by commas, or you can delete a level simply by editing it out of the list.Delete the extra levels you added interactively and click **OK**.

Contour Label Tool



Your contour plot is now complete except for labels. Click the **Label** tool, then click on any of the contours to add labels to the contour plot interactively. (To delete a label, click on it again.) The larger your contour window, the more clearly the contours will be delineated.

In Windows, the default number format and the Font button are displayed in the toolbar. In Power Macintosh, the default Font, Size and Format are contained in pop-up menus. From the Format menu, change the number format to **F6.0**.



In Windows, click the **Font button** to invoke the Font dialog. Change the character type to **Times New Roman** and the size to **8**. In Power Macintosh, make these changes using the pop-up menus. Your image should look like the figure below.

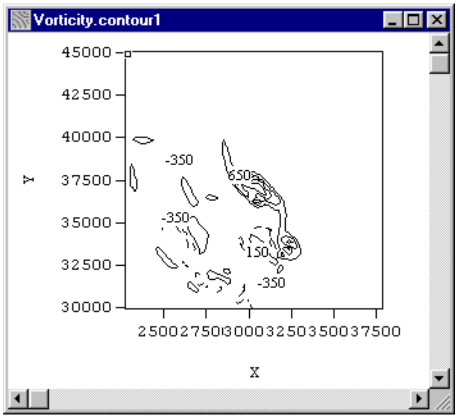
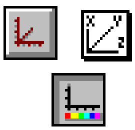


Figure 6-3: Contour Plot with Labels

Making an Overlay



The next step is to overlay the contour plot on the image. Before proceeding, hide the axes on the contour plot by first activating the **Axis tool**, then clicking the **Show/Hide Axes** button (Windows) or the **Show Axes** checkbox (Power Macintosh).

Now, select **Copy** from the Edit menu; activate the 'Vorticity.interpolated" window and select **Paste** (Windows) or **Paste Overlay** (Power Macintosh). Your image should look like that in Figure 6-4.

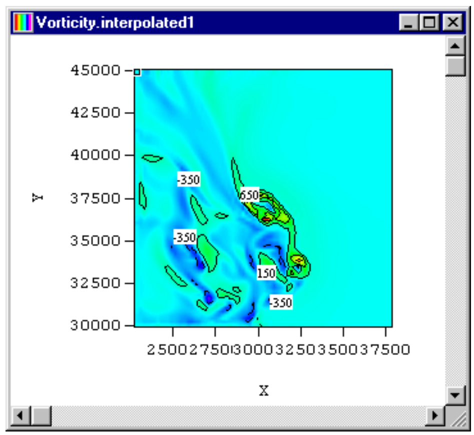
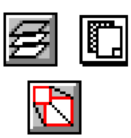


Figure 6-4: Overlay of Vorticity Contour Plot on Vorticity Interpolated Image

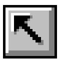


If the size of the contour plot in the overlay does not match that of the interpolated image, click on the **Overlay tool** in the tool palette, then click the **Size to Match button** in the toolbar.

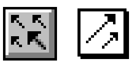
Before continuing, close the Vorticity.interpolated1 image, but leave the Vorticity.contour1 image window open. You are going to use this image later in this chapter. If you wish, minimize this window.

Vector Plots

In this section, we first explain how to generate a vector plot of the X and Y components of wind speed. We then explain how to overlay a contour plot of wind vorticity and the X/Y vector plot on a color interpolated image of the Z component of wind speed.



With the 'Xvel' dataset active, select **Vector Plot** from the Image menu. When the **Select tool** is active, the horizontal component is set to 'Xvel' and the vertical component is set to one of the other open datasets. For this example change the vertical component to be **Vertical (V) = Yvel**.



You can modify the appearance of the vectors. Click the **Vector tool**, and in the toolbar, change the number of vectors across from 22 to **15** (‘# Across: 15’) and the scale from 1.0 to **1.5** (‘Scale: 1.5’). In Windows, press **Enter**; in Power Macintosh, press the **Replot** button.Your plot should look similar to like that in Figure 6-5.

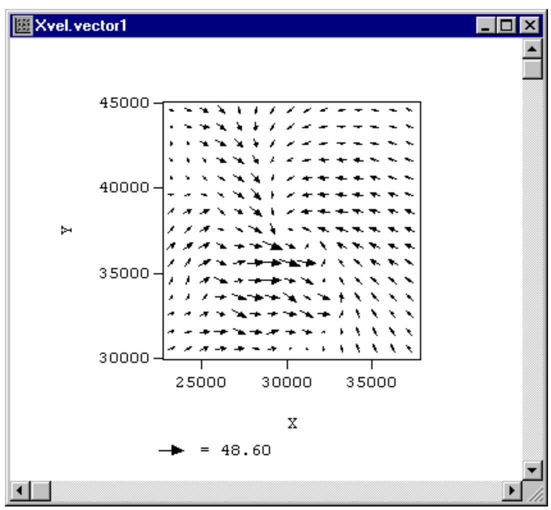


Figure 6-5: Finished Vector Plot

Combining Plots (Windows)

Minimize all dataset windows. Bring up the 'Zvel.interpolated1' image by selecting it from the Windows menu.

Select **Tile** from the Windows menu. You should have three windows displayed: 'Vorticity.contour1', 'Zvel.interpolated1', and 'Xvel.vector1'.

Now you are ready to build a composite image. Now activate the 'Xvel.vector1' window, turn off the axes, select **Copy** from the Edit menu, make the ‘Zvel.interpolated1’image active and select the **Paste** command from the Edit menu to paste it to the 'Zvel.interpolated1' image. Next, activate the 'Vorticity.contour1' window and select **Copy** from the Edit menu.

Now activate the 'Zvel.interpolated1' window and select **Paste** from the Edit menu. Your finished product should look like the image in Figure 6-7 on the next page.

Combining Plots (Power Macintosh)

Bring up the ‘Vorticity1.contour’ plot. Select **Copy** from the Edit menu. Make the ‘Xvel.vector1’ plot the active window and select the **Paste Overlay** command from the Edit menu. Your image should look like the figure below.

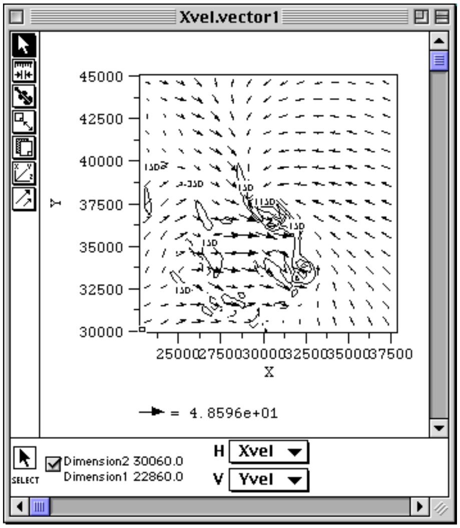
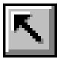


Figure 6-6: Combined Image

Minimize the dataset windows. Bring up the 'Zvel.interpolated1' image by selecting it from the Windows menu.

You should have three windows displayed: 'Vorticity.contour1', 'Zvel.interpolated1', and 'Xvel.vector1'.



Now you are ready to build a composite image. Activate the 'Xvel.vector1' window, turn off the axes, click the **Select tool**, then select **Copy** from the Edit menu.

Note

Power Macintosh: if you copy a contour plot with the Option key selected, when pasted your overlay will have white lines instead of black.

Now activate the 'Zvel.interpolated1' window and select **Paste Overlay** from the Edit menu. Your finished product should look like the image in Figure 6-7.

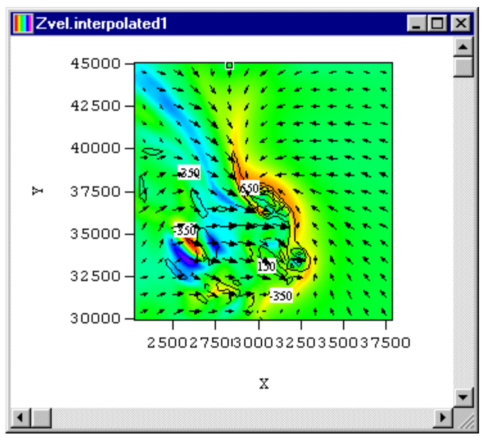
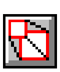


Figure 6-7: Overlay of Vorticity.contour1 and Xvel.vector1 on Zvel.interpolated1 Image

Again, if the size of the contour and vector plots do not match the color image exactly, select the **Overlay tool** then click the **Size to Match** button.



This will force the overlay to exactly match the current image.

The composite image that you just created is a visualization of four separate 2D data arrays (X, Y, Z velocity and vorticity)! You may want to select a different color table for this image to improve the contrast between the color image, contours and vectors.

Chapter 7:

Surface Plots and Image Macros

Surface plots are a three-dimensional representation of a two-dimensional dataset array. The height of each point on the surface is determined from the data values in the array.

For our surface plot examples, we again use 'Xvel.hdf'. So close all open files and re-open 'Xvel.hdf'.

Create a Surface Plot

With the 'Xvel' dataset window active, choose **Surface Plot** from the Image menu; a small, wireframe image like that in Figure 7-1 should appear. You may want to enlarge the window and move the image, as explained below.

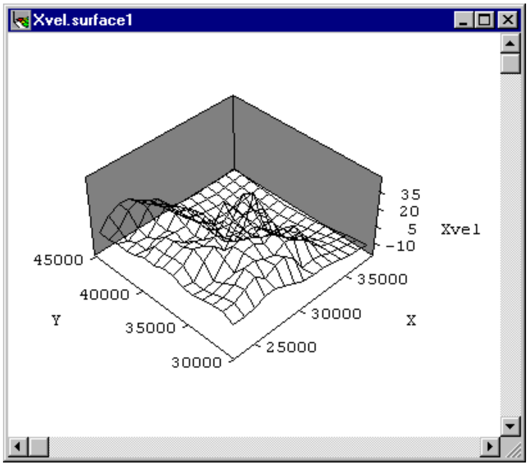
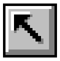


Figure 7-1: The X Component of Velocity

Image Angle, Position and Size



The tool bar offers a variety of options for translating, orienting, and sizing the image. Choose the **Select tool**, then click on the surface plot and hold down the mouse button (left mouse button on Windows). Move the mouse left and right to rotate the image on a horizontal plane and up and down to tilt the plot toward and away from you. While rotating the image, it changes to an orientation cube, as shown in Figure 7-2.

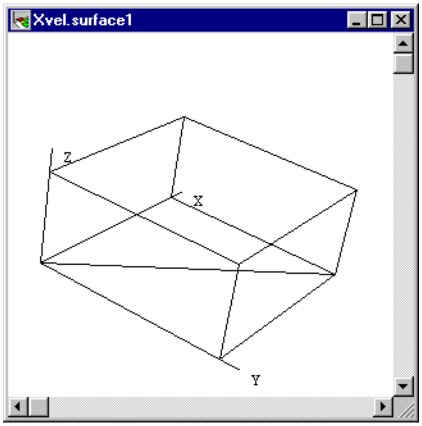
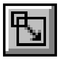


Figure 7-2: Orientation Cube

In Power Macintosh, you can hold down the Option key while dragging to reposition the mouse without spinning the plot.

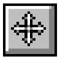
Notice that the X, Y, and Z angles are reflected in the toolbar. In Windows, you can also change the angles by typing new values in their respective text fields. In Power Macintosh, this is done by clicking the **Surface...** button then selecting **Edit Angles...**.

Sizing and Moving the Surface Plot (Windows)



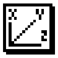
Next, make the **Resize tool** active. Instead of showing the angles of the cube, the toolbar now displays the size of the X, Y, and Z axes. Click on the surface plot, hold down the mouse button and move the mouse. The image again changes to a cube (as shown in Figure 7-2) and the mouse now resizes the plot.

The same actions with the Control (Ctrl) key depressed allow you to resize the surface in just X and Y coordinates. As you resize the image, the toolbar displays changes in the size of the X, Y, and Z axes.

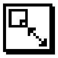


To move the surface plot on the page without rotating or resizing it, click the **Move Surface button** from the toolbar, then click the surface plot to move it on the page.

Sizing and Moving the Surface Plot (Power Macintosh)

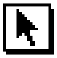


To size the surface plot interactively, first click on the **Axes tool**. Next click in the center of the image window and hold down the mouse button. To increase and decrease the size of the X and Y axes, move the mouse left and right. To increase and decrease the Z axis size, move the mouse up and down.



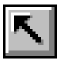
You may also need to use the Resize tool to increase the size of the image area that the surface plot is drawn on. Note that the **Fit to Window** button on the Axes tool resizes both the surface plot and the image area.

When you are done, release the mouse button and the surface plot will render in the new size that you selected. Note that when the Select tool is highlighted, mouse movements change the surface plot orientation. When the Axes tool is highlighted, mouse movements change the surface plot size.



Hold the Command key down while dragging the mouse to reposition the surface plot without spinning it.

Surface Parameters and Types (Windows)



Now go back to the **Select tool** and click on the **Surface Parameters...** button. The Surface Parameters dialog prompts you for information regarding the surface plot grid, plot parallax, and color variable. By default, grid lines are enabled for both rows and columns; you may disable either one but not both. Change the spacing for Rows to 2 and for Columns to 8; notice that the number for Count changes proportionally in relation to the dataset dimensions. When you are comfortable with the settings, click **OK**, and note the changes in the plot.

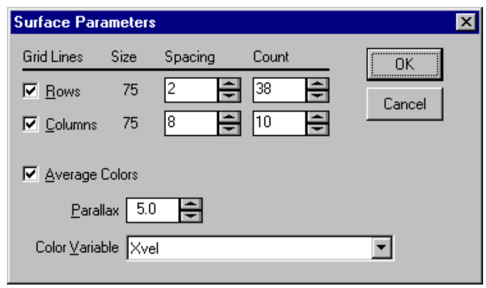
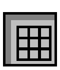


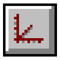
Figure 7-3: Surface Parameters Dialog

Surface Types



When you generate a surface plot, the Wireframe surface type is selected by default. This is because it generates faster than the other surfaces. If you select Hi-res Black & White or Hi-res Color Surface, the image takes longer to render when you move or make changes to the image. Note that if your current color table is set to GrayScale, you won't see color in the color surface types. Select **Rainbow** from the Color Tables menu. Now test the different surface types, but when you are finished, return to one with a low resolution (e.g., Wireframe).

Axis Tool



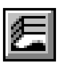
After moving, stretching, and rotating the surface plot, you will need to adjust the axes labels. Click on the **Axis tool**. The toolbar displays a number of options.

Figure 7-4: Axis Toolbar

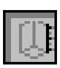
The first three buttons (from left to right) activate dialogs for labeling your row, column, and data axes. Click on the **Row** button and change the Label format to F6.0.

The fourth and fifth buttons control the color bar below the image. (The color bar buttons are inactive with wireframe and black and white surface types selected).

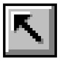
Surface Decorations

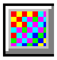


The last six buttons are specific to surface plots. Click on the **Surface Decorations** button and enable Skirt, Backstop, and Gridlines in the dialog that appears. Skirt has a value of 1, specifying the color to be the same as the first in the color table you chose from the Color Tables menu. Changing the skirt value from 1 to 256 changes the skirt to a special value for gray. Enter **0** to make the skirt white. The same explanation applies to Backstop; leave the Backstop at 256.



Use the last five buttons in the toolbar to specify the placement of axes labels. The first of these removes all the axes, leaving only the surface plot. The last four place the data axis in the origin, on the left, the right, and both the left and right, respectively. Click on the **Right Data Axes** button (second to the last button).





Now return to the **Select** tool in the Tool Palette, then select the **Hi-res Color Surface** button from the toolbar. Your surface plot should look similar to the one in Figure 7-5.

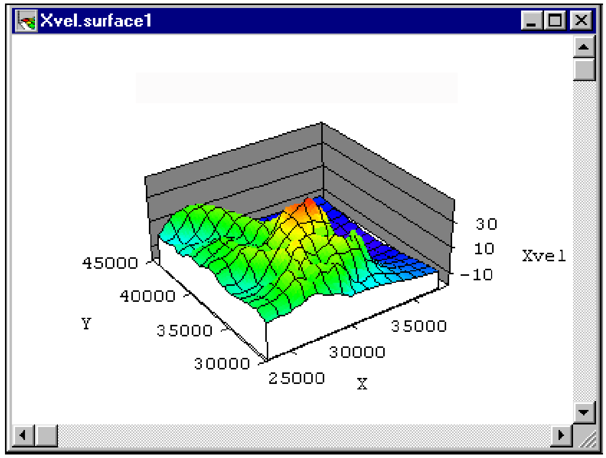
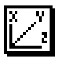


Figure 7-5: Finished Surface Plot using Xvel

Surface Parameters and Types (Macintosh)



Select the **Axes tool** and then click the **Labels...** button. Select **Right Data Axis** from the popup menu in the dialog. Also select **Surface Skirt** and give it color number **0** (white) and select **Backstop** and give it color number of **256**, a special value for gray. Finally, set **Grid Lines** on. Several of these options may already be selected, depending on how your Transform program is configured.

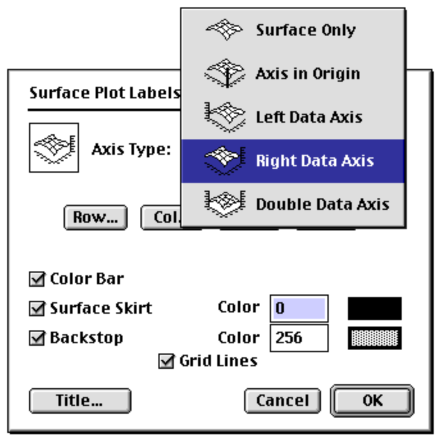


Figure 7-6: Surface Plot Labels Dialog

After making these changes, click on **Row...**. In the dialog, change the Label Format to **F5.0**.

Make whatever additional changes you wish to the label increment, label name, etc. When you are done, select the **Columns (X)** dialog via the popup menu, and make your changes to the Label Format, etc. there. Likewise, make your changes to the Data and Colorbar dialogs. When you are done, click **OK** to return.

Next, return to the **Select tool** and select **Hi-res Framed Color Surface** from the popup menu at the bottom of the image window.

Now that you have your surface plot in the size and orientation that you want, and have the labels exactly the way you want, the final step is to increase the number of lines in the surface plot. We save this step until the last because a high resolution surface plot takes much longer to render.

Click on the **Surface...** button, and change the 'Lines Every' fields for rows to 2 and columns to 8 in the surface dialog. Click on **OK** to return.

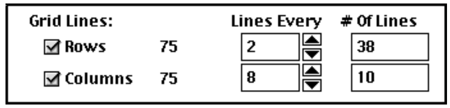


Figure 7-7: Surface Dialog

Depending on your choice of labels, size and orientation, your surface plot should now look similar to the following (we used the ‘Rainbow’ color table).

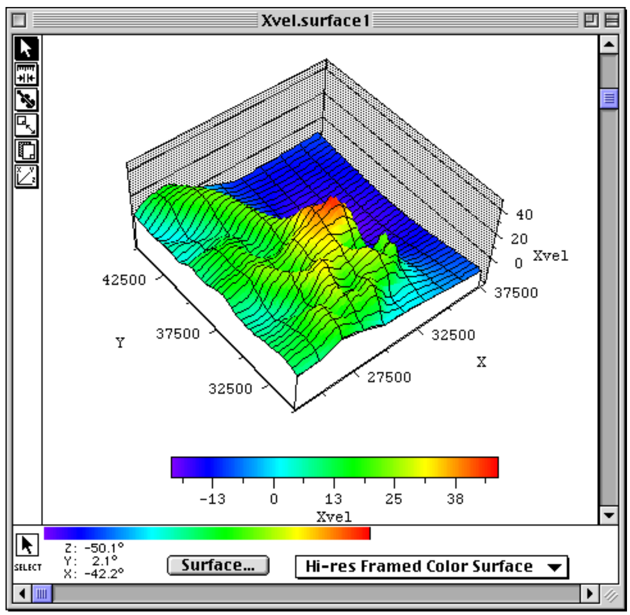


Figure 7-8: Completed Surface Plot

Create an Image Macro

Any image you create can be saved as a custom macro, which can then be applied to other datasets. To see how image macros work, we will create a macro for the surface plot you just created. We also use the file 'Yvel.hdf'.

Maximize the 'Xvel.surface1' window you just created. With the surface plot window active, select **Create Macro...** from the Macros menu. Transform will prompt you to name the macro. For our example, click **OK** to accept the default name. The macro will be saved and its name will be listed under the Macros menu.

Now open 'Yvel.hdf' and keep its dataset array window to the front. Under the Macros menu, select the macro you created from the 'Xvel' dataset. You should see a surface plot that looks like the one you created, except that it uses the 'Yvel' data and not the 'Xvel' data.

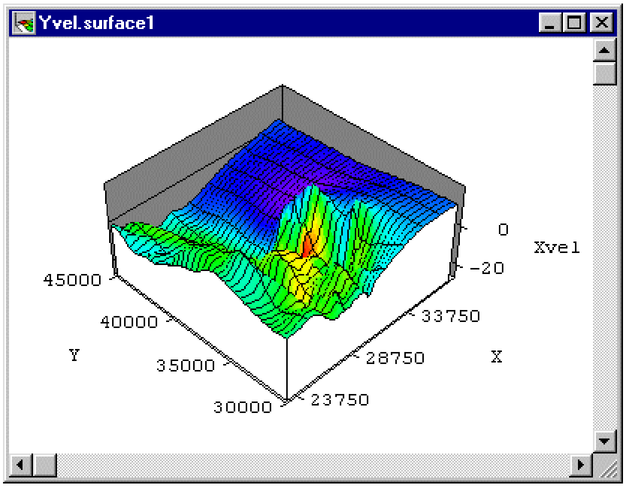


Figure 7-9: New Surface Plot using Yvel

Note

Windows displays the Yvel data scales where Power Macintosh displays the data scales for Xvel.

To edit your macro, select **Edit Macros...** from the Macros menu. The dialog shown below will appear.

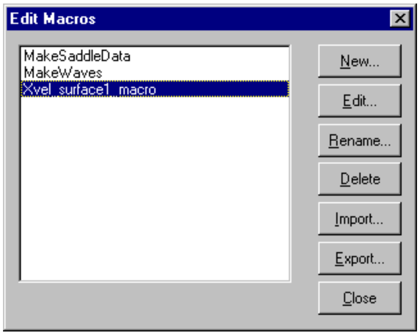


Figure 7-10: Edit Macro...Dialog

Select your macro, then click **Edit...**. You will see a line-by-line listing of your macro, like that in Figure 7-11.

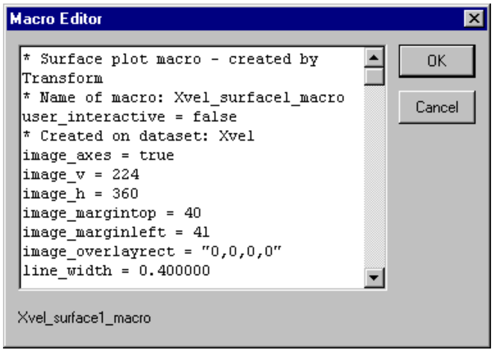


Figure 7-11: Macro Editor Listing

Change Scales

In Windows, select the line **data_autominmax=true** and change it to **data_autominmax=false**.

In Power Macintosh, these settings are correct. No action is required.

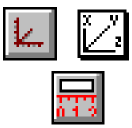
Change Data Axis Title

As a last step, modify both occurrences of the following line in the macro for the data axis and color bar:

axis_title = "Xvel" to be **axis_title = "Velocity"**.

Click **OK**, then **Close/Done** to return to the main window. Make the 'Yvel' dataset window active, and select the macro from the Macros menu again. You will see your changes in the new image that is created.

Change the Plot Title



Click the **Axes tool**. Click the **Plot Title** (Windows) or **Labels...** button, then the **Title...** button (Power Macintosh) and enter the title ‘Velocity Component’ in the dialog. Your axes labels may look different from those in Figure 7-12, since Transform uses the same font, font size, and spacing specifications of the last image that you generated.

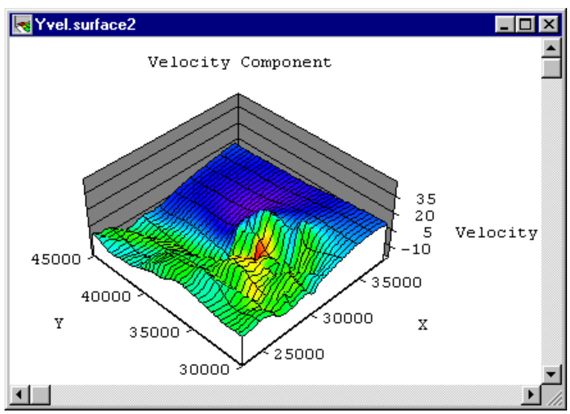


Figure 7-12: New Surface Plot Created from Edited Macro

Chapters 22 and 23 give more information on macros, along with additional examples and a listing of macro language subroutines, functions, and reserved variables.

Chapter 8:

Column Data

Transform only works with 2D matrix data. If your data is in text column format, Transform will import and convert the text column data to matrix data. Transform grids the data into a regularly spaced grid, based on the parameters you set in the import dialogs described in this chapter.

In this chapter, we show you how to work with text column data. You will first import the sample column dataset 'weather.txt'. Then you will manipulate the data, and create an image with a map overlay.

Before you start this part of the tour, close any open windows.

Importing Column Data

To start, select **Open...** from the File menu. Then select and open the file 'weather.txt' in the ‘Samples\Weather’ folder. The Import File Format dialog will appear.

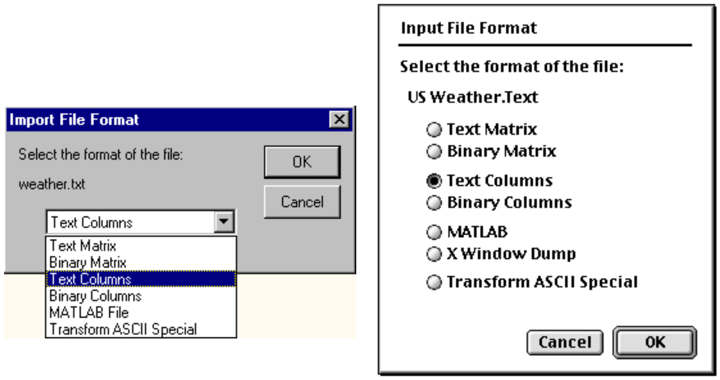


Figure 8-1: Import File Format Dialog (Windows and Macintosh)

In the Import File Format dialog, select **Text Columns**, then click **OK**. The Text Columns dialog will appear. Note that Transform has scanned the text file and entered dimensions for Header Lines, Rows, and Columns in the dialog.

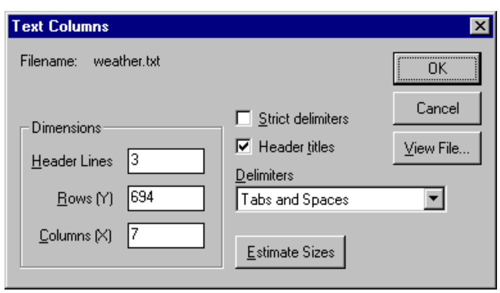


Figure 8-2: Text Columns Dialog

Verify that your numbers are the same as shown in Figure 8-2. To preview the data, click **View File...** shown in Figure 8-3.

Use the scroll bars in the View File dialog to scroll through the data. The first three lines of the data file are comments, called header lines. The last header line has titles for columns.

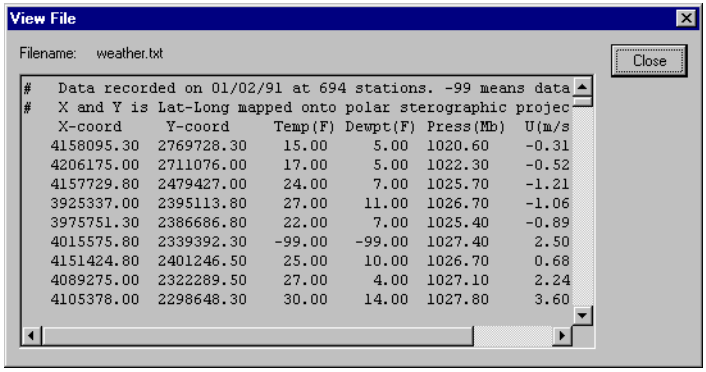


Figure 8-3: View File Dialog

With that noted, click **Close/Done** to return to the Text Columns dialog.

Now, make sure that you select the option of **Header titles (**or **Last header line contains column titles)**. This tells Transform to use names found in the last header line as the dataset names, rather than using column numbers to name datasets. Click **OK** to go to the Select Column(s) dialog.

Specifying the Columns and the Target Matrix

Now that you have indicated how the data is formatted, you must specify two columns as the independent variables and another column as the dependent variable. For our example, make sure the X column is 1, the Y column is 2, and the Data column is 3: Temp(F).

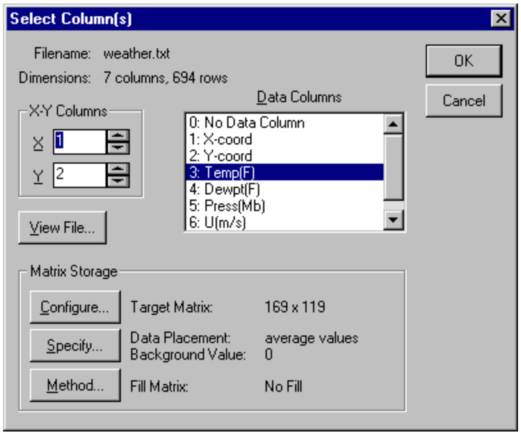


Figure 8-4: Select Column(s) Dialog

Now, make sure the Target Matrix size is 100 by 100. If it is not, click the **Configure...** button next to Target Matrix to bring up the Target Matrix Parameters dialog shown in Figure 8-5. Enter 100 for # of Points in both X and Y, then click **OK** to return to the Select Column(s) dialog.

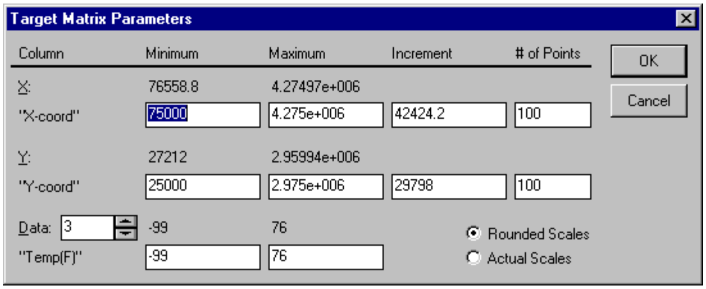


Figure 8-5: Target Matrix Parameters Dialog

Next, click **Specify...** to choose how to place your data values into the matrix. The Data Placement dialog in Figure 8-6 will appear.

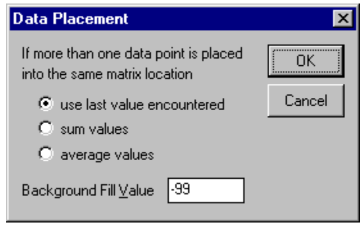


Figure 8-6: Data Placement Dialog

In this dialog, be sure that you select **use last value encountered** (Windows) or **Replace Previous Value** (Power Macintosh), and that you set the Background Fill Value to **-99** (which is the same number used in the dataset itself to specify a missing data value). Click **OK** to return to the Select Columns dialog.

Next, click **Method...** to specify the matrix filling method. The dialog shown in Figure 8-7 will appear.

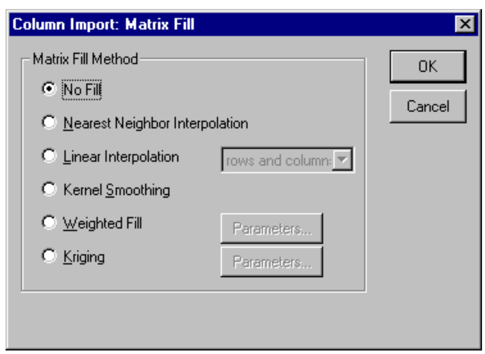


Figure 8-7: Column Import: Matrix Fill Dialog

Click **No Fill** (in Power Macintosh, also set the Missing Data value to **-99**). Click **OK** to return to the Select Columns dialog. The dialog should now look like Figure 8-8.

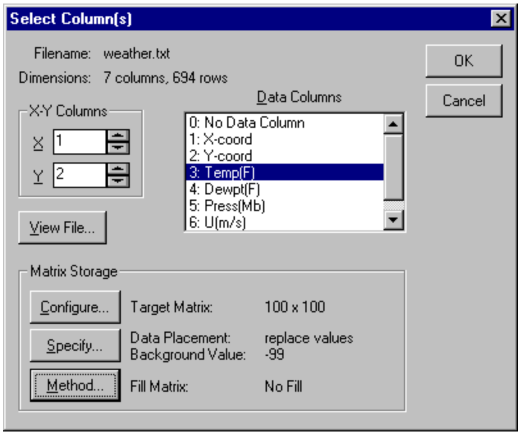
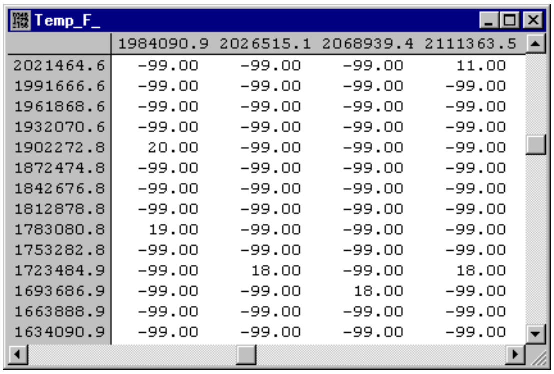


Figure 8-8: Select Column(s) Dialog with all Entries Made

Finally, click **OK** to convert and import the data into Transform.

First, Transform calculates the range of X and Y values in columns 1 and 2. Transform then divides these ranges into 100 intervals in X and 100 intervals in Y. This new 100 x 100 matrix is then filled with '-99', the specified fill value. Finally, all the actual data values from column 3 are placed in the matrix in the closest X-Y location, replacing the -99 in those locations.

Note that the name of the resulting dataset (Figure 8-9) has been changed from 'Temp(F)' to 'Temp_F_'. This is because parentheses are not legal characters for variable or dataset array names.

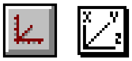

Temp_F_

	1984090.9	2026515.1	2068939.4	2111363.5
2021464.6	-99.00	-99.00	-99.00	11.00
1991666.6	-99.00	-99.00	-99.00	-99.00
1961868.6	-99.00	-99.00	-99.00	-99.00
1932070.6	-99.00	-99.00	-99.00	-99.00
1902272.8	20.00	-99.00	-99.00	-99.00
1872474.8	-99.00	-99.00	-99.00	-99.00
1842676.8	-99.00	-99.00	-99.00	-99.00
1812878.8	-99.00	-99.00	-99.00	-99.00
1783080.8	19.00	-99.00	-99.00	-99.00
1753282.8	-99.00	-99.00	-99.00	-99.00
1723484.9	-99.00	18.00	-99.00	18.00
1693686.9	-99.00	-99.00	18.00	-99.00
1663888.9	-99.00	-99.00	-99.00	-99.00
1634090.9	-99.00	-99.00	-99.00	-99.00

Figure 8-9: New Dataset Window

In Power Macintosh, choose **Attributes...** from the Numbers menu and change the Array Data to **F8.2** and the Row/Col Labels to **F9.1**

With the new dataset window up, select **Generate Image** from the Image menu.



Select the **Axis tool** and click the **Show/Hide Axes** button. Your image should look like a squeezed scatter plot of the United States. The background fill values of -99 are displayed as purple (the lowest color).

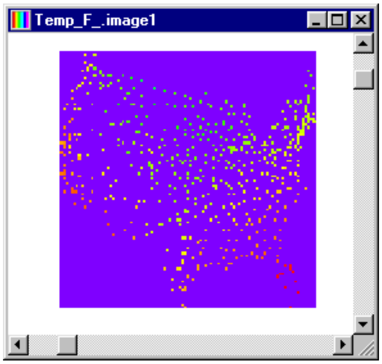
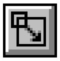
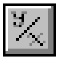


Figure 8-10: Image Corresponding with the New Dataset Window

Modifying the Aspect Ratio

We will now try to make a better image by first fixing the aspect ratio, the squeezing of the image.

Aspect Ratio Button (Windows)

Select the **Resize tool** and press the **Aspect Ratio** button. Your image should now be adjusted to look like that shown in Figure 8-11.

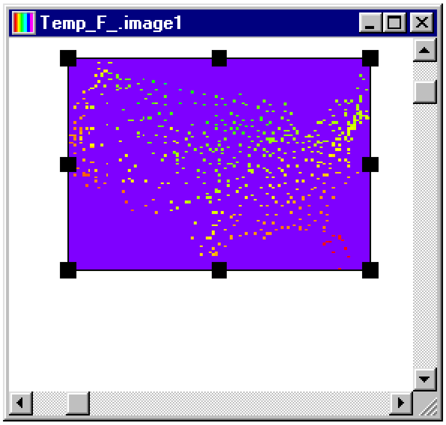


Figure 8-11: Image with Corrected Aspect Ratio

Calculating the Aspect Ratio (Power Macintosh)

We will now try to make a better image by first fixing the aspect ratio, the squeezing of the image. To generate an image with the proper aspect ratio, enter these lines into the notebook window, select all three lines, and then select **Calculate From Notes** from the Numbers menu.

```
image_v = 200
image_h = 200 * abs( colrange(Temp_F)/rowrange(Temp_F) )
call image(currentdataset)
```

The reserved variables **image_v** and **image_h** specify the vertical and horizontal size of images. The functions **colrange()** and **rowrange()** calculate the horizontal and vertical numerical ranges of the specified dataset, so **colrange()/rowrange()** is the image aspect ratio. The subroutine line **call image(currentdataset)** generates a raster image of the current dataset. The resulting image should look like the following.

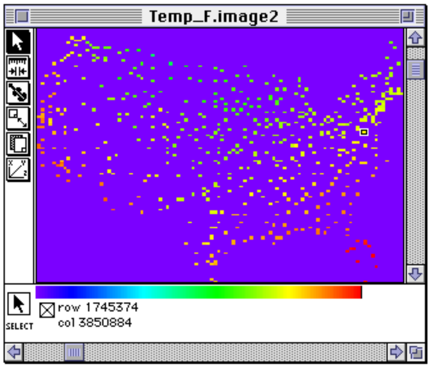


Figure 8-12: New Image

Modifying the Data Minimum and Maximum

The colors are being mapped between a data minimum of -99 to a data maximum of 76, even though the lowest valid data value is actually -15 (recall that -99 is the background fill value).

Calculating the Data Minimum and Maximum (Macintosh)

You can see this by clicking on the **Min/Max tool** on the raster image, and then clicking on the **Color Bar...** button. You should see the dialog shown below.

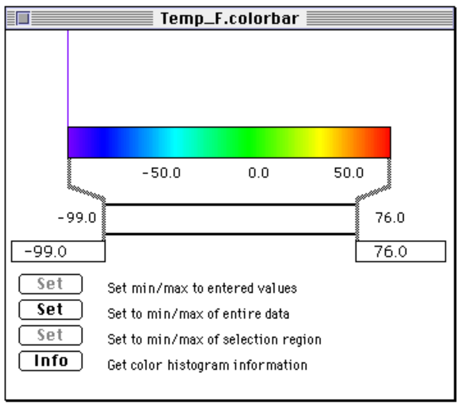


Figure 8-13: Color Bar Window

Enter **-15** for the lowest value, then click the **Set** button next to 'Set min/max to entered values'. Your dialog should look like the figure below.

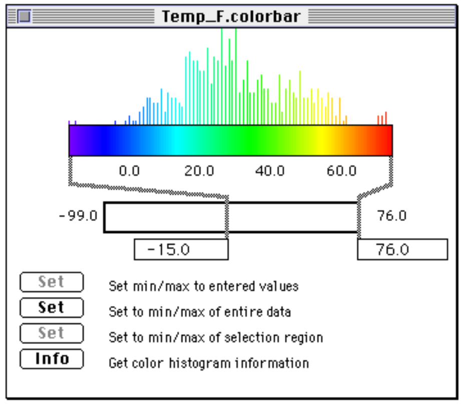


Figure 8-14: Changes to the Color Bar

Close this window, then click **Replot**. Your image will now be more accurate.

Adjusting the Data Minimum and Maximum (Windows)

The colors are being mapped between a data minimum of -99 to a data maximum of 76, even though the lowest valid data value is actually -15 (recall that -99 is the background fill value).

To adjust the range, select the **MinMax tool** and enter **-15** in the Data Min field and press Enter. Notice how the contrast in your image is enhanced.

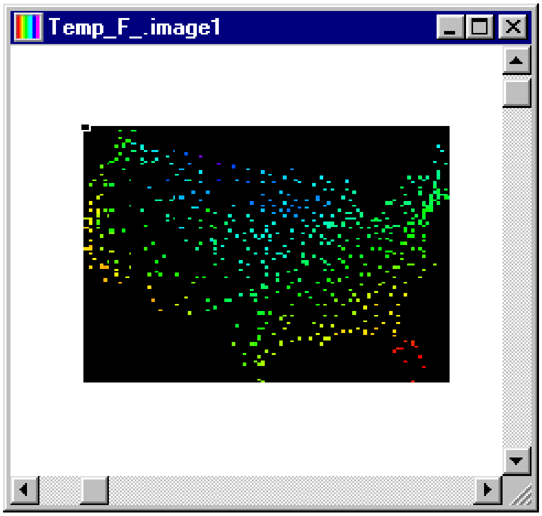


Figure 8-15: Image Contrast Enhanced

Filling Missing Data Values

We will now fill in all of those -99 data points. With the Temp_F dataset the active window, select **Fill Missing Data...** from the Numbers menu. The Fill Missing Data dialog, shown in Figure 8-16, will appear.

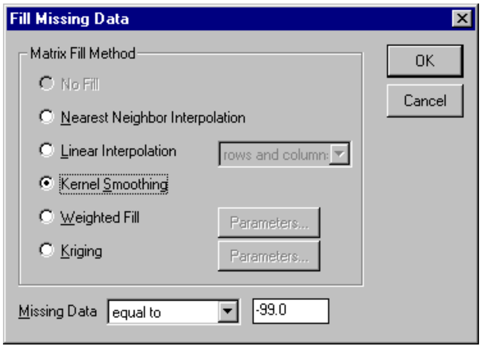


Figure 8-16: Fill Missing Data Dialog

Select **Kernel Smoothing** (the fastest and smoothest method here) and be sure that 'Missing Data' is 'equal to' **-99**. Click **OK** to produce an interpolated version of the dataset, named 'Temp_F__md'.

In the interpolated data set, the true data values have been used to fill in the full matrix of values. The original data elements have been smoothed along with the new data elements, to generate a smooth dataset.

Now select **Interpolated Image** from the Image menu to create an interpolated image as shown in Figure 8-17. We turned off the axes and corrected the aspect ratio, as described earlier.

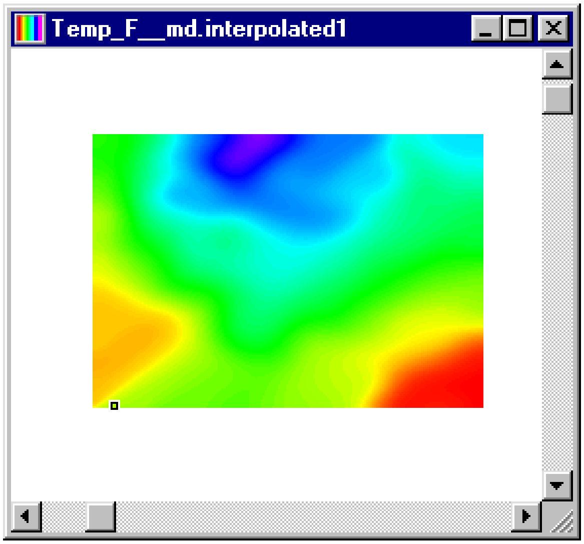


Figure 8-17: Interpolated Image

Add Graphics Overlay

Next, open the file 'us_map.hdf' in the ‘Samples\Weather’ directory. The file consists of a blank dataset with an overlay of a continental U.S. map. Make the ‘US_Map_Overlay.contour1’ (Windows) or ‘Blank_Data.image1 (Power Macintosh) image window active. You are going to copy the graphic of the United States that appears in this window as shown below.

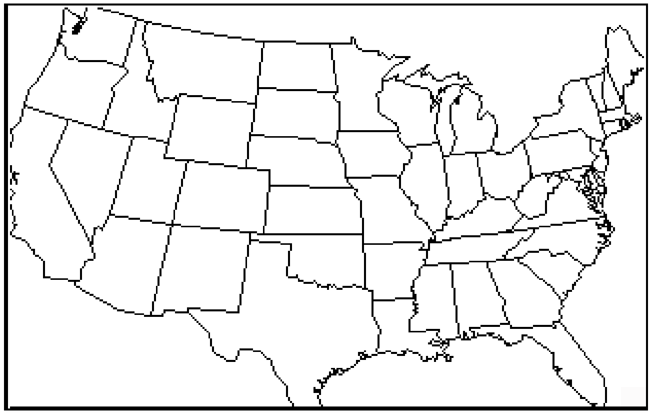
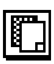


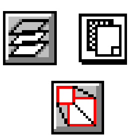
Figure 8-18: Overlay

In Windows, select **Copy As...** from the Edit menu to open the Copy As dialog. In the left-hand list in this dialog, select **Top Overlay**, and click **OK** to copy the overlay (only) to the clipboard.



In Power Macintosh, select the **Overlay tool**, then select the **Copy Overlay** command from the Edit menu.

Now make the interpolated Temperature image window active, and select **Paste** (Windows) or **Paste Overlay** (Power Macintosh) from the Edit menu. A very large U.S. map will be overlaid on the image.



To make the map the proper size, select the **Overlay** tool and click the **Size to Match** button, which appears in the toolbar. The overlay tool lets you independently size and position the graphics overlay. In any case, the resulting smooth image with the overlay should look like the following:

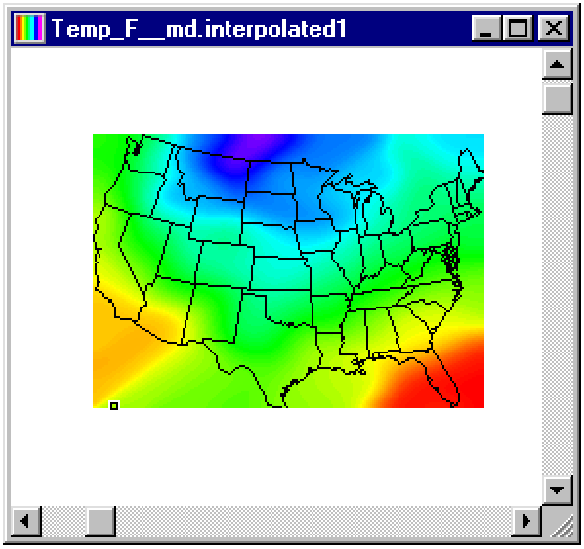


Figure 8-19: Final images

Because this particular overlay has virtually the same boundaries as the data, it matched the image with little adjustment. In general, however, you may have to carefully size and position an overlay to match your other image.

Part III: Reference Manual

Chapter 9:
Importing Files

Transform can import many file types, including HDF, TIFF, generic binary, generic text, GIF (Windows), BMP (Windows), PICT (Power Macintosh), PBM, FITS, XWD and MATLAB 4.0 files. Some file types are imported automatically, others require you to provide information to describe the file. This chapter specifies all of the types and describes the dialogs used for importing data.

All Fortner Software visual data analysis products use the Hierarchical Data Format (HDF) as their primary data storage format. HDF is a scientific data file format developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign. It is a machine-independent binary file format standard for storing matrix, column, and polygonal data from a wide variety of disciplines. HDF files can store floating point data, scaling information, color images, text annotation, and data objects (for more information, see Appendix C).

Opening the Data File

Use the **Open...** command from the File menu to begin the import process. If the file format is automatically recognized, Transform opens the file and reads the dataset into memory. It then creates a dataset window to display the data. If images or notes were saved with the dataset, they are also displayed. If the file type cannot be detected automatically, you are prompted for more information about the file and the way the data is stored (see below).

For files with multiple data records and files with three-dimensional arrays of data, Transform allows you to select which portion of the data to read into memory.

Automatic Import File Types

Transform reads the first part of the data file to try to determine its type. The following file types are automatically recognized.

Transform HDF Files

HDF files created or saved from Transform can contain two dimensional datasets, notebook information and images. When Transform opens a file of this type, the dataset, Notebook and any images that are in the file are opened automatically. If a Transform HDF file has been edited in Noesys, a dialog will appear on import giving you the choice to import the file into a new Transform-specific HDF file or open the file as a Transform file at the risk of losing HDF data objects not supported by Transform.

Transform can read slices from three-dimensional datasets stored in HDF files. Only axis-parallel slices are supported. Information about reading slices from three-dimensional arrays is discussed later in this chapter.

Other HDF Files

When opening HDF files that were not originally created in Transform, the multiple record selector will appear (described later in this section) with a list of all the HDF objects that are in the file that can be read by Transform. Those objects are then read into a new HDF file that can be manipulated and saved.

In Power Macintosh, if you open an unsigned integer*1 dataset, or file, Transform opens the dataset and automatically creates an image from the data. This is because Transform assumes that all unsigned integer*1 data is image data (e.g., TIFF or PICT).

HDF Images

Imported 8-bit images are read as two-dimensional arrays of 8-bit data values and converted to a normal dataset with values between 0-255.

HDF Vset (Power Macintosh only)

HDF Vset datasets may be read as column data. The first Vgroup is read and each Vdata record is used as one column of data. Refer to the Select Columns dialog for gridding and filling options.

ASCII Special

ASCII Special is a Transform-specific format for a text file that represents a two-dimensional array with row and column scales. Usually ASCII Special files are detected automatically, but if your file contains header lines, you may need to read it in as a text file. ASCII Special text files should be in the format shown in Table 9-1.

```
#rows #cols
max_value    min_value
row1  row2   row3   row4 . . . .
col1  col2   col3   col4 . . . .
data1 data2  data3  data4 . . . .
```

Table 9-1: Format for ASCII Special File

The ASCII numbers may be in integer, floating point, or exponential format, separated by spaces, tabs, or end-of-lines. Maximum and minimum values define the range of number values in the region of interest. If set to 0 and 0, Transform will calculate the minimum and maximum for you. Table 9-2 shows a sample ASCII special dataset with 5 rows, 10 columns, and 50 data points.

```
5     10
9.9e0        0.0e0
10.0  20.0   30.0   40.0   50.0
1     2      3      4      5      6      7      8      9      10
0.1   0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9    1.0
1.1   1.2    1.3    1.4    1.5    1.6    1.7    1.8    1.9    1.0
3.1   3.2    3.3    3.4    3.5    3.6    3.7    3.8    3.9    1.0
6.1   6.2    6.3    6.4    6.5    6.6    6.7    6.8    6.9    1.0
9.1   9.2    9.3    9.4    9.5    9.6    9.7    9.8    9.9    1.0
```

Table 9-2: Sample ASCII Special Text File

Note

For ASCII Special text files, the data MUST be in this exact format for it to be read correctly. There are no optional fields. If you do not care about numerical row and column labels, then use the simpler text matrix import instead.

Microsoft Windows BMP (Windows only)

Microsoft Windows Device Independent Bitmap (BMP) files can contain images that are 1-, 4-, 8-, or 24-bits per pixel. The 1-, 4-, and 8-bit images have color maps, while the 24-bit images are direct color. These files store low byte first and have no word alignment. Each file contains a file header, a bitmap header, a color map (unless the image is 24-bit direct color), and the image. Four- and eight-bit images may use an optional RLE compression scheme, otherwise the data is uncompressed. Bits are logically stored a row at a time, and each row is padded to a multiple of four bytes. Currently, Transform supports only 4- and 8-bits-per-pixel, uncompressed BMP files.

FITS Files

Flexible Image Transport System (FITS) is the bitmap file format universally used by astronomers to store and transfer astronomical images. Originally, the FITS format was designed for magnetic tapes, but it is also used on disks and CD-ROMs. The data in a Basic FITS file is an N-dimensional array. Every FITS file must contain SIMPLE, BITPIX, NAXIS, and END cards, in that order, and can contain other records interspersed among them. Transform supports FITS files containing two- or three-dimensional arrays of 8-bit unsigned, and 16-bit and 32-bit signed integer values, as well as 32-bit and 64-bit floating point data. The data is scaled according to the BZERO and BSCALE keywords before import. The ASCII header, which contain information on the data, is placed in the notebook window. For Transform to read a FITS file, SIMPLE must be present, BITPIX is required, and NAXIS must be equal to 2 or 3. BZERO and BSCALE are optional and are used to control the scaling.

GIF Files

Graphics Interchange Format (GIF) files are bitmap (raster) image files commonly used for on-line transmission of graphics data across networks. GIF images are read into Transform as an array of 8-bit data values, which are then used to create a normal dataset. Transform supports 2-, 4-, 16-, 32-, 64-, and 256-color GIF images, in ’87a’ or ’89a’ format.

TIFF Files

Tag Image File Format (TIFF) is a (raster) bitmap image type used for data exchange for desktop publishing and related applications. It is independent of a computer's architecture, operating system, and graphics hardware. TIFF 1, 2, 4 and 8-bit images are read as an array of 8-bit data values, which are then used to create a normal dataset. TIFF 24-bit images are converted to 8-bit before import. Transform does not support multiple-strip TIFF images or compressed TIFF images.

PICT (Power Macintosh only)

PICT images are drawn as an array of 8-bit data values, which are then used to create a normal dataset. An image window is automatically created to display the image. PICT 24-bit images are converted to 8-bit before imports. Note that some 8-bit images use 256 color entries, whereas Transform uses only 254 colors, so in rare cases you might see a minor color shift.

PICS Files (Power Macintosh only)

PICS files are made up of multiple PICT images. Transform lets you select any PICT image or images from the PICS file to import. See the description later in this chapter for details on selecting from multiple-record files.

PBM Files

Portable Bitmap (PBM) files are drawn as an array of 8-bit data values which are then used to create a normal dataset. An image window is automatically created to display the image. 24-bit PBM images are approximated as 8-bit images before being converted to data. PBM, PNM, PGM and PPM file formats are supported.

Import Files with Multiple Records or Objects

Some MatLab, HDF, and all PICS files contain multiple records or objects. For these files, the Multiple Record dialog appears, shown below. For HDF files, you see a list of HDF objects and sizes, by name. For MatLab or PICS files, you see a list of numbered records and their sizes.

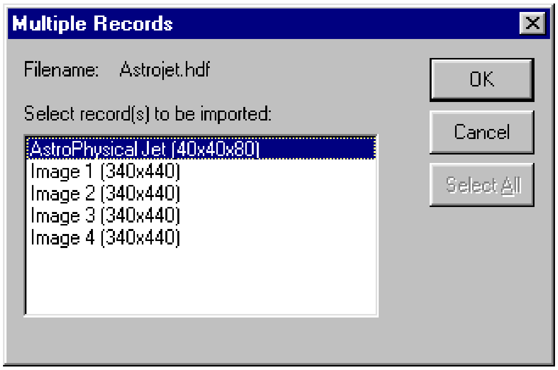


Figure 9-1: Choosing Record(s)

In the scrolling list, select any or all of the records listed. Use the **Select All** button to choose them all. Click **OK** to proceed. Transform imports each of the selected records as a separate dataset.

Note

Transform saves one dataset in each file. Attempts to save a dataset with the same file name as a previous dataset will overwrite the previous file. However, by using the HDF libraries in a program of your own, you may create a file with more than one array of data.

Importing Slices from 3D Arrays

For 3D datasets, you have to choose which 2D slices to import. Each slice is opened as an independent 2D dataset. When you select a 3D array, the Slices/Select Slices dialog appears.

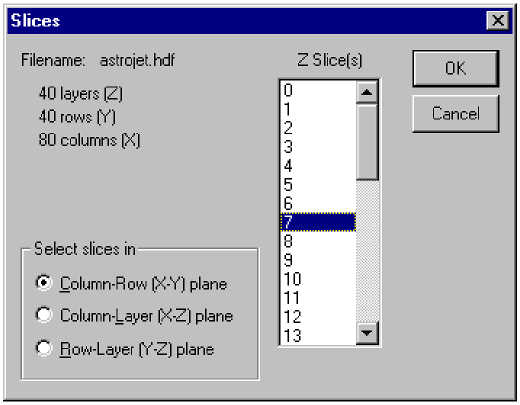


Figure 9-2: Slices Dialog

The radio buttons on the left give you a choice of axis direction for the slices. The scrolling list displays one number for each slice in that axis direction. For one of the X-Y planes, Transform skips to the appropriate location in the file and reads one slice. For Y-Z or X-Z planes, Transform must read the entire file to extract the requested information; this may take some time.

You can select a range of slices by selecting the first slice, and then shift-selecting the last slice. You can select a series of noncontiguous slices by holding down the command key as you select additional slices.

Importing Files that Require Additional Information

Transform requires additional information when importing text matrix, binary matrix, text column, binary column, XWD or MATLAB datasets. If Transform does not automatically recognize the selected file type, the Import File Format dialog (Figure 9-3) appears. The following sections provide a brief explanation for binary vs. ASCII text, column vs. matrix, and each of the acceptable import file types.

ASCII Text vs. Binary

ASCII text files contain human-readable text and numbers. You should be able to open them in a word processor and read the numbers. Binary format files can be read only by machines and fall into one of the following types: byte, short integer, long integer, floating point, or double-precision floating point.

Matrix vs. Column

Transform only works with 2D matrix data (note that Fortner Software's 'Noesys' data management application works with multidimensional data). If your data is in text column format (Table 9-3), Transform will import and convert the text column data to matrix data. Transform grids the data into a regularly spaced grid, based on the parameters you set in the import dialogs.

X	Y	Data
0.3	0.3	0.0350
0.7	1.0	0.0714
0.5	1.5	0.3853
1.1	0.8	0.4911
1.0	1.0	0.2422
1.0	1.2	0.9207
1.5	0.5	0.5744
1.3	1.0	0.3305
1.5	1.5	0.8485

Table 9-3: Text Column Data

In Figure 9-3 you see a scatter plot of the text column data.

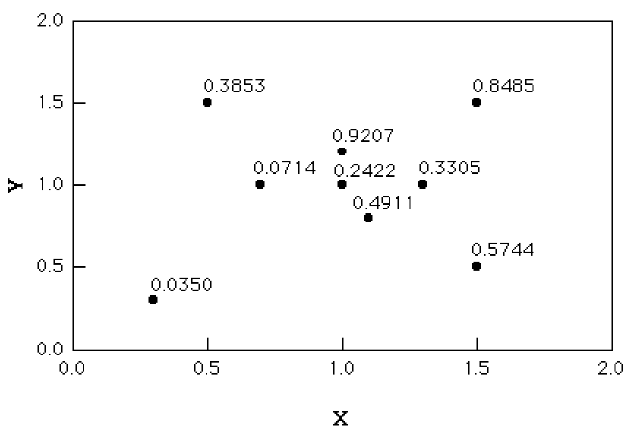
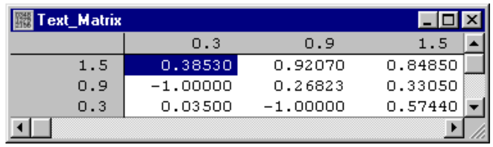


Figure 9-3: Scatter Plot of Text Column Data

Figure 9-4 shows you how Transform places the real data values into the nearest grid position.


Text_Matrix

	0.3	0.9	1.5
1.5	0.38530	0.92070	0.84850
0.9	-1.00000	0.26823	0.33050
0.3	0.03500	-1.00000	0.57440

Figure 9-4: Transform Matrix

Notice that the value at X = 0.9 and Y = 0.9 does not correspond to any data value in Figure 9-5. This is because three of the data values fall into the same grid location. Transform averages those values (0.4911, 0.2422, 0.0714) to produce the value you see (0.2682). You could also set Transform to “use last value encountered” (Windows) or “Replace Previous Values” (Power Macintosh), or to “sum values”.

Note also that at X = 0.3 and Y = 0.9, and X=0.9 and Y = 0.3 there is a data value of -1. This is the fill value for areas where there is no real data. If you look at these areas in Figure 9-5, you will see there is no data there.

Text Matrix Files

Text Matrix files contain data in a two-dimensional matrix. The numbers start with the first row, going from left to right, and continue row by row from top to bottom. Any number of points may be on one line, as long as the data is in order. For more information, see the section on Importing Text Matrix files.

Binary Matrix Files

Binary matrix files contain binary data in two-dimensional matrix format. The numbers start with the first row, going from left to right, and continuing row to row from top to bottom. The data may be byte, short integer, long integer, floating point, or double precision—signed or unsigned. For more information, see the section on Importing Binary Matrix files.

Text Columns Files

Text column files contain text data arranged as columns giving the X-location, Y-location and data values. Each line of the file represents one data point. Multiple data values may be included for each X,Y location. Several gridding and filling options are available. For more information, see the section on Importing Text Column files.

Binary Columns Files

Binary column files contain binary data arranged as columns giving the X-location, Y-location and data values. Each line of the file represents one data point. The data may be byte, short integer, long integer, floating point, or double precision, signed or unsigned. For more information, see the section on Importing Binary Column files.

MATLAB Files

MATLAB files contain data saved in the default format for MATLAB version 3.5 or 4.0. You can choose any dataset from the file to read as an array of double-precision floating point numbers. If you are using MATLAB 5.0, you will need to export data using the `-v4` flag to save into a version 4.0 format. For more information, see the "Frequently Asked Questions" section of Appendix G: Troubleshooting.

XWD

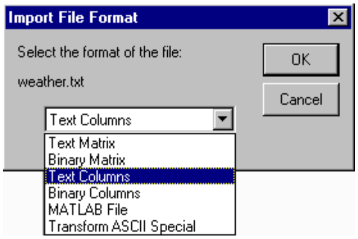
Transform also reads X Window Dump (XWD) files which are commonly used on Unix workstations. The image (any bit depth) is read from the file and converted into a floating point dataset. 24-bit images are converted to 8-bit images before being converted to data. No other intervention is required for X Window Dump files.

Transform ASCII Special Files

Usually ASCII Special files are detected automatically, but if not, you may force Transform to try and read it in this form. See the previous section for more information on this format.

Select the Import File Type

The Import File Format dialog shown below allows you to import the file types described above.



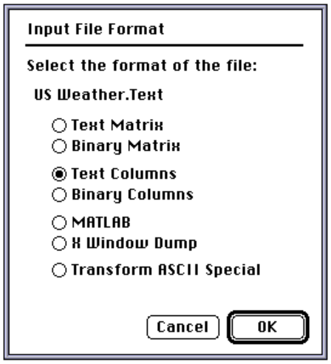


Figure 9-5: Import File Format Dialog (Windows and Power Macintosh)

The dialogs associated with importing text and binary matrix, text and binary columns are provided in the remaining sections of this chapter.

Importing Text Matrix Files

Selecting **Text Matrix** in the Import File Format dialog invokes the Text Matrix dialog shown in Figure 9-6.

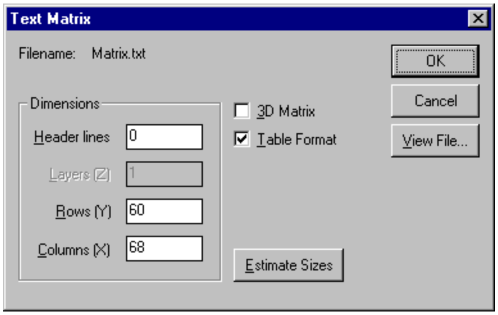


Figure 9-6: Text Matrix Dialog

3D Matrix

Select **3D Matrix** if you have a three-dimensional array of data in the file.

Table Format

Select **Table Format** if your files contain scale values at the top of each column and at the beginning of each row. Table 9-4 shows the same array as Table 9-2 but in table format.

	1	2	3	4	5	6	7	8	9	10
10.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
20.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.0
30.0	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	1.0
40.0	6.1	6.2	6.3	6.4	6.5	6.6	6.7	6.8	6.9	1.0
50.0	9.1	9.2	9.3	9.4	9.5	9.6	9.7	9.8	9.9	1.0

Table 9-4: 2D Text Matrix in Table Format (Scaled Matrix)

Header Lines

Header lines contain comments about your files and cannot be read into Transform. Enter the number of lines of text for Transform to skip. These text lines will automatically appear in the Notebook that is associated with the imported dataset.

Estimate Sizes

Click on this button to have Transform estimate row and column size. Transform assumes that all the numbers for each row are on that same line of the file. If this is not the case, you must override the incorrect estimates.

Rows (Y) and Columns (X)

Here enter the number of rows and columns of data to read from the file. If you selected the 3D Matrix checkbox, you also need to enter the number of layers for the third dimension.

View File

This option lets you view the data in the file before you import. The View File dialog is discussed at the end of this chapter.

Importing Binary Matrix Files

Selecting **Binary Matrix** from the Import File Format dialog invokes the dialog shown in Figure 9-7. The options on this dialog are described below.

For raw binary files, you can specify the rows and column in the file name itself, set off by parentheses. For example, myfile (200 x 300). The Binary Matrix dialog will use these numbers for the row and column sizes. This also works for 3D files when you specify (layers x rows x columns).

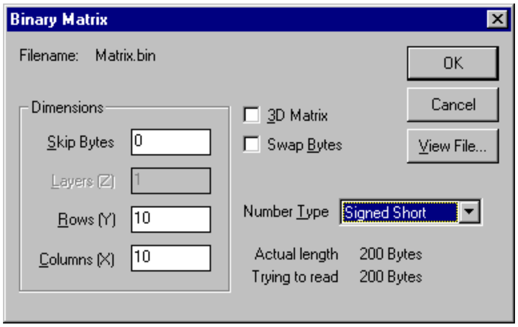


Figure 9-7: Binary Matrix Dialog

Number Type

Select the number type from the menu, shown in Figure 9-9.

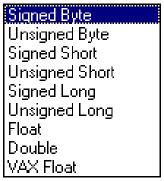


Figure 9-8: Number Type Menu

3D Matrix

Select **3D Matrix** if you have a three-dimensional array of data in the file. When you are finished with this dialog, the Select Slices dialog (described later in this chapter) prompts you to choose which two-dimensional slices to import.

Swap Bytes

Select the option **Swap Bytes** if your data comes from a computer where binary data is stored with the reverse byte order (Intel byte order).

Skip Bytes

Enter the number of bytes for Transform to skip before reading data.

Rows (Y) and Columns (X)

Here enter rows and columns of data to read from the file. If you selected the 3D Matrix checkbox, you also need to enter the number of layers for the third dimension.

View File...

This option lets you view the data in hexadecimal format before you import.

Importing Text Column Files

Importing text column files is a three-step process. First, from within the Import File Format dialog, select **Text Columns** and click **OK**; second, after designating preferences inside the Text Columns dialog, click **OK**; third, choose preferences within the Select Columns dialog. Selecting **Text Columns** invokes the dialog shown in Figure 9-9. The options on this dialog are described below.

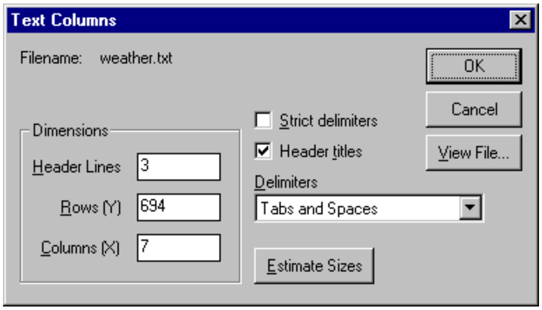


Figure 9-9: Text Columns Dialog

Delimiters

Select the desired text delimiter from the menu, shown in Figure 9-10.

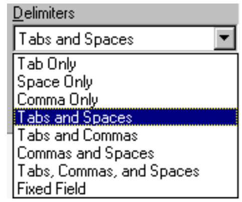


Figure 9-10: Delimiter Types

Importing Fixed Field Column Files

Selecting **Fixed Fields** from the Delimiters pull-down menu changes the **View File...** button to **Set Columns...**. Click this button to specify your fixed field columns in the dialog shown below.

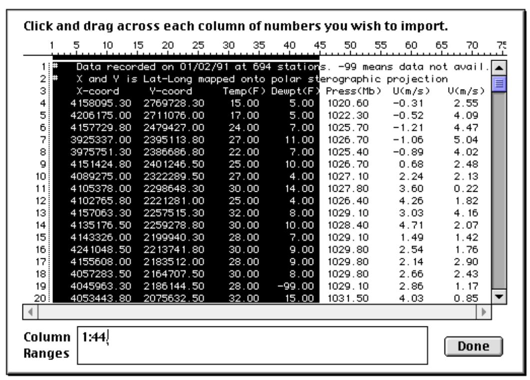


Figure 9-11: Fixed Columns Dialog

Click and drag anywhere in the data portion of the window to select each column of data to import. The listing at the bottom shows you the exact character positions of the columns you select. You may also enter column specifications directly in this listing. Use the form `first:last`, where `first:last` are the first and last character positions for that column. The column specifications are separated by commas. If you overlap a new column onto an existing one, the earlier one is replaced with the new entry.

Strict Delimiter (Windows) or
Separate column for each delimiter (Power Macintosh)

Select this option if you have data columns with missing data, indicated, for example, by two tabs in a row. If you have a data value in each column, leave the box empty.

Header Titles (Windows) or
Last header line contains column titles (Power Macintosh)

Click on this checkbox if your text columns file has a name listed at the top of each column and those names are separated by the same delimiters as the data.

Header Lines

Enter the number of lines of text for Transform to skip. The number includes the header line containing the titles.

Estimate Sizes

Click to have Transform estimate row and columns size, given your choice for delimiters. Transform assumes that all the data values for a row are on the same line. If this is not the case, you must override the incorrect estimates.

Number of Rows and Number of Columns

Here, enter the number of rows and columns of data to read from the file.

View File...

This option lets you view the data in the file before you import. The View File dialog is discussed at the end of this chapter.

OK and Cancel

Click **OK** to go to the Select Columns dialog, described after the “Importing Binary Column Files” section. Click **Cancel** to exit the import dialogs without importing.

Importing Binary Column Files

Importing binary columns files is a three-step process. First, select **Binary Columns** in the Import File Format dialog and click **OK**; second, after designating preferences inside the Binary Columns dialog, click **OK**; third, choose preferences within the Select Columns dialog (described in the next section).

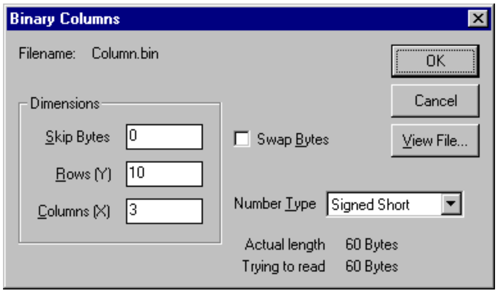


Figure 9-12: Binary Columns Dialog

Number Type

Select the number type from the menu.

Swap Bytes

Select this option if your data comes from a computer where binary data is stored with the reverse byte order.

Skip Bytes

Enter the number of bytes for Transform to skip before reading data.

Number of Rows and Number of Columns

Here enter the number of rows and columns of data to read from the file.

View File...

Lets you view the data in hexadecimal before you import. The View File dialog is discussed at the end of this chapter.

OK and Cancel

Click **OK** to go to the Select Columns dialog, described in the section below. Click **Cancel** to exit the import dialogs without importing the file.

Selecting Columns

If you are importing either text or binary column data, the next step is the Select Column(s) dialog, shown in Figure 9-13. The options on this dialog are described below.

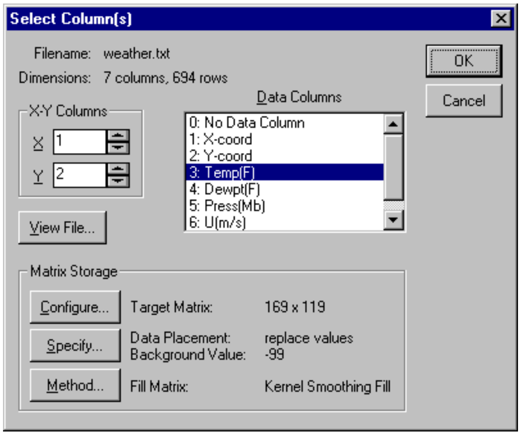


Figure 9-13: Select Column(s) dialog

X-Y Columns (Windows) or Coordinates Columns (Power Macintosh)

Select which column contains the data for Dimension 1 (X) and Dimension 2 (Y). These are the two independent variables in the matrix.

Data Columns

All of the columns are listed, along with the column names if available. Choose the data column or columns from the list. Use the Shift key for continuous selection or the Ctrl (or Command) key for discontinuous selection. For every data column that you select, Transform will generate a separate dataset. If you select column 0, ‘No Data Column’, as your data column, Transform assumes that every X, Y pair has a value between 0 and 5.

View File...

Lets you view the data before you import. The View File dialog is discussed in the next section.

Configure...Target Matrix

Click the **Configure...** button to define the grid that Transform will use in converting your column data into matrix format.

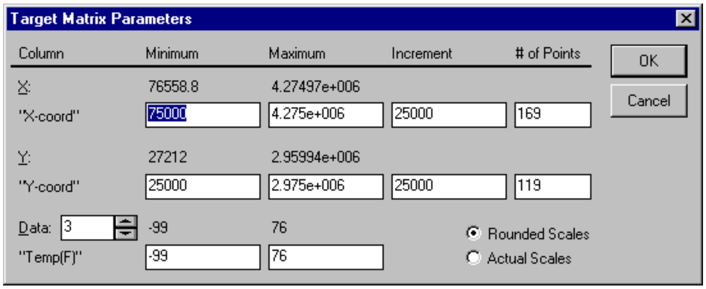


Figure 9-14: Target Matrix Dialog

Minimum and Maximum—To specify the minimum and maximum values for the X and Y numerical scales of the target matrix, enter values inside the boxes to the right of the designated columns. The minimum and maximum values that Transform found in the selected X and Y columns are posted above each box in which you enter values. You may want to reduce the X and/or Y range to omit data values that lie outside your region of interest.

Increment and # Points—Specify the X and Y numerical scales spacing of the target matrix. If you change Increment, then # Points changes, and vice-versa.

Data—For the data columns, you can set minimum and maximum data values of interest. Any values outside the range specified for that column are omitted. The actual data column minimum and maximum appear directly above these fields. Change the number in the Data box or use the arrows to cycle through the eligible data columns.

Rounded Scales and Actual Scales—Enters the exact minimum and maximum X and Y values found in the column data. Click on Round Scales to have Transform suggest X and Y scale values with a rounded increment.

Specify... Data Placement

Click the **Specify...** button in the Select Columns dialog to open the Data Placement dialog. This dialog lets you establish a method for handling multiple values per cell and the value for cells that receive no data values.

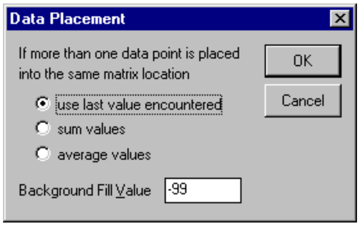


Figure 9-15: Data Placement Dialog

Use Last Value Encountered (Windows) or
Replace Previous Value (Power Macintosh)—Second or succeeding data values simply replace earlier entries.

Sum Values—Later values are summed so each cell contains the total of all data points that mapped to that cell.

Average Values—Each cell gets the average of all data points that are placed in that cell.

Background Fill Value—Specifies the data value to use for cells that get no data values. Transform attempts to provide a suitable fill value by default.

Method... Fill Matrix

Click the **Method...** button in the Select Columns dialog to open a dialog that lets you specify what interpolation method you want Transform to use to fill out the matrix from your scatter data. Click **No Fill** to just place your scatter data and leave the rest of the Target Matrix set to the Background Fill Value. Fill methods are discussed in detail in Chapter 11: Data Preparation.

Note

For large datasets, we recommend that you import using the **No Fill** option, save your dataset as HDF and then fill the data using the algorithms available from the Numbers menu as described in Chapter 11.

View File

In most of the import dialogs, you have the opportunity to view the contents of your data file with the View File dialog. It lists all of the data in your file with row and column numbers to help you fill in the required fields for importing. If this dialog displays data that you cannot read, you may be trying to import binary data as text data.

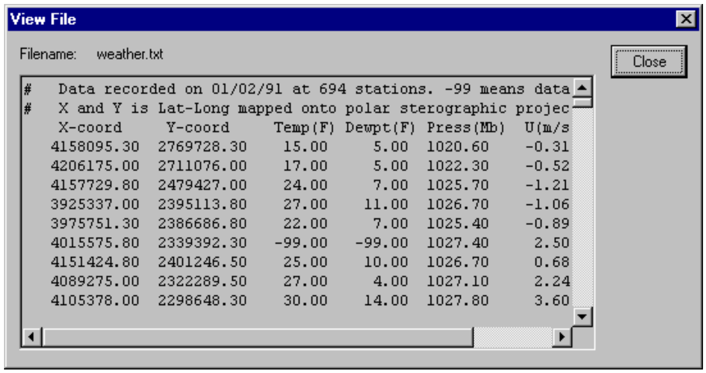


Figure 9-16: View File Dialog - ASCII

The binary form of the View File dialog is shown below. It lists each data byte in hexadecimal, 16 values per line. On the right side, it lists the ASCII equivalent of each byte, if any. If you can read all the text on the right side, you are probably trying to import text data as binary data.

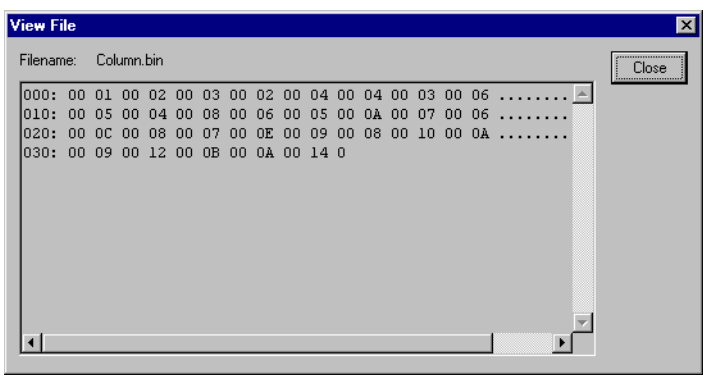


Figure 9-17: View File Dialog - Binary Data

Chapter 10:

Data Windows

When you load or create a dataset, Transform creates a dataset window that shows the values formatted as a spreadsheet. If you have notebook information stored with the dataset, Transform also opens a Notebook window.

Data in Transform

Transform works with two-dimensional arrays of IEEE floating point numbers. Transform's working representation of a dataset has the following properties:

- Single-precision, 32-bit floating point numbers are used for all values.
- Each array can be defined in terms of three variables: two independent variables that correspond to the row and column labels, and a dependent variable that has a value at every point in the array.
- The names of the variables are stored and used when appropriate. The dependent variable name also functions as the name of the dataset window and is used in notebook calculations.
- A list of numbers exist for each of the independent variables. One list has an entry for each column and defines values along the width of the grid. The other list has an entry for each row, and defines the values along the height of the grid.
- Transform stores the grid size as number of rows (rows) and columns (cols).
- The values of the dependent variable are located inside the grid, one at each grid point. The total number of values is rows*cols.
- Fortran format fields are maintained for the variables. When the floating-point numbers appear in the dataset window, they are printed in the specified formats.

The floating point values stored by Transform are 4-byte, 32-bit IEEE standard floating point numbers. 24 bits are used for the mantissa.

Significant precision:	7 to 8 digits
Maximum positive number:	3.4E+38
Minimum positive number:	1.2E-38
Maximum negative number:	-1.2E-38
Minimum negative number:	-3.4E+38
Infinity:	INF
Not a Number:	NaN (divide by 0, log of zero)

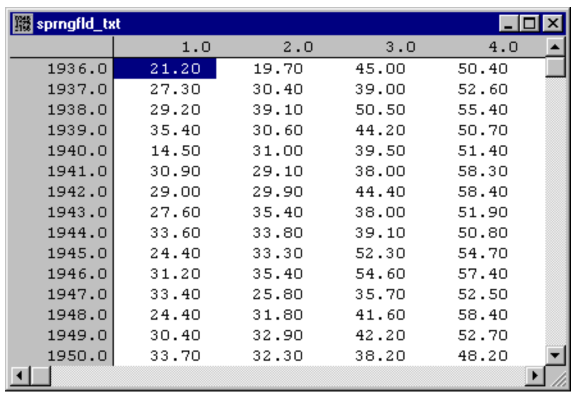
Table 10-1: IEEE Floating-point Values

Open/Create New Datasets and Attributes

This section describes the commands used to open a file into a dataset window, create a new array and modify the dataset display.

Open... Command

Select **Open...** from the File menu to choose a file to open in Transform. If the file type is automatically recognized by Transform, the dataset window appears, as shown in Figure 10-1. If the file type is not automatically recognized, refer to Chapter 9 for information on file types, data types, and how to ensure that your data can be read into Transform.



sprngfld_txt

	1.0	2.0	3.0	4.0
1936.0	21.20	19.70	45.00	50.40
1937.0	27.30	30.40	39.00	52.60
1938.0	29.20	39.10	50.50	55.40
1939.0	35.40	30.60	44.20	50.70
1940.0	14.50	31.00	39.50	51.40
1941.0	30.90	29.10	38.00	58.30
1942.0	29.00	29.90	44.40	58.40
1943.0	27.60	35.40	38.00	51.90
1944.0	33.60	33.80	39.10	50.80
1945.0	24.40	33.30	52.30	54.70
1946.0	31.20	35.40	54.60	57.40
1947.0	33.40	25.80	35.70	52.50
1948.0	24.40	31.80	41.60	58.40
1949.0	30.40	32.90	42.20	52.70
1950.0	33.70	32.30	38.20	48.20

Figure 10-1: Dataset Window

New... Command

Use the **New**... command from the File menu to create a new array of all zeros or ones without opening a file. The New Dataset dialog shown in Figure 10-2 appears. Enter a new name or leave the default name of “Untitled” in the first edit box. In the rows and columns boxes, enter the desired size for your new array or accept the default of 20 rows and 20 columns. The smallest size is 1 by 1 and the largest size is limited only by available memory. Two radio buttons allow for the choice of all 0’s or all 1’s for the data values in the array.

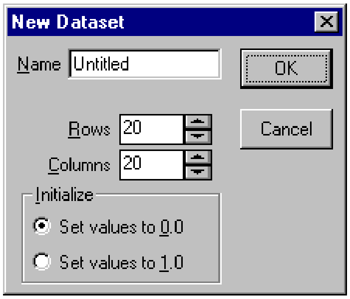


Figure 10-2: New Dataset Dialog

Click **OK** to create and display the new dataset in its own window.

Attributes... Command

The name of the dataset, the printing formats used in the dataset window and other settings can be changed in the Attributes dialog. To access the Attributes dialog for the current dataset, choose **Attributes...** from the Numbers menu.

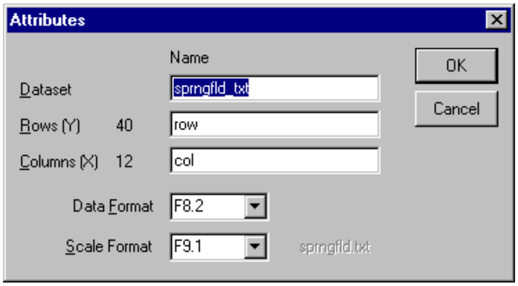


Figure 10-3: Attributes Dialog

The name displayed in the lower right of the dialog is the filename associated with the current dataset. You can only change the dataset name in the Attributes dialog, not the name of the file. To save the dataset with another filename, use the **Save As...** command.

Names

Transform uses the dataset name as the title for the dataset window and the prefix name for all windows associated with this dataset. You will also use this name in macros to refer to this dataset. If you change the name using this dialog, the change will be reflected immediately in the name at the top of the data window and all associated windows.

The Row and Column names are labels for the row (Y) and the column (X) dimensions of the dataset. These entries appear as labels for the selection region printed in image windows, as the axes text labels in plots, and in printouts.

Display Formats

Data Format specifies the printing format for all array data in the dataset window. Scale Format specifies the printing format for the row and column scale labels in the dataset window. All formats are specified in Fortran-77 format, which are described in Table 10-2. When you click **OK**, the dataset window redraws with the formats you specify. This only changes the printing format of the numbers, not their accuracy for image generation or calculations.

Format Specifications

Display format specifications are used for array data, for row/column labels, for numerical scales on the plots, and for labels on contour plots. The options are:

`In` Integer format where ‘n’ is the number of digits allowed in the number. There are no decimal points or exponents printed for integers.

`Fm.n` Floating-point format where ‘m’ is the total number of characters allowed, including the decimal point and minus sign (if negative). ‘n’ is the number of digits to follow the decimal point. Exponents are not printed.

`Em.n` Exponential format where ‘m’ is the total number of characters allowed, including the decimal point, sign, and exponent. ‘n’ is the number of digits which are to follow the decimal point.

Table 10-2 shows the value of pi printed in several example formats. The ¤ character shows where spaces occur.

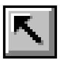
I5	¤¤¤¤3
F5.0	¤¤¤¤3
F5.2	¤3.14
F10.5	¤¤¤3.14159
E9.2	3.14e+000
E20.6	¤¤¤¤¤¤¤¤3.1415593e+00

Table 10-2: Value of pi Printed in Different Fortran-77 Formats

Tools in the Dataset Window (Windows only)

When a dataset window is active in Transform for Windows, the tool palette shows two tool icons, Select and MinMax. Each of these tools gives you data manipulation options for the current dataset.

Select Tool



The Select tool for dataset windows brings up a toolbar which shows you the location of the current selection region and allows you to change individual data values. An example of the toolbar is shown in Figure 10-4. If multiple rows or columns are selected, the boxes display the entire range of selected data.

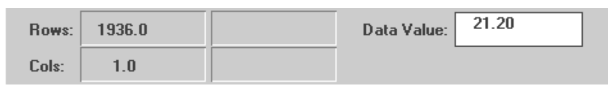


Figure 10-4: Select Toolbar

To change a single data entry, select the value you wish to change, click in the Data Value text field, then type a new value. (Data selection is discussed more fully in Chapter 20.) Press Enter to make the new value take effect.

MinMax Tool

The MinMax tool for dataset windows brings up a toolbar which displays the minimum and maximum values in the current selection region, the current minimum and maximum settings for the dataset, and three button options. An example is shown in Figure 10-5.

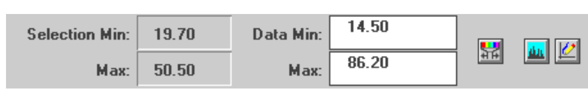


Figure 10-5: MinMax Toolbar

The first button invokes the Dataset Min/Max dialog; this dialog allows you to change the current dataset min/max settings. The second button resets the current data range to the actual minimum and maximum data values. The third button updates all image windows that are dependent on this dataset with the new minimum and maximum settings.

Dataset Min/Max Dialog

Click the Dataset Min/Max button in the toolbar to bring up the dialog shown in Figure 10-6.

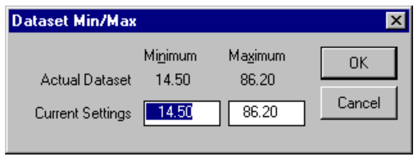
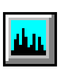


Figure 10-6: Min/Max Dialog

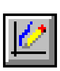
The Dataset Min/Max dialog displays both the actual and the current data range. If you know that certain data values are invalid or are outside your range of interest, change the current settings for the minimum and maximum to include only your desired data values. These values are used whenever images or plots are generated from the Image menu. You can also use the MinMax tool for each image individually to set the data range. See Chapter 13 for more information on how the data range is used and controlled.

Reset



Click this button to reset the current data range to the actual smallest and largest numbers in the dataset.

Update All WIndows



After you have set your current data range, you can update all associated image windows at once. Click this button to redraw all images and plots that depend on this dataset. The effect is the same as entering the identical data range into the MinMax tool of each individual window.

The Notebook Window

The Notebook window in Transform is a place to add comments about your data and to enter and execute macro commands. An example of a Notebook window with text is shown in Figure 10-7. For a description of using the notebook for calculations and examples, see Chapter 22.

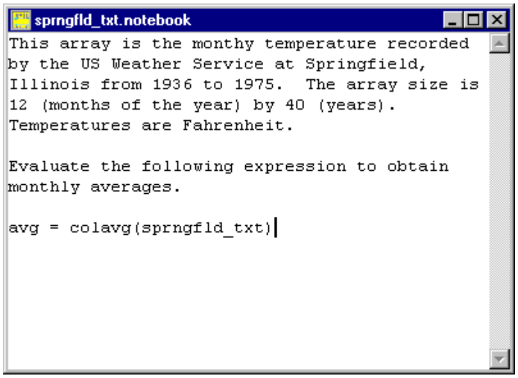


Figure 10-7: Notebook Window

If a Notebook window does not exist for your current dataset, you may create one by selecting **See Notebook** from the Numbers menu. A blank window will appear, where you can type and edit information about the dataset. In Windows, text font and size for the window are set in the Preferences dialog. In Power Macintosh, the text font and size for the window are set with the **Font**... command from the Edit menu.

If you save a dataset for which you had opened a notebook window, that notebook window is automatically saved when you save the file. When you subsequently reload the dataset, the notebook window is loaded and opened automatically.

Note

There is a size limit of 32K to the contents of a Notebook window.

Chapter 11:

Data Manipulation

Several of Transform’s commands manipulate data, creating new datasets based on old ones. This chapter describes the commands and methods used to manipulate and fill data.

Data Manipulation Methods

This section describes the data manipulation commands available from the Numbers menu.

Extract Selection

You can create a new dataset by extracting a region from an existing dataset. To do this, select the desired region of values in a data or image window. Then choose **Extract Selection** from the Numbers menu. A new data window will appear with just the selected data values. The new dataset has the name of the original dataset with an '_x' extension added. The corresponding row and column scales are automatically extracted.

Change Data Entry

To change a single data entry or the format in which data points or scale labels appear, on Windows use the text fields provided in the toolbar when the Select tool is active.

In Power Macintosh, select a data point, choose the **Change Data Entry...** command from the Numbers menu or double-click on any data value. The Edit Data dialog will appear, as shown below.

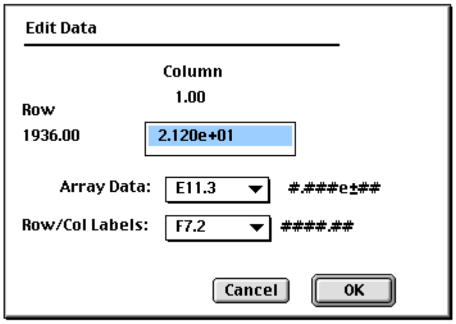


Figure 11-1: Edit Data Dialog (Power Macintosh)

The row and column number of the selected data point are shown. Enter a new value for the data element in the box.

Array Data:

Here you can specify the format for all data elements in Fortran-77 floating point and exponential formats. Select the desired format from the pop-up menu. If none of these prepared formats suits your purposes, you may enter a custom format using the Attributes dialog discussed in Chapter 10.

Row/Col Labels:

Same as above, but for the row and column labels.

Generate Scales

To change the scales of an existing dataset, use the **Generate Scales**... command. from the Numbers menu to invoke the dialog shown in Figure 11-2.

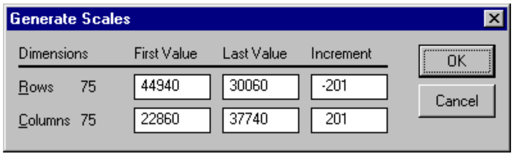


Figure 11-2: Generate Scales Dialog

In this dialog, you may choose new starting and ending values for the row or column scales or both. If you enter a new start or end value, the interval field automatically updates to show you the new interval. Scales may either increase in value with a positive interval or decrease in value with a negative interval. Swapping the axis direction can be done by simply switching the start and end values for the scales. **Generate Scales...** can create only evenly spaced scales.

When you click **OK**, a new, copied dataset is created with the new scale values. The new dataset has the name of the original dataset with a '_s' extension added.

Smooth Data

To smooth your data, select **Smooth Data...** from the Numbers menu to bring up the dialog in Figure 11-3.

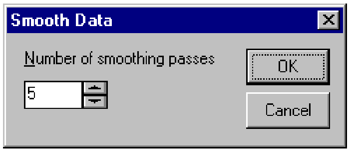


Figure 11-3: Smooth Data Dialog

Specify the number of smoothing passes to be performed, then click **OK**. The smoothing is done by averaging every point with its eight neighbors. This process is repeated for the specified number of passes. Transform creates a new dataset for the smoothed data. The new dataset has the name of the original dataset with a '_sm' extension added.

Resample Data

To create a new dataset of a different size from an existing one, choose **Resample Data...** from the Numbers menu to display the dialog shown in Figure 11-4.

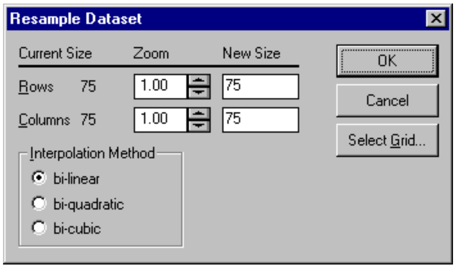


Figure 11-4: Resample Data dialog

The options in this dialog are described below. When done, click **OK** to create the new dataset. The new dataset has the name of the original dataset with a '_rs' extension added.

Zoom, New Size

Specify the size of the new dataset by clicking on the up or down arrows to change the Zoom or by just entering the new number of rows and columns.

Because resampling is done on inside values only, doubling the size of the dataset reduces the interval by a little more than one half. The new array is one number too big for an exact match. Subtract one from the new size if you want the new scale interval to be exactly one half the old scale interval.

Interpolation Method

Choose whether the interpolation should use a bilinear, bi-quadratic, or a bicubic formulation. Data values are interpolated first along rows and then along columns. The interpolation method is used only when resampling to a larger dataset. For resampling to a smaller dataset, an averaging method is used. Every data point in the original dataset that maps to a single point in the result is averaged to calculate the new value.

Select Grid...

If you are resampling to create a different data distribution, click **Select Grid...**. The dialog shown in Figure 11-5 appears.

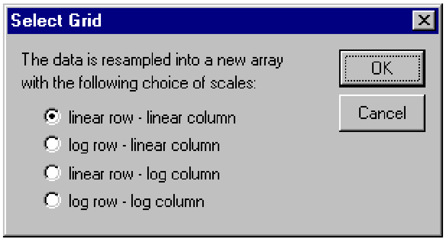


Figure 11-5: Select Grid dialog

Choose whether to use a linear scale or produce one or both scales with axis values distributed along a logarithmic scale. In this case, the dependent values are redistributed according to a logarithmic scale within the matrix.

Note

Even if one or both scale axes are distributed logarithmically, **Generate Image** will not create an image with a logarithmic effect because Transform takes the new logarithmic scale into account when the image is generated. To see a logarithmic image, use the **Generate Scales...** command to force the scales to be uniformly spaced (linear) before generating an image.

Because resampling is done on inside values only, doubling the size of the dataset makes the array two numbers too big for an exact match. subtract two from the new size if you want the new scale interval to be exactly one half the old scale interval.

Fill Missing Data

Transform's **Fill Missing Data...** command is used to replace unwanted data values with estimates of what those data values should be based on neighboring data. Unwanted values usually represent background values or omissions in the data which are set to some arbitrary number, like -999. In Transform, the methods for replacing missing data values can be accessed either from the **Fill Missing Data...** command on the Numbers menu or when importing a data file. The Fill Missing Data dialog, when accessed as a menu command, appears as shown in Figure 11-6 below.

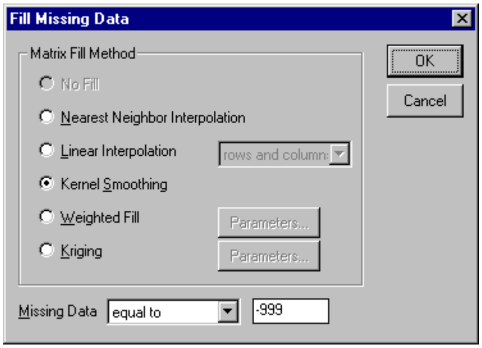


Figure 11-6: Fill Missing Data Dialog

When you click **OK**, the fill process produces a new data window with the missing data values removed. The new dataset has the name of the original dataset with an '_md' extension added. The options in the dialog are discussed below.

Matrix Fill Methods

Transform offers six methods for replacing missing data values with values interpolated from true data values: No Fill, Nearest Neighbor Interpolation, Kernel Smoothing, Linear Interpolation, Weighted Fill, and Kriging. Some methods are faster, easier, and less accurate, like kernel smoothing; others take longer but are more accurate, like kriging. All of the methods become less accurate as the number of missing values increases.

No Fill

This option is only available when the **Methods...** button in the Select Columns dialog is pressed during import. Select this option to bypass Fill Missing Data during import. The option is not available from the Numbers menu.

Nearest Neighbor

The easiest method of filling missing data values is the nearest neighbor fill method, which replaces each missing value with exact copies of the value of its nearest non-missing neighbor. The nearest neighbor method always preserves the true data values and does not average the values.

Linear Interpolation

The linear interpolation method replaces each missing value with a linear interpolation of its nearest neighbors (i.e., data values in the same row, the same column, or both). Select Rows to interpolate from left and right neighbors, Columns to interpolate from above and below neighbors, and Rows and Columns to do both interpolations and then take the average. Linear interpolation tends to show horizontal and vertical streaks but always preserves the true data points. The linear interpolation method is ideal when there are only a few missing data values.

Kernel Smoothing

To calculate the value of a missing data point, kernel smoothing uses the average of the missing value's eight neighbors. Multiple passes are used until no more missing data points remain. This method has a heavy smoothing effect on not only the missing data values but also on known values. Hence, your original data values are not preserved.

Weighted Fill

In the weighted fill method, each missing data element is set to the weighted average of a region surrounding the missing data element. Known data values that are close to the missing data element are weighed heavily, while those far away are weighed lighter. See the section below on Weighted Fill Parameters for more details.

Kriging

Select this option to used Kriging to fill each missing data value with a statistically weighted average of neighboring points. See the Kriging and Kriging Parameters sections below for more explanation and available parameters.

Preserve defined data point values (Power Macintosh)

Select this checkbox to guarantee that all true data values are preserved. If this option is not selected, the original data values may be modified by the filling process. Often the resulting dataset is smoother with this option deselected (for use with Weighted Fill only). Nearest Neighbor, Linear Interpolation and Kriging always preserve the original values. Kernel smoothing never preserves the original values.

Note

On Windows, this option is available in the Weighted Fill Parameters dialog described in the next section.

Missing Data

The combo box at the bottom of the dialog and the edit field next to it specify which values are to be treated as missing data. These numbers are background, or undefined, and need to be eliminated from the dataset. There are three menu choices: equal to integer, less than, and greater than. You may choose an exact value if all of your missing data values are set to the same number (-999 for example). Otherwise, missing data can be defined as all numbers less than the indicated value or all numbers greater than the indicated value.

If you are importing a column dataset, this choice is not available. During import, the background value is already specified in the Data Placement dialog.

Weighted Fill Parameters

In the Fill Missing Data dialog, click the **Parameters...** button next to the Weighted Fill option to display the dialog shown in Figure 11-7. In weighted fill, each missing data value is set to the weighted average of its nearest true data values. The weights depend on the distance of the true values and on a distribution function that you specify. The selections for distribution function and range of the spatial distribution function are discussed below.

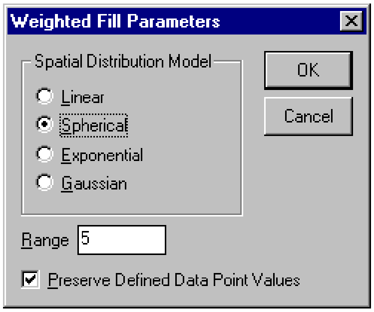


Figure 11-7: Weighted Fill Parameters Dialog

Linear

When you use the Linear Distribution Function method, weights drop off linearly (1/d) with distance from missing data value.

Spherical

The weights drop off as the reciprocal of the distance squared (1/d^2) with distance from the missing data value.

Exponential

The weights drop off as an exponential (e^-d) with distance from the missing data value.

Gaussian

The weights drop off as a normalized Gaussian distribution with distance from the missing data value.

Range

Range specifies the distance from the missing data value to look for true data values, in column/row index number units. The default distance is based on the percentage of missing data values in the dataset, given by the following formula:

```
dist = 2 * sqrt (#_of_pts/(#_of_pts - #_of_missing_pts))
```

If you want a smoother fill, increase the fill distance. For more accuracy, reduce the distance. A distance of ‘1’ means no averaging takes place.

Preserve Defined Data Point Values (Windows)

Select the **Preserve Defined Data Point Values** checkbox to guarantee that all true data values are preserved. If this option is not selected, the original data values may be modified by the filling process. Often the resulting dataset is smoother with this option turned off.

Note

On Power Macintosh, this option is available in the Fill Missing Data dialog.

Kriging

Kriging is the most accurate fill method, especially if the spatial continuity of the attribute (sample values) is more continuous in some directions than in others. For example, if the spatial continuity is more continuous in horizontal than vertical directions. In order to “krig” unsampled locations, the kriging algorithm must know characteristics of the spatial continuity associated with the sample data. This information is supplied through a variogram model or Sill, Nugget, Range, and model type. Together, these parameters describe the spatial continuity.

Kriging estimates the value of an attribute at unsampled locations by computing a weighted linear combination of the neighboring sample values. Kriging uses spatial continuity information to compute “optimal” weights. What are optimal weights? Consider a number of estimates that were calculated using an identical data configuration. There will be an estimation error associated with each of these estimates. Although we do not know these errors, they exist and cannot be eliminated. The best solution is to “minimize the variance” of these errors which is precisely what kriging does. This is why kriging is said to be the most accurate linear estimator.

For additional kriging information, see the following publication:

Isaaks, E.H., & Srivastava, R.M. (1989). *An Introduction to Applied Geostatistics*. Oxford University Press.

Kriging Parameters

In the Fill Missing Data Dialog, click the **Parameters...** button next to **Kriging** to invoke the Kriging Parameters dialog shown in Figure 11-8.

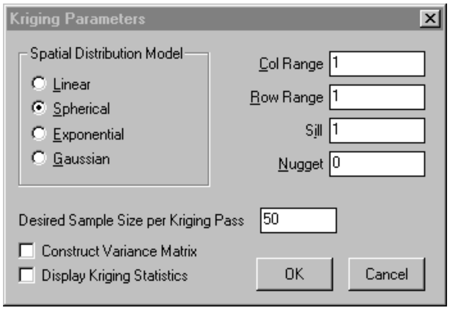


Figure 11-8: Kriging Parameters Dialog

Spatial Distribution Model

Each of these models describes how the spatial continuity changes with separation distance. Usually, the model is chosen that best fits the sample variogram calculated from sample data. Figure 11-9 shows examples of the spherical, exponential and Gaussian models with the same range and sill.

- Linear — The equation for the linear model is given by:

  $\gamma(h) = a * h$

  where h is a distance vector, and a is the slope. Note that this model has no sill.

- Spherical — This is perhaps the most widely used model of spatial continuity. Its equation is given by:

  $\gamma(h) = 1.5 * h/R - 0.5 * (h/R)^3$ for $h > 0$ and $h < R$

  = 1.0 for $h \quad R$

  = 0 for $h = 0$

- Exponential — The equation for the exponential model is given by:

  $\gamma(h) = 1 - e^{-3\,h/R}$

- Gaussian — The Gaussian model is characteristic of extreme spatial continuity. Since earth science data is usually not noted for its extreme continuity, this model is used less frequently than the others. Its equation is given by:

  $\gamma(h) = 1 - e^{-3\,(h/R)2}$

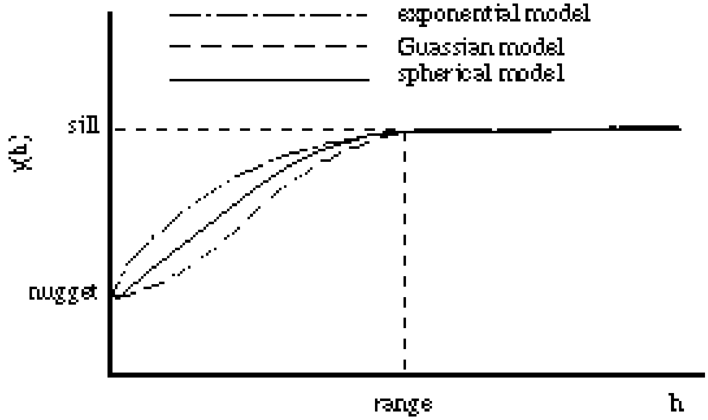


Figure 11-9: Three Commonly used Variograms Models - the Spherical, Exponential, and Gaussian Shown Here with the Same Range and Sill

Range

The range specifies the distance at which the variogram model achieves the sill (see Figure 11-9). When entering range values, higher numbers correspond to a high correlation between data points, whereas lower numbers indicate the opposite. The higher the range value, the further the weighting average for calculating missing values will extend. Since range is a directional feature, the range in the direction of increasing column numbers may be different from the range in the direction of increasing row numbers. Figure 11-10 shows an example where the column range is 12 while the row range is 2. The ratio of the column to row range is called the anisotropy ratio. This feature is most useful when the spatial continuity of the attribute under study shows stronger spatial continuity in some directions than in others. For example, the sample values in Figure 11-10 show a much stronger correlation in the direction of increasing columns than in the direction of increasing rows. Note that the kriged values also tend to reflect this anisotropy.

Sill

The sill is the plateau (if any) that the variogram reaches (see Figure 11-9). Usually the sill is constant for all directions. Note that the sill value is usually similar to the variance of the input data. Thus, the variogram sill can usually be standardized to a value of 1 by dividing by the variance of the input sample data.

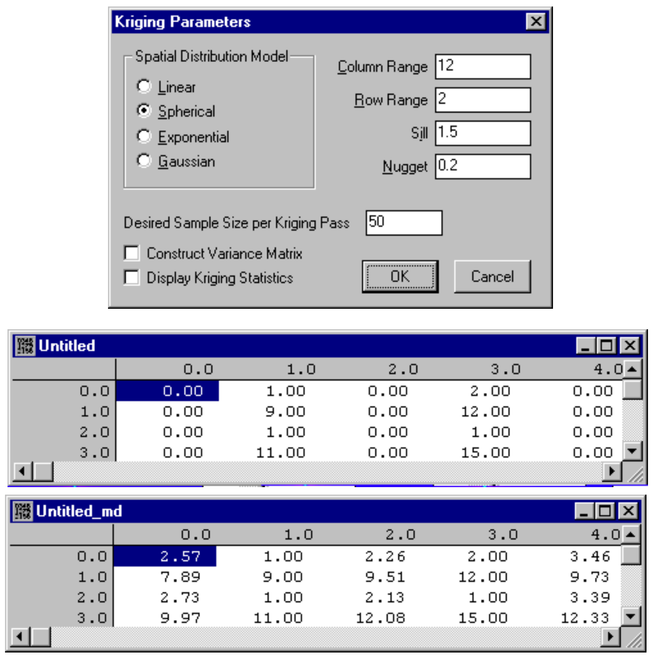


Figure 11-10: Anisotrophic Model

The examples above illustrates a matrix fill using kriging with an anisotrophic model.

Note

The Column Range, Row Range, and Sill fields must be greater than zero when Kriging. Also, the Nugget field must be greater than or equal to zero and less than the Sill value.

Nugget

The nugget effect is an apparent discontinuity at the origin of the variogram. This may be a combination of both sampling error and genuine short-scale variability at very small distances.

Desired Sample Size Per Kriging Pass

If the actual number of data points exceeds this value, the Kriging routine will then divide the dataset into multiple regions and krig each region individually. You should use the largest possible sample size to get the greatest accuracy. However, large sample sizes (>100) are very computational intensive.

Construct Variance Matrix

If selected Transform produces two datasets instead of one. The second dataset contains the kriging variance for each kriged point in the matrix. The variance data set has the name of the original data set with a '_var' extension added.

Display Kriging Statistics

If selected, a dialog of statistical information about the kriging operation is displayed.

Chapter 12:
Image Windows

Transform helps you look at your data using images, line graphs, and contour, surface, vector, polar (Macintosh), and histogram (Windows) plots. Each image window is built on a similar foundation and has similar tools. Throughout this chapter an image window is used to illustrate the attributes shared by all image types.

The Reference Manual dedicates one chapter for the discussion of each image type. The image commands and the chapters in which they are discussed are listed below.

- Generate Image (Chapters 12 and 14)
- Interpolated Image and Polar Plot (Chapter 14)
- Line Graph (Chapter 15)
- Contour Plot (Chapter 16)
- Vector Plot (Chapter 17)
- Surface Plot (Chapter 18)
- Histogram (Chapter 19)

Create Images and Plots

Images are generated by selecting the image command from the Image menu. For example, to generate a simple image, activate a dataset window, then select the **Generate Image** command from the menu. A new window displaying that image is automatically generated.

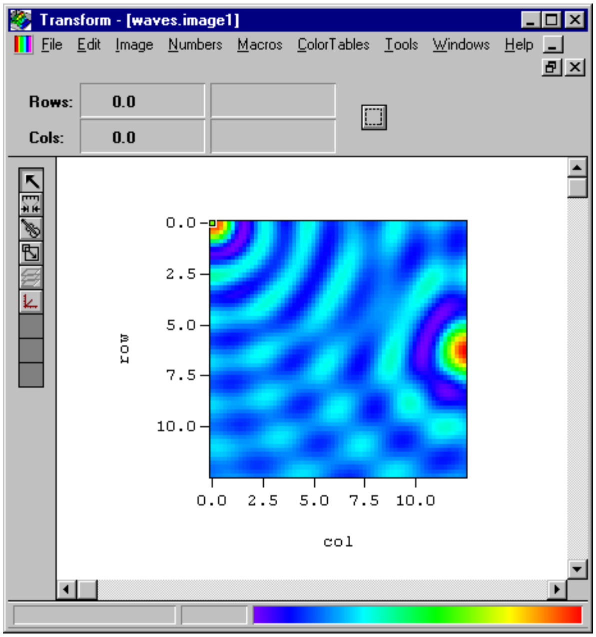


Figure 12-1: Simple Image

The window name in the title bar displays the name of the dataset that Transform used to create the image. It includes a suffix that matches the image type and a number (e.g., waves.image1). Beneath the image is the canvas—the imaginary surface the image rests on. Scroll bars control the position of your view onto the canvas.

The Tool Palette

The tool palette is located along the left-hand side of the image window. The tool palette has space for nine tool buttons: Select, MinMax, Fiddle, Resize, Overlay, Axis, Vector, Label, and Contour. These tools are shown in Table 12-1 and are described in detail in this and later chapters. The tools available depend on the image type.

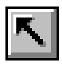
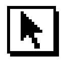
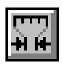
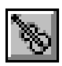
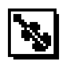
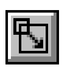
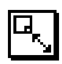
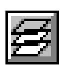
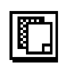
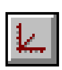
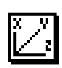
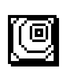
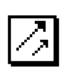
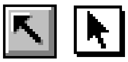
Windows Tool Icons	Macintosh Tool Icons	Tool Name	Description
		**Select**	Sets the selection region
		**MinMax**	Sets the minimum and maximum data values
		**Fiddle**	Manipulates the color table
		**Resize**	Resizes or moves the image region
		**Overlay**	Resizes or moves the overlay (if present)
		**Axis**	Controls axes labels and titles
		**Label**	Places text labels for levels on contour plots
		**Contour**	Places a level on contour plots
		**Vector**	Controls appearance of arrows on vector plots

Table 12-1: Tool Palette

The Select Tool



The default tool for all new image windows is the Select tool, which allows you to define a selection region for an image by clicking and dragging. All plot types, except the surface plot and histogram (Windows), allow selections that are coordinated between windows. Clicking the Select tool provides additional options in the toolbar (Windows) or the bottom margin of the image window (Power Macintosh).

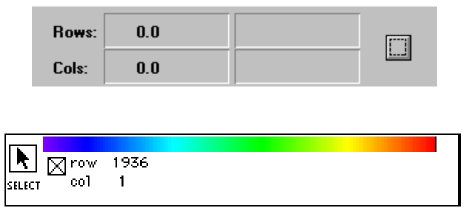
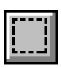


Figure 12-2: Select Tool Options (Windows and Power Macintosh)



In Windows, the coordinates of the current selection region are displayed in the toolbar. The **Show Selection** button in the Select toolbar toggles the display of the selection region in the image.

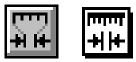
In Power Macintosh, the coordinates of the current selection region are printed at the left side. The checkbox next to the selection range toggles the display of the selection region. For all color plot types a color bar appears just below the image. From left to right, the colors in the bar represent increasing data values from the smallest to the largest displayed in the image.

Tip

Power Macintosh: to quickly return to the Select tool after using a different tool, use the Escape key (Esc).

The arrow cursor keys on the keyboard can be used with the Select tool to change the position of the selection region. Each arrow direction key (up, down, left, and right) moves the entire selection region one data element in that direction. The selection region size always remains the same size. If the region bumps against the edge of the array, the arrow key in that direction is ignored.

The MinMax Tool



The MinMax tool displays and allows you to change the minimum and maximum data values Transform uses for the color, height, contour, or vector scaling of the current plot. In Windows, any values outside the range shown are either ignored or replaced during the image generation process; in Power Macintosh, values outside the range are either skipped, ignored or smoothed over. The Windows toolbar and bottom margin of the image window (Power Macintosh) for the MinMax tool are shown in Figure 12-3.

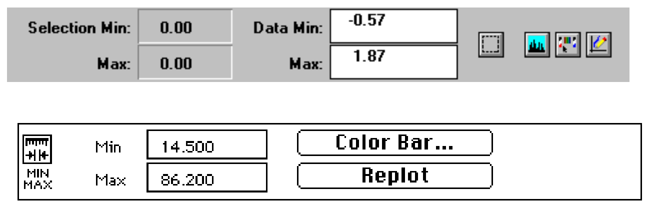


Figure 12-3: MinMax Options (Windows and Power Macintosh)

The MinMax tool allows you to specify values for high and low data outliers (Windows), reset data minimums and maximums, and update other windows. By specifying a minimum and maximum data value, you can focus your analysis on a particular range of interest. Transform allows you to associate a range with the dataset. All plots or images created will respect the dataset's range of interest. In addition, you can modify the range of individual plots or images. The range of interest is generally modified with the MinMax tool. (MinMax and outliers are discussed in more detail in Chapters 3 and 13.)

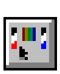
Windows MinMax Toolbar

In Windows, in addition to showing the range of interest, the toolbar shows the range of data values in the selection region.

Reset Data Min/Max

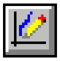
Clicking on **Reset Data Min/Max** resets the data min/max to the absolute min/max of the entire dataset.

Edit Outlier Colors



Editing the color values for high and low data outliers is discussed in Chapter 13.

Update All Windows



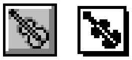
When you have more than one window open and want to change the data range in all of them simultaneously, click on the **Update All Windows** button. This changes the data range for all associated windows.

Power Macintosh MinMax Options

The bottom margin of the image window displays and changes the minimum and maximum data values Transform uses for the color, height, or vector scaling of the current plot. Edit the value range and invoke the new values with the **Replot** button (you also can activate the new values by pressing Enter). Transform recalculates the entire plot using the new min/max values, and then re-displays it.

When the **Color Bar...** button is available (all except vector plots) click this button to bring up a Color Bar window, the same as using the **Color Bar** command from the Image menu. Any new values you set in the color bar window while the Min/Max tool is active are copied to the active edit boxes in the image window. When you click **Replot**, the new values take effect.

Fiddle Tool



The Fiddle tool allows you to interactively change the current color table. The toolbar (Windows) and bottom margin of the image window (Power Macintosh) for the Fiddle Tool is shown below.

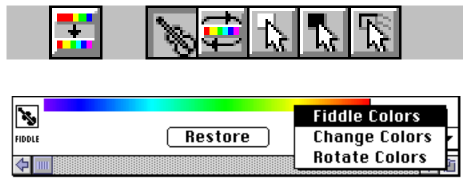
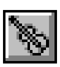


Figure 12-4: Fiddle Options (Windows and Power Macintosh)

Fiddle buttons for Windows and pop-up menus for Power Macintosh are discussed below.

Fiddle Colors

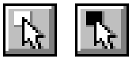


Fiddle Colors lets you play with the color mapping while looking for interesting aspects of your image. Technically, Transform takes the mouse position and remaps the current color table by compressing or shifting the color range. Clicking at the center of the image leaves the colors where you started. Clicking the mouse at (or dragging to) a position to the left or right shifts the color range left or right. Clicking or dragging above the center expands the color range. Conversely, clicking or dragging below the center line compresses the color range. With the GrayScale color table, the effect is quite like adjusting the contrast and brightness on a black-and-white TV screen. Note that while the color mapping is changing dramatically, the original data array is not affected in any way. The color scaling equation of original data to color numbers is unchanged; you are only changing the color assignments.

Rotate Colors

Rotate Colors shifts all of the colors to the left or right as you move the mouse. Click anywhere in the image and drag the mouse to the left or right. As the mouse moves, the colors move at the same rate. Note that the colors which fall off the right side of the bar are brought in at the left side (and vice versa).

Change Colors



Change Colors selectively changes any color or range of colors to white or black on Windows, or to white on Power Macintosh. Click on the image at any location to change that color entry.

If you drag the mouse after a click, Transform changes only the color under the mouse to white or black. Holding down the Ctrl key (Windows) or Shift key (Macintosh) restores the original color at the point under the mouse. Holding down the shift key while dragging has a cumulative effect—all the color numbers between the last pixel color and the current pixel color are changed to white or black.

Drag Colors (Windows only)

Drag Colors lets you use the mouse to take a color from any position in the color table and copy it to any other position in the color table. Click on a color in the image. With the mouse button depressed, drag the mouse to any other location in the image and release the button to drop the initial, copied color into the new location, replacing the color that was there.

Restore

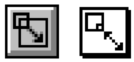
Restore changes the color table in your image to what it was before you fiddled with the colors.

Note

Power Macintosh: if you choose a new color table from the Color Tables menu, read in an image from disk, or load a color table from disk, the previous table cannot be restored.

When you save, only the modified table is saved with the image.

Resize Tool



This tool is used to change the size and location of the image. Figure 12-5 shows an example of a toolbar (Windows) and bottom margin of the image window (Power Macintosh) for the Resize tool. The current position and size of the image are shown. In Windows, you may type new values directly into any of these fields and have the image size automatically recomputed. Buttons provided access the image size dialog, fit the image to the current window size, and force the image aspect ratio (Windows).

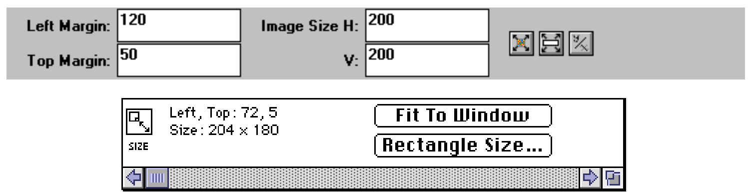
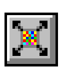


Figure 12-5: Resize Tool Options (Windows and Power Macintosh)

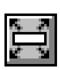
The resizing options for both Windows and Power Macintosh are described below.

Fit to Window



To change the size of the image to the currently visible canvas area in the window, first resize the window to the desired size, then click **Fit to Window**.

Rectangle Size



To set the image size of a currently active image, click on the **Rectangle Size...** button. The Image Size dialog shown in Figure 12-6 appears.

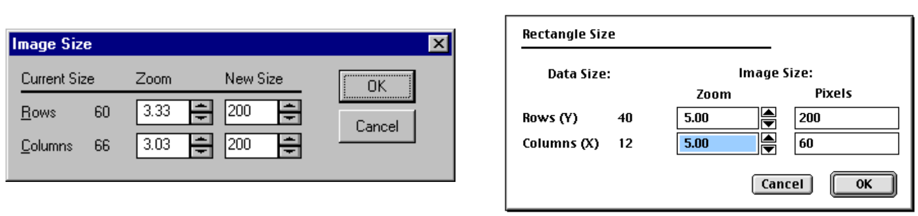
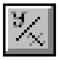


Figure 12-6: Image Size dialog

The Zoom boxes let you change image size by specifying an expansion factor to be applied to the dataset dimensions. The New Size (Windows) or Pixels (Macintosh) boxes let you specify image size directly; the values in these boxes match the image size displayed in the toolbar or at the bottom of the image. You can also set the size of a future image, by choosing **Image Size...** from the Image menu to invoke the Image Size dialog. The new size takes effect for the next image generated.

1. Fix Aspect Ratio (Windows)



The **Aspect Ratio** button changes the image size so that the aspect ratio of the image matches the dataset. The aspect ratio is calculated from the dataset's scale ranges.

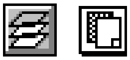
Interactive Sizing and Moving

In Windows, to interactively size the image, click on any of the eight black rectangles surrounding the image and drag the rectangle frame to a new size. When you release the mouse, the image is recalculated at its new size and location.

In Power Macintosh, click in the small white size box at the lower right of the image and drag the rectangle frame to a new size. When you release the mouse, the image is recalculated at its new size to fit the new frame. To maintain the aspect ratio of the image while resizing, hold down the option key during the resizing operation.

To move the image, click the mouse inside the image and drag the entire frame to a new position on the canvas. The size and location display are updated as you move and stretch the image.

Overlay Tool



The Overlay tool lets you paste Windows Meta File (WMF) or PICT (Power Macintosh) images on top of an image display. An overlay may be a collection of arrows, boxes, and other annotations, or the outlines of a map, or a plot created within Transform itself. Figure 12-7 shows the Windows and Power Macintosh options when the Overlay tool is selected.

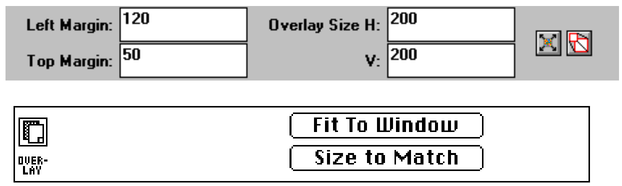
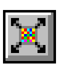


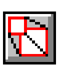
Figure 12-7: Overlay Options (Windows and Power Macintosh)

Fit to Window



Clicking the **Fit to Window** button changes the size of the pasted overlay to match the currently visible canvas area in the window.

Size to Match



Clicking the **Size to Match** button changes the size of the pasted overlay to match that of the underlying image. The size and location fields are updated as you move and resize the overlay.

Note

In the case where contour plots are overlayed onto images, the contours will be half (1/2) a pixel smaller than the image on the edges.

Creating Overlays

Window Meta File (WMF) or PICT (Macintosh) images can be copied to the Clipboard and pasted on top of an image display in Transform. Though overlays can be pasted from other applications, an overlay is typically a plot from another Transform window.

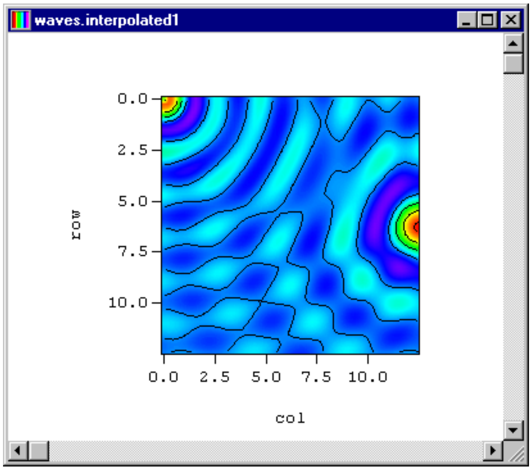


Figure 12-8: Overlay of a Contour Image on an Interpolated Image

The image used for an overlay can be any image copied to the clipboard. After copying the desired image, select the image window that will receive the overlay and choose **Paste**/**Paste Overlay** from the Edit menu. Transform draws the overlay on top of the existing image.

You can also display multiple overlays—a vector plot and a contour plot on a raster image, for example.

See the discussion of the Overlay toolbar earlier in this section for information on how to resize overlays.

Delete Overlay

To delete an overlay in the currently selected image window, select the Overlay tool then choose **Clear/Clear Overlay** from the Edit menu. In Power Macintosh, you can also press the Delete key from the keyboard. In some modes, the Delete key is being used for other functions, so you may have to switch to the Select tool first.

Raster Overlays

Some overlays, notably raster images created in Transform, are opaque. Because they completely obscure anything drawn beneath them, they are not very useful as overlays.

Color in Overlays (Power Macintosh)

When the Macintosh display is set to 256 colors, a raster image created with Transform generally uses all 256 colors. There are no colors left over to share with the overlay, so a color overlay, whether it contains color lines or a raster image, may not be drawn with authentic colors. In cases like these, the colors are approximated to the closest available colors.

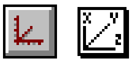
Either match the required colors between your color overlays and the image underneath or use a monitor setting with more than 256 colors (usually 24-bit color with 16.7 million colors). With black and white line drawings, you should not experience any of these problems.

Double Overlays (Power Macintosh)

An example of a double overlay would be a contour plot on top of a vector plot on top of a raster image. Since there is only one overlay displayed per image window, a second step is required to accomplish a double overlay. Start with the top layer, the contour plot, in this example. Prepare the plot and copy it to the Clipboard. Now prepare the vector plot and paste the contour plot into the vector window as an overlay. With the Selection tool active, copy the contents of this window to the Clipboard; the contour plot is copied along with the vector plot. Next, generate the raster image and paste the double overlay into the new window.

A similar maneuver can be used with an external drawing program to double up on maps and annotations. Add another layer to the double overlay to create a triple overlay or even more. Chapter 6 walks you through an example of multiple overlays.

Axis Tool



The Axis tool controls the appearance of the title, axes, and color bar annotations in the image window. The toolbar (Windows) and bottom margin of the image window (Power Macintosh) for the Axis tool is shown in Figure 12-9.

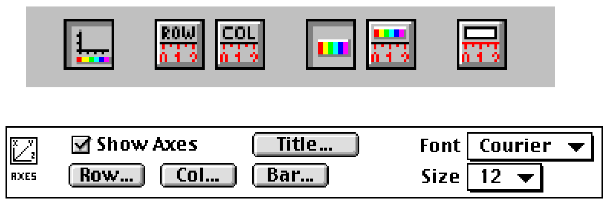
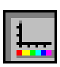


Figure 12-9: Axes Options (Windows and Power Macintosh)

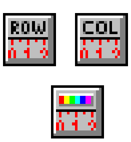
Surface plot windows have their own separate axis controls, described later in the section on surface plots.

Show Axes



The **Show Axes** button/checkbox toggles axis annotations for the image on and off, including the title, tickmarks, axis labels and the color bar (especially useful for creating overlays).

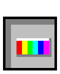
Row, Column, and Color Bar Axis Labels



Click the **Row** button, **Col** button or **Bar** button to change the axis labels or tickmarks for the corresponding axis. The dialogs that open when you click any of these buttons is discussed later in this chapter.

In Power Macintosh line plots, the **Bar...** button changes to **Data...** to allow modification of the data axis labels. Depending on the type of line plot, horizontal or vertical, either **Row...** or **Col...**will be inactive (grayed out). **Bar...** is unavailable for plots which do not use color.

Enable Color Bar (Windows only)



Change the toggle setting for Enable Color Bar to turn on and off the display of the color bar in the image window.

Image Title

Click the image **Title**... button to bring up a dialog for entering the plot title. The title will be displayed in the current font and text size. It will be centered over the image.

Font and Size (Power Macintosh)

The **Font** and **Size** popup menus set the default font for the window. This setting applies to the axis labels, axis names, and the image title

Axis Labels Dialog

Clicking one of the row, column or color bar axis buttons from the Axis tool allows you to edit all of the characteristics of that axis in the Axis Labels dialog, shown in Figure 12-10.

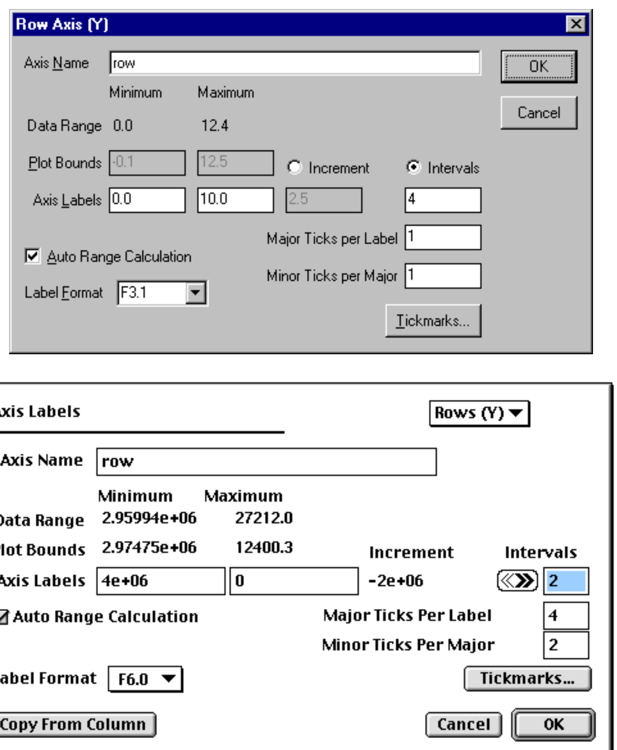


Figure 12-10: Axis Labels Dialog (Windows and Power Macintosh)

Axis Name

This name is displayed next to the axis labels. For the horizontal axis, the name appears just beneath the labels. For the vertical axis, the name appears rotated 90 to the left of the labels.

Data Range

Displays the actual range of values for this axis. This cannot be changed.

Plot Bounds

Displays the range of values for this axis that are plotted. These numbers cannot be changed from the Axis dialog for the Row(Y) and Column(X) axes on color images, contour plots, and vector plots. If you want your data to be scaled differently, choose **Generate Scales...** from the Numbers menu.

You can change these numbers on the Row and Column axes for surface plots, where you can use it to truncate your surface plot. The plot bounds for all color bars, and for the data axis on surface and line plots, are identical to the data minimum and maximum values available from the MinMax tool.

Axis Labels

Displays the starting and stopping value for axis labels. You can edit these values to start and stop labels at values other than the plot boundaries.

Axis Label Increment and Intervals

These boxes let you enter either the spacing for axis labels in the increment box or the total number of intervals in the intervals box. One box automatically updates the other when changed. In Power Macintosh, click the double-arrow between the boxes to switch the active box.

Auto Range Calculation

When Auto Range Calculation is enabled, Transform will calculate the axis minimum and maximum values for you. Transform uses the desired number of intervals and the plot bounds to calculate a nice increment and bounds for the labels. With Auto Range turned on, you will be able to change the number of intervals, but you cannot change the label bounds or the increment.

Note

If Auto Range Calculation is turned on, you can still enter a number of intervals. However, your actual number of intervals may turn out to be slightly different than what you entered, so the label spacing stays at a 'nice' number

Major Ticks Per Label

This box lets you specify the number of intervals to divide each label interval. These are marked with major tickmarks.

Minor Ticks Per Major

This box lets you enter the number of intervals to divide each major interval. These are marked with minor tickmarks. The first minor tickmark is drawn at the major tick, so it is not visible.

Label Format

You can enter a custom format in the box or select a format from the combo box/pop-up menu.

Tickmarks...

The **Tickmarks**... button provides access to the Tickmarks dialog, shown in Figure 12-11. This dialog lets you set size and style of tick marks and grid lines.

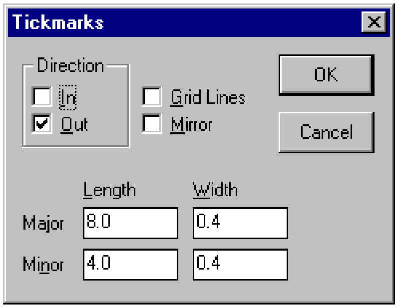


Figure 12-11: Tickmarks dialog

In - Clicking this check box draws tickmarks from the axis toward the center of the plot (inside tickmarks).

Out - Clicking this check box draws tickmarks from the axis away from the plot (outside tickmarks).

Grid Lines - Clicking this check box draws a grid line across the plot for each major tickmark.

Mirror - Clicking this check box draws identical tickmarks on the opposite edge of the image area (top and bottom, left and right).

Major and **Minor -** Entering values in these text fields sets the lengths and widths for major and minor tickmarks, in points. The sizes may be in fractions of a point (e.g. 0.5 point).

Copy From Column (Power Macintosh)

Click this button to copy the printing format, label intervals, major intervals, minor intervals and tickmark characteristics from another axis record. If you are editing the Row axis, then the button copies information from the Column axis, otherwise the button copies from the Row axis.

Other Tools

Three additional tools apply to specific plot types, and are discussed in the chapters for those plots. These are the Vector tool, Label tool, and Contour tool.

Chapter 13: Color and Color Tables

Most of Transform’s images and plots use color to represent data values. This chapter describes the color options and use of color in Transform.

Visual Data Analysis

Grayscale images reflect a very simple way of displaying data values visually. Higher numbers are assigned lighter shades of gray and lower numbers are assigned darker shades of gray. The maximum data value in a dataset maps to white, and the minimum value maps to black. Because your eye recognizes this scale of intensities, it becomes an intuitive system for interpreting, at a glance, the relative data values in the array. The color table for this mapping is named 'GrayScale'.

People are also accustomed to interpreting the rainbow of colors that make up the visible light spectrum. Hence, images generated in Transform are assigned the 'Rainbow' color table, by default. This table has a smooth transition from indigo and blue colors that represent low values, to orange and red colors that represent high values. Other color tables may not be so intuitive but can be equally powerful as you interpret data. The 'Rainbow-Banded' color table has darker bands of color equally spaced along the spectrum to help create a contour effect. The 'Purple Haze' table has four cutoff points to mark specific data values of interest.

In raster images, each pixel in the image is a color derived from a data value in the array. Surface plots can use color to show the data values from one array combined with a surface formed from the data in another array. Contour plots can be drawn in color to reinforce the data value represented by each contour line.

Color Scaling Equation

Transform uses a linear scaling equation to calculate color numbers from the array of data values. Colors on the left side of the color bar are mapped to the low values and colors on the right side of the color bar are mapped to the high values of the array.

As an example, take the case of color numbers assigned to shades of gray proceeding from color number 1 (black) to color number 254 (white), and a data array with numbers between 100 and 25,400. The number 21,824 would be assigned color number 218 (light gray). The number 154 would be assigned color number 1 (black).

Simple Version - Scaling Equation

A simple scaling equation that produces these mappings is shown below. Using the default color range of 1 to 254, the calculation of the colors to use for each pixel is evaluated for every pixel in the array. The **maximum** and **minimum** values referred to in the equation are the maximum and minimum values in the data to be scaled.

```
range = maximum - minimum
colorvalue = 1 + 253 * (datavalue - minimum) / range
```

The assignment of a color table determines which colors you see on the screen. The color table matches a visible color with each of the numbers from 1-254. In Windows, if a calculated color value is <1, it is treated as a "low outlier." If the value is >254, it is a "high outlier." Outliers are handled specially and are assigned black, white, the first color from the table, or the last color from the table (see below).

Full Version - Scaling Equation

The full scaling equation includes the minimum and maximum color values. The full 256 colors from 0-255 are not required to create images. You can change minimum and maximum color numbers to a smaller range. The final equation produces a linearly scaled set of color values from **color_min** to **color_max**. In Windows, outliers are any data values which fall outside of this color range.

```
range = maximum - minimum
colorrange = color_max - color_min
colorvalue = color_min + colorrange * datavalue - minimum) / range
```

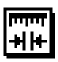
For interpolated images, each pixel color is determined by the same equation. Before the pixel is colored, a smoothed value is calculated with bilinear interpolation. This interpolated data value calculates the color instead of the exact data values in the grid.

Modify the Data-Image Correspondence

Here are four ways you can change how colors are used to represent data values:

- Change the data min/max
- Transform the data using notebook calculations
- Choose a different color table
- Change the colors used for low and high outliers

Change Data MinMax

If some of your data values lie outside your range of interest, you can change the data min/max settings. Outliers will not be eliminated from the dataset—just handled specially during image generation. Use the **MinMax tool** or Color Bar window (Power Macintosh) to change these values.

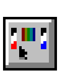
Transform Data

You can transform the data using built-in expressions and the **Calculate From Notes** command under the Numbers menu. For example, if you wanted logarithmic scaling of the colors in your image, you would need to transform the data using Transform's `log( )` function (since color is scaled linearly). See Chapter 22 for more detail.

Choose Different Color Table

Choosing a different color table changes the colors you see in the image. The scaling of data values to color numbers is not affected by the choice of color table; only the matching of color numbers to the colors changes. Choose the table that best fits the data you are trying to analyze. Use the ColorTables menu or the **Custom Color Tables...** (Windows) or **Load Color Table...** (Macintosh) command to change color tables.

Edit Outlier Colors (Windows)



For outliers, data values outside the selected min/max range, you may choose how those values are displayed in images. Under the MinMax tool, click the Edit Outlier Colors button to bring up the dialog shown in Figure 13-1.

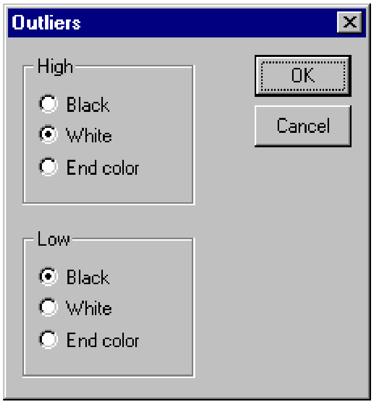


Figure 13-1: Edit Outlier Colors dialog

For low outliers (values less than the minimum) and high outliers (values greater than the maximum), you may choose black, white, or end color to display those values. End color is the first or last color in the current color table. Low outliers use the first color, high outliers use the last color.

Note

Whenever you have outliers, there are multiple data values being displayed with the same color. Take this into account during analysis.

Color Tables Menu

The mapping of numbers to colors can be changed in any image with the use of the Color Tables menu. Select any one of the color tables from this menu to assign new colors to the currently active image.

Figure 13-2: ColorTables Menu (Windows)

Many color tables are available in the Color Tables menu; some are briefly discussed below.

Note

Power Macintosh: 24-bit images imported into Transform are resampled to 8-bit using the **24-Bit Sampled** color table on the Color Tables menu. to get the most accurate approximation of the original image, use this color table.

GrayScale—From black to white, increasing levels of gray.

Rainbow—The color spectrum or hue that includes the following colors: blue, cyan, green, yellow, orange, red.

Rainbow-Banded—The rainbow table with bands of darker colors spaced throughout the spectrum.

Apricot, Ether, Lava Waves—Each of these tables contains many colors with banding effects to bring out contours in the data. They can be misleading in some cases, but can also be effective in bringing out hidden data features.

Purple Haze—An example of a table with sudden transitions in the scale, which mark specific cutoff points in the data.

Note

To get more than one image to reflect the same colors, select the same new color table for each one of the images. When two images are set for different color tables, the colors in the front-most image window will take precedence; the colors for the other image window are ignored. This does not apply to monitors set to greater than 256 colors. 16 million (24-bit) colors can simultaneously display multiple images with different color tables.

Most Recent Custom Color Table (Windows)

The second entry in the Color Tables menu contains the name of the most recently used custom color table. As described below, the Fiddle tool, opening a saved file, and importing a color table, all can create a custom color table. Any time one of these tables becomes active, its name appears in the menu. It remains in the menu until another custom color table becomes active.

Custom Color Tables... Command (Windows)

Custom color tables are created by using the Fiddle tool, opening certain saved files, or importing color table files. The **Custom Color Tables**... command keeps track of and gives you access to these tables. Transform remembers the list of custom color tables until you exit the program.

Using the Fiddle tool modifies the color table used for the current image window. This modified color table is named and kept as a custom table. Color tables modified in Fiddle mode are named 'Windowname.fiddle', where 'Windowname' is the name of the window used. Rotate, Change to White, Change to Black, and Drag modes create a new color table with a name of the form 'Windowname.change'.

Any color table imported from a file (see Import below) is also remembered as a custom color table. And, when you open a file with a color table other than one of the built-in tables, that table is kept.

Choose **Custom Color Tables**... from the Color Tables menu to bring up a list of stored custom tables in a dialog, as shown in Figure 13-3.

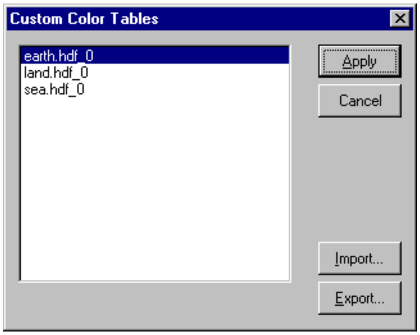


Figure 13-3: Custom Color Tables dialog

This dialog lists all of the custom color tables you have created or accessed since running Transform. Select any table from this menu and click Apply to apply this color table to the current image window. When you do apply the table, it appears as the second entry in the Color Tables menu.

Import...

Transform can load new color tables from files on disk. Included with Transform is a directory with several more example tables to choose from. Click the **Import**... button in the dialog to identify the file to open. These files may be HDF or binary color table files (see below for more information).

Export...

Transform can also export any of these tables. Select the table you wish to export. Click the **Export**... button to save a copy of the table in a file. You may export either HDF or binary color table files by choosing the type from the save file dialog.

HDF Color Tables

Color tables are stored in HDF files as a separate type of data object, whether or not any other kinds of data are in the file. The files in the color tables examples directory are HDF files which contain only one color table each.

When Transform saves an image window, a color table also is saved with the file. Therefore, color tables can be obtained from HDF data files created by Transform and other programs. As long as an image has been saved in a Transform save file, Transform can retrieve its color table from the Custom Color Tables dialog.

Binary Color Tables

If the color table file is not an HDF file, Transform assumes it is a binary color table file. These files must be unsigned byte, exactly 768 bytes in length. The first 256 bytes represent the red color table entries, the second 256 bytes the green, and the last 256 bytes the blue. These numbers are interpreted as 8-bit components for a 256-entry look up table.

Load Color Table... Command (Macintosh)

Transform can load new color tables from files on disk. Included with Transform is a folder with several additional example tables to choose from. Use the **Load Color Table...** command from the Color Tables menu to identify the file to open. These files may be HDF, KLUTZ, or binary color table files.

HDF Color Tables

Color tables are stored in HDF files as a separate data object, whether or not any other kinds of data are in the file. The files in the color tables examples folder are HDF files which contain only one color table each.

When Transform saves an image, a color table also is saved with the file. Therefore, color tables can be obtained from HDF data files created by Transform and other programs. As long as an image has been saved in a Transform save file, Transform can retrieve its color table with the **Load Color Table...** command. If the file contains more than one color palette, Transform loads the first palette it encounters.

KLUTZ Color Tables

Transform can load a color table from any color table file with the file type of 'CLUT' (the type saved from the desk accessory called KLUTZ). These files contain color table resources of type 'clut' that Transform can read and apply to the current image.

Note that he KLUTZ color tables mentioned here are identical to the PICT color tables that you can save in the View Utility (described later in this chapter).

Binary Color Tables

If the color table file is neither HDF nor CLUT, Transform assumes it is a binary color table file. These files must be exactly 768 bytes in length. The first 256 bytes represent red color table entries, the second 256 bytes green, and the last 256 bytes blue. These numbers are interpreted as 8-bit components for 256-entry look up table.

Macintosh System Color Table

Some programs work best when the image uses the Macintosh default system color table (provided on disk), but this table does not contain a smooth gradation of color needed to be useful for most Transform imaging. Other color drawing or presentation programs use different predefined color tables, so in the sample color table files provided with Transform we have included some for matching with these programs.

Color in Images and Plots

Raster images, surface plots and contours all use color in the manner just described. Raster images are based on the idea of representing values with color; each pixel is displayed using the color associated with an array data value.

Color in Surface Plots

Color is optional in surface plots because a perspective view of height is used to represent data values. The colors can be drawn from the same dataset, reinforcing the surface height with color, or another dataset may be used to show two arrays of data at the same time: one array contributing the values for the surface height and another array contributing the data values for the color. For more information on the use of color in surface plots, see Chapter 18: Surface Plots.

Color in Contour Plots

Color also is optional for contour plots. The lines can be colored according to their given data values. For more information on color in contour plots, see the Chapter 16: Contour Plots.

Color Bar Window (Macintosh Only)

Because Transform uses color to represent data values from your 2D array, the matching of colors to numbers becomes very important. The Color Bar window makes the process of generating a color image from the array of numbers easier to comprehend, and lets you set specific scaling factors to be used in the process.

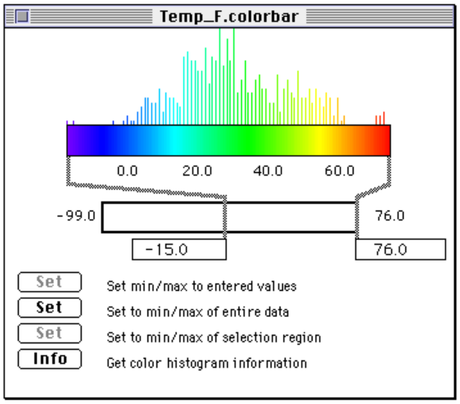


Figure 13-4: Color Bar Window

To create a Color Bar window for the current dataset, select **Color Bar** from the Image menu. Transform first calculates a histogram of the distribution of data in the dataset, then displays it in a new window. When you change the maximum and minimum settings as discussed below, the histogram is recalculated from the source data and is re-displayed as necessary.

Another way to access the Color Bar window is from the **MinMax tool**. A button for it is provided in the bottom margin of an image window when the MinMax tool is active. A color bar invoked from an image window is no different than one generated using the menu command. The parts of the Color Bar window are discussed in the following paragraphs.

Color Bar

The color bar is the central feature of the Color Bar window. This sequence of color lines represents the smallest numbers in the dataset on the left, continuously increasing to the largest numbers on the right. The labels below the color bar are automatically scaled to the values in the data to display how the numbers in an image are to be converted into colors when the image is generated.

Histogram

Above the color bar is a histogram of the distribution of data values in the dataset. It is scaled to fit the space provided, with the height of each color line representing its relative frequency of occurrence in the data. The histogram is automatically redrawn when changes are made to the region of interest.

Min/Max Controls

Below the color bar are the controls for the three methods of setting the minimum and maximum values for the region of interest. Changing these settings lets you control color scaling for image creation and for the histogram display. There is more information about the manipulation of min/max settings later in this chapter.

Note

If you have a large dataset, you can press Command-period to abort the calculation of the histogram in the Color Bar dialog. If you do this, not only will the displayed histogram be incorrect, but the displayed Min, Max values will be incorrect.

Color Information Dialog

Click **Info** to bring up the Color Information dialog shown in Figure 13-5. This dialog documents the specifics about the histogram in the Color Bar window.

Number of points	Number of rows times the number of columns
Number of low outliers	Number of data points that fell below the minimum value selected for the histogram
Number of high outliers	Number of data points that were greater than the maximum value selected for the histogram
Most frequent color #	Color in range 1-254 used the most often according to the scaling calculation used by Transform
Frequency scale	Number of points that were mapped to the most frequent color number

.

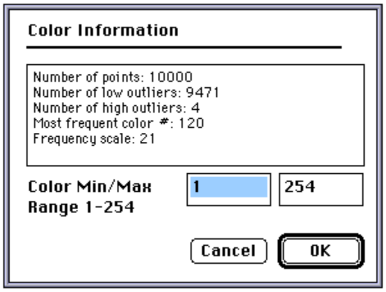


Figure 13-5: Color Information Dialog

Color Min/Max

Also in the Color Information dialog are settings for the minimum and maximum color numbers to use. For most operations, you should use the full capabilities of your Macintosh's 256 color screen—the range from 1 to 254. This is the maximum range of scaled values Transform can use because the generation of images uses one byte of memory per screen pixel. The usable color range does not include color 0 or color 255.

Note

If the color min/max is set to other than 1/254, then labeled color bars will be displayed with incorrect colors.

To increase compatibility with other programs on the Macintosh or to create special images, you can reduce the number of colors you want to use in the images generated in Transform.

Note

When importing PICT files, the best image is generated when the data min/max matches the color min/max. You can set the data min/max of an image using the Color Bar window. You can set the color min/max using the Color Information Dialog. Note that changing the color min/max affects only new images, not current images.

Copy Color Tables to/From Color Bar Window

Color tables can be copied and pasted to and from the color bar window. With a frontmost Color Bar window, choose **Copy Colors** from the Edit menu to copy a color table to the Clipboard. This table can be pasted into other image and Color Bar windows.

Save Color Bar Information

The significant settings from the color bar window (color table used and min/max settings) are saved when an image window is saved. However, the color bar window itself is not stored when the dataset is saved; nor is it automatically recreated when the dataset is opened.

Note the Color Bar window is not included when windows are tiled.

Min/Max Data Value Settings

This section explains how you use the Color Bar window to set the minimum and maximum values for your region of interest. Setting min/max values directly controls how the 254 colors in the color table are to be used to create meaningful images.

For most datasets, Transform finds the minimum and maximum values and uses the color range 1 to 254 for all images. When an image is created, the resulting linear scaling of values covers all of the values in the dataset. None of the values are considered outliers because they all are within the min/max range.

With the data min/max settings, you can spread colors out across your region of interest and not waste any on outliers that are uninteresting or wrong.

Example - Suppress Outliers Display

For example, suppose you have a temperature-monitoring instrument that reports temperatures from 65 to 72 degrees in your office building during the week. This week, it failed for an hour and reported 0 degrees for several readings. When Transform finds the minimum and maximum values, it will report 0 and 72 as the limiting values, and scale all of the useful readings at the upper end of the range of colors. To prevent this, use the Color Bar window to set the minimum value for consideration to 65 degrees. Now the image generation process can assign colors in the image more appropriately.

Mask Outliers

In color images, values that are below the region of interest (low outliers) are assigned color 1, the bottom color on the scale. Values that are above the maximum (high outliers) are assigned color 254. The outlier areas of the resulting image will be represented by just those two colors, yet may represent many different data values. If this causes problems, reset the region of interest to the minimum and maximum values of the entire dataset.

Note

If you need to make sure that the color outliers are unique, set the color range in the Color Information dialog to [2, 253] so that no values inside the region of interest will be assigned the same colors as the outlier values.

User Larger Region of Interest

It is possible, sometimes preferable, to set the minimum and maximum to values outside the range of data in the dataset. Settings these values to exact numbers in the color bar window gives you direct control over the scaling equation. When you choose a larger range, some colors in the color bar will be unused. The histogram shows which color numbers become unused in this case.

Min/Max Controls - Direct Setting

Below the color bar in the Color Bar window are the controls for two methods of setting the minimum and maximum values for the region of interest.

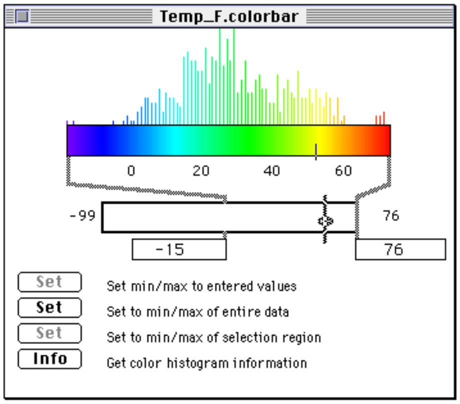


Figure 13-6: Color Bar Window

The first method is a direct setting on the box provided in the center of the window. The left side of the box represents the smallest value in the actual dataset (computed for you) and the right side is the largest value. Click on either the minimum or the maximum marker. Hold the mouse button, drag the marker to its new location, and release. You should be able to see a floating marker as you drag it from side to side. The setting takes effect immediately and the histogram will be redrawn according to the new value.

While you are dragging across the screen, any values that fall within the current color bar are reflected on the color bar with a small indicator. This may help you adjust your settings and line up your desired values with particular locations on the histogram.

Min/Max Controls - Second Method

A second method for changing the min/max settings involves the two edit boxes below the color bar.

Displayed in the left edit box is the current minimum value of interest, and in the right box is the current maximum value of interest. Click on either box to activate it, then enter your desired minimum or maximum value. For these new values to take effect, you must press the Set button.

Using the Selection Region

Select the region of interest in the data or image window then use the Color Bar window's Set button to set the min/max range to the smallest and largest values in the selection region.

Note

The first button may be dimmed because you have not yet entered any values into the edit boxes.

View Utility (Power Macintosh only)

The View Utility, located in the 'Transform:Extras' folder (see the on-line manual in the 'Manuals' folder), displays images (HDF, PICT) and PICS animations. The View Utility also contains a color table editor for creating and editing color tables. You can save them as HDF files and apply them to Transform images using the **Load Color Table** command from the Color Tables menu. You can also save color tables as binary files, import them into Transform and add them permanently to the Color Tables menu using the macro language. See Appendix G: Troubleshooting for specific instructions.

Chapter 14:

Raster Images

In raster images, each data point in the array is assigned a color according to its value. Then, as in paint-by-numbers, each pixel in the image is filled in, one color per data value. Raster images are the basis of discussion in Chapter 12. Please refer to that chapter for more information.

Types of Raster Images

Transform can create the following types of color raster images: simple, interpolated and polar (Macintosh). Note that line graphs, histograms, surface, contour and vector plots are described in separate chapters.

Simple Image

Once your dataset is loaded, simple images like Figure 14-1 are created with the **Generate Image** command in the Image menu.

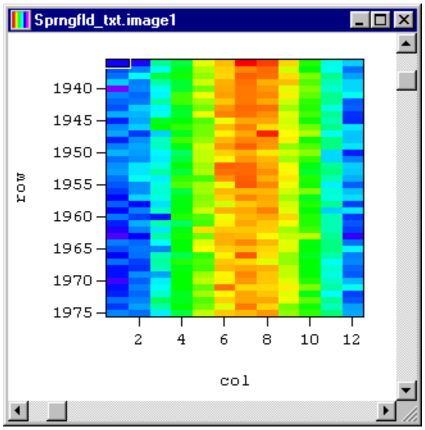


Figure 14-1: Raster Image using sprngfld_txt

When the data array has fewer points than the number of pixels to display, each data point is enlarged to a rectangle of pixels. For example, one data value may become a square of 3 x 3 pixels, all set to the same color. This creates a chunky or jagged effect in simple images.

You can count one block of color for each data value. When pixels are replicated, Transform takes into account the distances between points on the grid. Most datasets have evenly spaced numerical rows and numerical column labels so there is no unusual effect; the blocks of color are all the same size. When you use unevenly spaced scales, the blocks are larger where the scale values indicate the data points are further apart and smaller where they are closer together.

Interpolated Image

The **Interpolated Image** command in the Image menu creates from the current dataset an interpolated image similar to the simple, scaled image. The same paint-by-number process is used where each data value is mapped to a particular color chosen to represent that value. The difference is that if the image is enlarged, bilinear interpolation causes a gradual transition in the colors of the intervening pixels between the data points so there is no chunky effect.

The interpolated (smoothing) effect takes into account the distances between points on the grid when the numerical row labels or numerical column labels are unevenly spaced. Where the scale values indicate the data points are farther apart, more interpolated pixels are placed between them. Where they are closer together, fewer smoothed pixels are used.

Figure 14-2 shows an interpolated raster image generated from ‘sprngfld_txt’, the same dataset as used in Figure 14-1. Unlike the simple image, in the interpolated image, the blocks of color for data values are smoothed together.

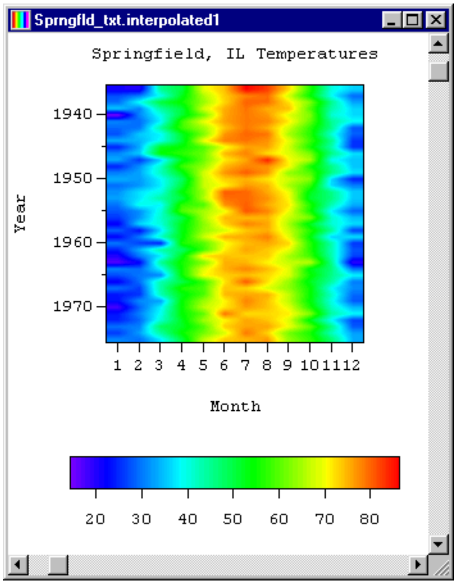


Figure 14-2: Interpolated Image using sprngfld_txt

Polar Image (Power Macintosh only)

For creating polar images, use the Polar Image command from the Image menu. These images use the simple, scaled method of generating the image with no additional smoothing. To create the polar effect, the pixels are placed according to a polar interpretation of the row and column scaling information. The row scales are treated as the set of radius values and the column scales are treated as the set of angle values in radians. By default, the angle is measured clockwise from the horizontal axis, with the origin at the center of the window.

The angle values in a polar image are limited to a range of 2 , one full circle. An example is shown below.

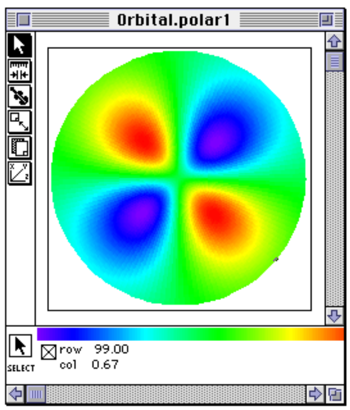
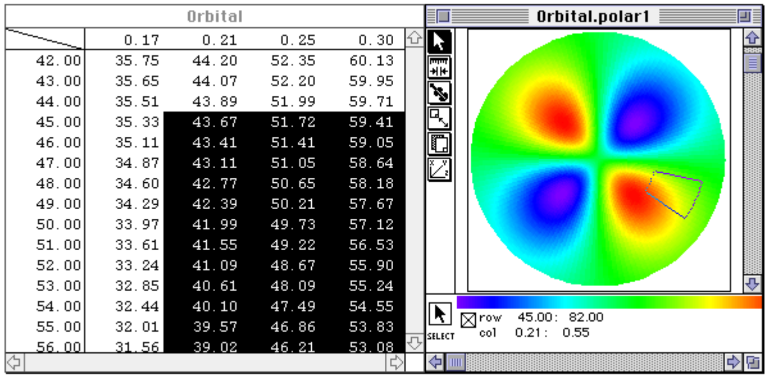


Figure 14-3: Polar Image

Select Polar Image Values

The Select tool for rectangular images is described in the image window section. Polar images, however, have a different shape for the selection area.

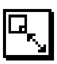
When you select a region of data values from a data window, an annular region is highlighted in the associated polar image, rather than a Cartesian rectangle, as shown below.



	0.17	0.21	0.25	0.30
42.00	35.75	44.20	52.35	60.13
43.00	35.65	44.07	52.20	59.95
44.00	35.51	43.89	51.99	59.71
45.00	35.33	43.67	51.72	59.41
46.00	35.11	43.41	51.41	59.05
47.00	34.87	43.11	51.05	58.64
48.00	34.60	42.77	50.65	58.18
49.00	34.29	42.39	50.21	57.67
50.00	33.97	41.99	49.73	57.12
51.00	33.61	41.55	49.22	56.53
52.00	33.24	41.09	48.67	55.90
53.00	32.85	40.61	48.09	55.24
54.00	32.44	40.10	47.49	54.55
55.00	32.01	39.57	46.86	53.83
56.00	31.56	39.02	46.21	53.08

Figure 14-4: Polar Data Selected

Polar Image Size



The Resize tool for rectangular images is described in the Image Reference chapter. For polar images, the options displayed in the window include a **Circle Size...** button instead of a **Rectangle Size...** button. These options correspond to the **Rectangle Size...** and **Circle Size...** commands in the Image menu.

Note

Changing the image size interactively using the size tool on Polar images changes not the image size but the clipping region. Click on the **Circle Size...** button to change the size of the image.

Polar data is defined on a circle with angles (columns) from zero to 2 ,and radius values (rows) from zero to the maximum radius. When you choose to change the default size for a polar image that you will be generating, you choose an expansion factor that determines the final size of the image. In addition to changing the size of a polar image, you also can choose the direction of the zero angle and the portion of the circle to view.

1. Choose **Circle Size...** from the Image menu or click on the **Circle Size...** button from the size tool for a polar image. The Polar Size Selection dialog box appears (Figure 14-5).

2. If you want to change the direction for the zero angle, select one of the direction buttons (North, East, South, West). Angles proceed clockwise from the selected direction.

3. Drag out a rectangle within the picture frame to select a viewing region. When you generate the image, the relative portion of the image inside your viewing rectangle is visible in the window. The size of the circle in this window represents the scaled size of the maximum radius value (row label) given in the dataset.

4. Specify an expansion factor to be applied to your viewing rectangle, by selecting the up or down arrow labeled ‘Expansion Factor’ or by entering a positive value in the text box between these arrows. For your reference, the numbers labeled ‘Window Width’ and ‘Window Height’ indicate the size of the resulting window.

5. Select **OK** or press Return to set the new size. If this was accessed from a menu command, the new size takes effect for the next polar image to be generated. If accessed from the Resize tool in a polar image window, the new size takes effect immediately and the image is redrawn.

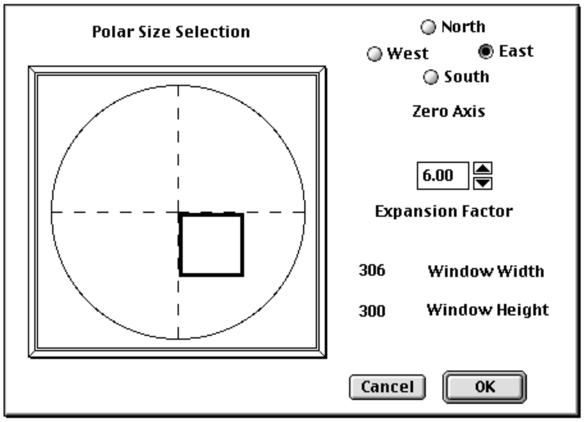


Figure 14-5: Polar Size Selection Dialog

Chapter 15:

Line Graphs

Transform's line graphing capabilities plot either a row or column of data from the dataset versus the scales for the appropriate axis. The values of any row or column of the dataset can be plotted by selecting the desired row or column.

Create a Line Graph

Choose **Line Graph** from the Image menu to open a line graph window. A line plot representing a row of data values from the array is initially drawn in the window; the scales printed on the horizontal axis are from the row or column scale values, and the scales printed on the vertical axis reflect the current minimum and maximum settings.

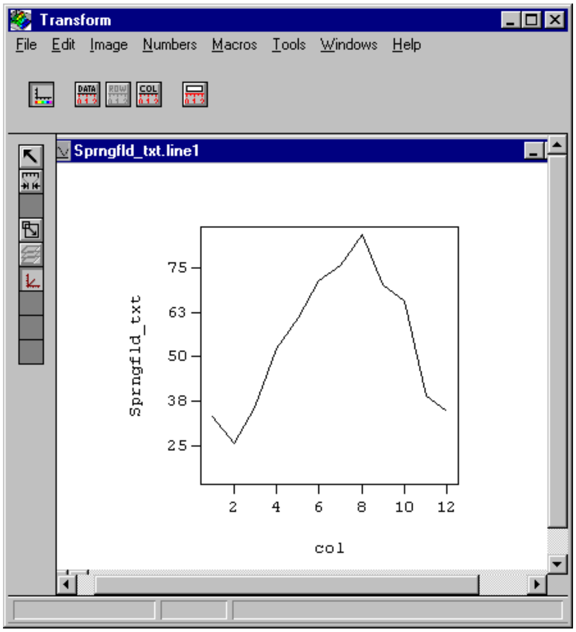
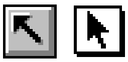


Figure 15-1: Line Graph

Select Tool



When the Select tool is active, the toolbar (Windows) or bottom margin of the image window (Macintosh) appears as shown in Figure 15-2. Each of its buttons is described below.

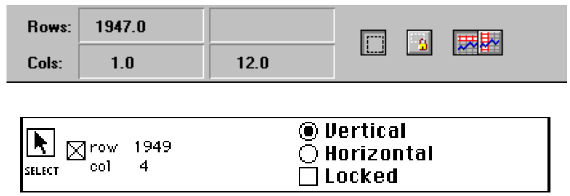


Figure 15-2: Line Graph Options (Windows and Macintosh)

You can select regions of data values in line graph windows. As you select different rows or columns in the data window or select points on an image, the line is automatically replotted to reflect the new selection. However, you can select regions in only one dimension at a time, horizontally or vertically.

Single Value Selected

If a single data point is selected, the horizontal line graph plots the data values from left to right along the row that contains the selected data value. On this line, a small selection rectangle appears on the graph to mark the relative position of the column that contains the selected point.

If a column plot is requested, the plot is drawn from left to right representing the data values from the top to the bottom of the column that contains the selected value. The selection rectangle in the line graph window shows the position of the row for the selected value.

Region Selected

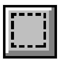
When the selection consists of more than one value and a row plot is requested, the top row of the selection region is plotted. When a column plot is requested, the left column of the selection region is plotted.

Clicking in a line graph window selects a region of data values, just as it does in other types of windows. You can click and drag a selection across the graph to include multiple values in the selection region.

Note

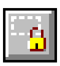
For a row plot, the left and right columns of the selection region are specified by the user but the top and bottom settings are fixed. For a column plot, the top and bottom rows of the selection are specified by the user and the sides of the region are fixed.

Show/Hide Selection Region



Show/Hide Selection button (or checkbox at the left of the bottom margin in Power Macintosh) turns the visible selection box on and off as it does for all image types.

Lock Selection Region



Locked button/checkbox keeps the window from updating when you do not want it replotted. Normally, a line graph window is updated and replotted every time a new selection is made in the dataset.

To compare more than one row plot or column plot on the same screen, create the first line graph and lock it. Then create the second line graph and pick a new row in the dataset to change the line graph to the new profile. Figure 15-3 shows a comparison of two rows from the same dataset.

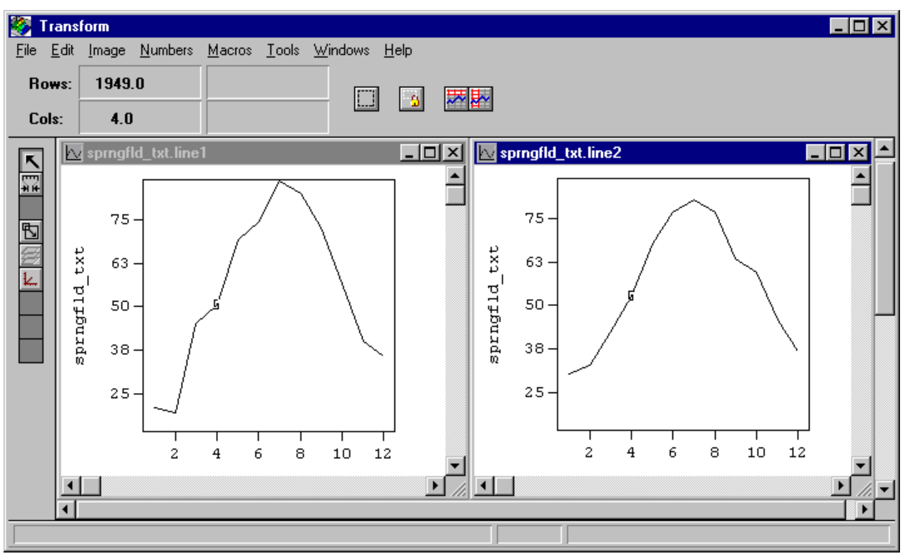
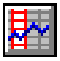
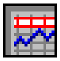


Figure 15-3: Comparison of Two Rows

Column and Row Plots





To plot a column of values (a column plot), click on the **Vertical** button in the toolbar. Click on the **Horizontal** button to plot a row of values (a row plot). Figure 15-4 shows two line graph windows from the same dataset. One plots the values from a row of data and the other plots the values from a column of the same dataset.

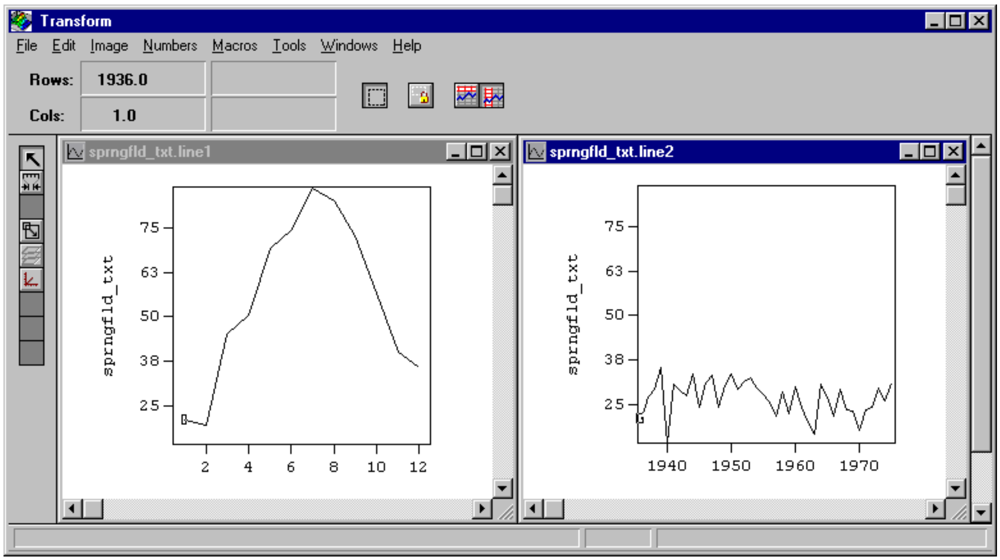
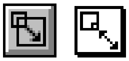


Figure 15-4: Row and Column Line Graphs

Line Graph Animation

You can move the selection region within the dataset using the arrow keys on the keyboard. Line graphs continuously update and replot the data values along the left (or top) edge of the selection region. Therefore, every time you press an arrow key the plot changes one step. Repeated key presses create an animation effect of a moving line traveling through the array.

Resize Tool



Line graphs can be easily sized with the Resize tool. Simply grab one of the corners, depress the left mouse button, and drag the image. In the toolbar, you can resize the line graph by typing new values in the Image Size text fields. Because the vertical scale is associated with the data range, unlike any of the other plot types, dragging the image may be more useful than the **Image Size.../Rectangle Size..**. command, which relates the size of the image to the scales in the dataset.

Chapter 16:

Contour Plots

Many kinds of data are best displayed with a contour plot. The most familiar kind of contour plot is a topographical, such as those sold by the U.S. Geological Survey. The rate of change of elevation is represented by the spacing between contour lines. The closer together the lines are, the more elevation is changing.

Transform draws a contour line marking the path you would travel through the dataset if you wanted to stay at the same data value. A set of these contour lines makes up a contour plot. This section explains how to control which data values are contoured, how to smooth the lines, and how to use color, dashed lines, and labels to define the information.

The dataset can be considered a grid of data points connected by grid lines. Transform uses an interpolation algorithm to determine the points where the contour line passes between the data values and, hence, where it crosses the grid line. Contour lines will always complete a closed curve, unless the curve runs off the edge of the data grid. Although they can bump into each other, they cannot cross.

Create a Contour Plot

To create a contour plot from the current dataset, select **Contour Plot...** from the Image menu. In Windows, the contour plot will be generated in an image window as shown below. In Power Macintosh, the Contours dialog will first appear (described later in this chapter).

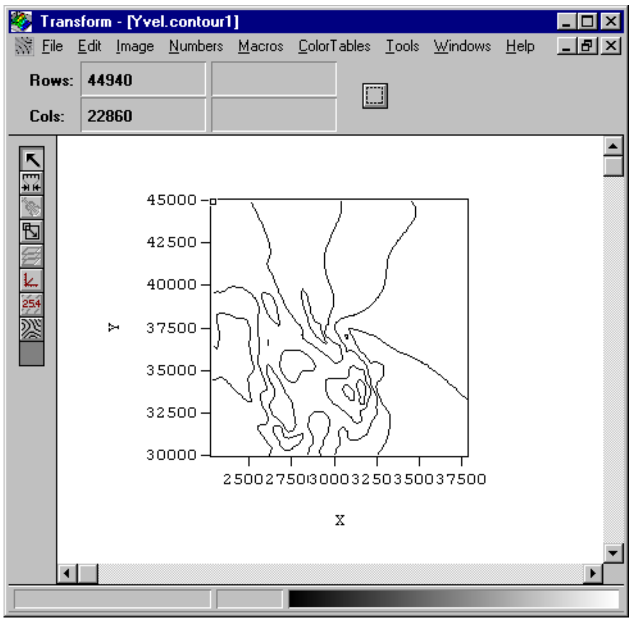
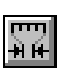


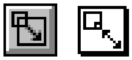
Figure 16-1: Contour Plot

MinMax Tool

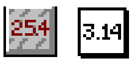
The value range displayed by the MinMax tool controls which contour lines are drawn in the final plot. All contour lines that are specified in the dialog are followed without regard to the MinMax settings, but only those within the range are drawn. To update the plot, change the values in the image Min and Max boxes in the toolbar and press Enter or **Replot**.

Resize Tool



The size of the contour plot can be changed without requiring the lines to be followed again. Change the size of the plot with the mouse or with **Image Size...**. Each time the lines are redrawn, they are drawn in the highest resolution at the full size of the plot.

Label Tool



Labels are interactively added to the contour lines by activating the Label tool then clicking the mouse on any contour you want labeled.

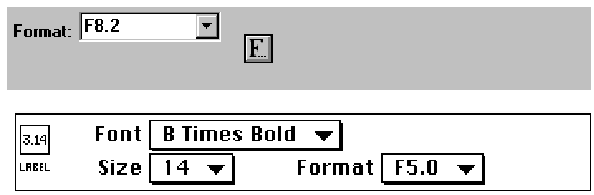
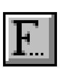


Figure 16-2: Contour Label Options (Windows and Macintosh)

As shown in Figure 16-3, a text label appears at the nearest point on the contour. To remove a label, simply click on it again. Each contour label shows the data value associated with a particular contour line. To select a different number display format for the labels, select a new format label from the Format pop-up menu. Once chosen, press Enter.



You can also change the font, style, and size of labels by clicking the **Font** button/pop-up menu to open the Font dialog.

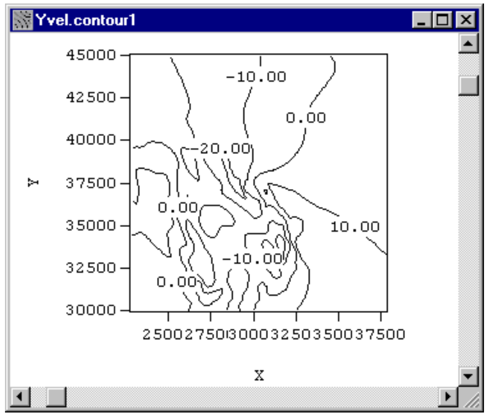
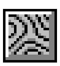


Figure 16-3: Contour Plot with Labels

Contour Tool (Windows)



For contour plots only, the tool palette includes the Contour tool, which provides most of the control over how contours are displayed. With the Contour tool, you can add contour levels interactively and directly on the contour window. Move the arrow cursor to point to the position where you wish to draw a contour. Now click the mouse button. As soon as the new contour has been followed through the dataset, it is drawn according to the active drawing options. Although contours can be added interactively, they can only be deleted in the Contour Levels dialog. Figure 16-4 shows the Windows toolbar when the Contour tool is active.

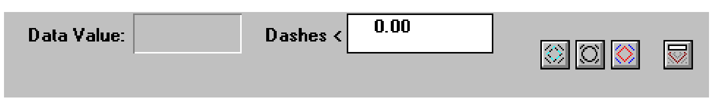
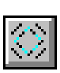


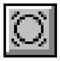
Figure 16-4: Contour Tool Toolbar

Dashed Lines



When you turn on the **Dashed Lines** button, contour lines for data values greater than or equal to the split value are solid, and lines for values less than the split value are drawn as dashed lines. The default for the split value is 0, but you can change it in the edit box provided.

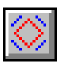
Smoothed Lines



Turning on the **Smoothed Lines** button enables a Bezier smoothing algorithm which adds more line segments that smooth out the sharp angles in the plot. Figure 16-5 shows how smoothed lines add more points to the line drawing to eliminate corners.

Figure 16-5: Lines Without Smoothing and With Smoothing

Color Lines



The **Color Lines** button allows you to choose whether the lines are drawn in black or according to the currently selected color table. If you select color, the colors for each line are chosen in the same way as for raster images. A contour line drawn at a value of 25 is the same color as a data value of 25.

Contour Levels

Click the last button in the Windows toolbar to invoke the Contour Levels dialog.

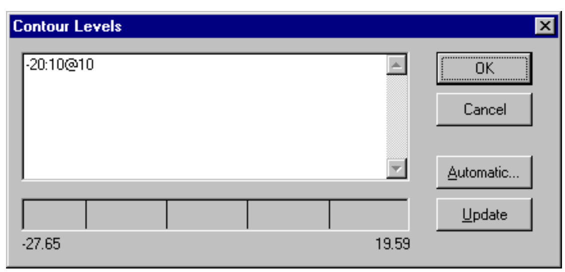


Figure 16-6: Contour Dialog

The main text box lists a contour specification like -20:10@10, which means lines are drawn from -20 to 10 in increments of 10. To change or add to the specifications, simply click in the text box and type other values or ranges separated by commas. Across the bottom of the dialog is a line display of the specifications. Along the right of the dialog box are four buttons: **OK, Cancel, Automatic...**, and **Update**.

The **Automatic**... button invokes a dialog where you can set the number of contour levels you want displayed. Changing the automatic number of contour levels deletes all the current contour levels specified in the dialog and finds evenly spaced contour levels according to the number of levels entered.

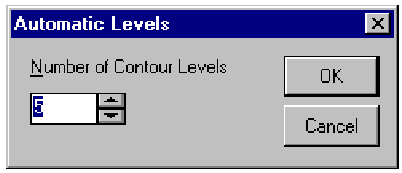


Figure 16-7: Automatic Levels Dialog

Clicking the **Update** button tells Transform to read all of the levels specified in the text box and display them on the line display below.

When you are finished with the dialog, click **OK** to use the new list; click **Cancel** to leave the levels where they were when you started. If you picked new levels, Transform proceeds to find all the levels in the text list and to follow the contour levels in the dataset. After the lines are found, they are drawn in the contour window.

Contour Tool (Power Macintosh)

With the Contour tool, you can add contour levels interactively and directly on the contour window. Move the arrow cursor to point to the position where you wish to draw a contour. Now click the mouse button. As soon as the new contour has been followed through the dataset, it is drawn according to the active drawing options. Although contours can be added interactively, they can only be deleted in the Contour Levels dialog. Figure 16-8 shows the bottom margin of the image window when the Contour tool is active.

Figure 16-8: Contour Tool Options (Macintosh)

The Contour tool on Power Macintosh provides access to the Contours dialog which provides the same options described above for the Windows Contour tool.

Click the **Contours...** button in the bottom margin of the image window (or select **Contours...** from the Image menu) to open the Contours dialog shown below.

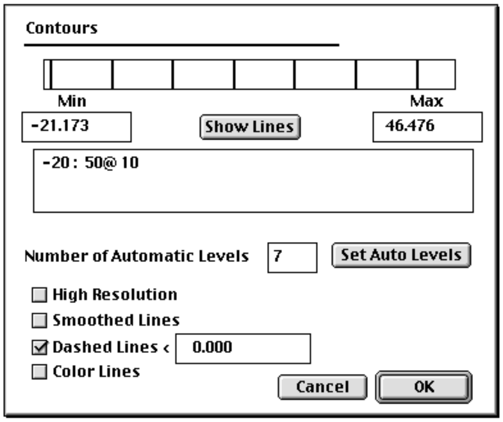


Figure 16-9: Contours Dialog

Note

Adding more than 25 contour lines interactively will cause Transform to quit. You may add as many lines as you wish using the Contour Levels dialog.

Line Display

The line display rectangle contains vertical lines representing the currently chosen contour levels. To add a level with the mouse, click in the line display at the point where you wish to add the contour. If you hold the mouse down, you can drag the line until it reaches the desired location. As you drag the line, the level that you are currently pointing to will appear under the line display.

When you release the mouse button, the line becomes fixed and the contour level is added to the list in the main text box. The data value is rounded according to the data format in the data window. For instance, if the main display format is F6.2 and you click on data value 39.4325, then the value is truncated to 39.43; this value is used to specify the contour.

Min and Max for Contours

The Min and Max boxes contain the minimum and maximum data values in the active range of the dataset. These are the same numbers found in the MinMax tool and the Color Bar window. If a contour level is outside this range but still specified in the contour list, it will be followed but not drawn when the lines are drawn on the plot.

Show Lines

To see where the currently specified contour levels are, select **Show Lines** to update the line display rectangle. A vertical line appears in the line display rectangle for every contour line specified in the main text box. The left side of the line display rectangle represents the minimum active data value, the right side represents the maximum active data value, and the lines are drawn on a linear scale in between. If a contour line falls outside of this range, it is not displayed.

Main Text Box

The main text box contains the full specification of which contour levels are to be drawn. Each level is listed by its data value. You can also specify ranges, with lines drawn at specified intervals. For example, an entry of -20:10@10 means lines are drawn from -20 to 10 in increments of 10. To add or delete contours, edit the text in this text box.

Add single contours by adding that level to the list. Specify as many decimal digits as you need (about 8 digits of accuracy). Each level must be separated by a comma or a space. Ranges of lines that are evenly spaced can be entered in the format A : B @ C where A is the beginning of the range, B is the end of the range, and C is the spacing. To see the new lines in the line display, click on the **Show Lines** button. There is a 256 character limit on the length of the string in the text rectangle.

Set Auto Levels

The **Set Auto Levels** button removes all the current contour levels and finds evenly spaced contour levels according to the number entered in the Number of Automatic Levels box. Enter a new value of automatically spaced levels in the box and click Set Auto Levels. The new levels are added to the main text box and also drawn in the line display rectangle as if Show Lines were clicked. All former lines and text are deleted.

High Resolution

High Resolution means that the contouring of the data grid is made more precise by interpolating a new point in the middle of each cell in the grid before the contour is followed through the new, finer grid. Turn this checkbox on to effectively double the data resolution. Contours are recalculated to make the high resolution take effect.

Smoothed Lines

Turning on the **Smoothed Lines** checkbox enables a Bezier smoothing algorithm which adds more line segments that smooth out the sharp angles in the plot. Figure 16-10 shows how smoothed lines add more points to the line to eliminate corners.

Figure 16-10: Lines Without Smoothing and With Smoothing

Dashed Lines

When you turn on the **Dashed Lines** checkbox, some of the contour lines are drawn as solid lines and some are drawn as dashed lines. Lines for data values greater than or equal to the split value are solid, and lines for values less than the split value are drawn as dashed lines. The default for the split value is zero, but you can change it in the edit box provided.

Color Lines

The **Color Lines** checkbox allows you to choose whether the lines are drawn in black or according to the currently selected color table. If you select color, the colors for each line are chosen in the same way as for raster images. A contour line drawn at a value of 25 is the same color as a data value of 25.

OK

When clicked, Transform finds all the levels in the text list and follows the contours levels in the dataset. After the lines are found, they are drawn in the contour window. If you wish to stop the process, press command-period.

Copying Contour Plots

Choose **Copy** from the Edit menu to copy the contour plot and **Paste/Paste Overlay** to overlay it onto another image. In Power Macintosh, if you copy a contour plot with the Option key selected, your overlay will have white lines instead of black. It is recommended that you turn off axes before you copy a contour window.

Note

Each data value in a raster image generates a rectangle of color centered on the data value; therefore, a raster image extends slightly above, below, to the left and to the right of the location of the bordering data values. Contours, in contrast, are drawn only to the data values, not beyond them. Therefore, when overlaying a contour on top of a raster, there may be a small gap between the edges of the contours and the border of the raster image.

Chapter 17:

Vector Plots

Vector plots are often used for flow diagrams, to show the speed and direction of the flow. A wind speed diagram, for example, can show the relative strength of the wind and which direction it points.

The arrow at the bottom of the vector image is a scale arrow that is equal to the vector maximum. You cannot change it or remove it from the vector plot.

Two arrays of data are required to create a vector plot because the relationship between two orthogonal data arrays is used to determine which direction each arrow points. This chapter provides details on how Transform creates and uses vector plots.

Drawing Vector Plots

A vector plot takes two conforming arrays of data (same size and shape) and creates an image filled with rows of arrows, evenly spaced. The arrows are sized and rotated according to the data values. One value is taken from the first array representing the horizontal component, and another value is taken from the second array representing the vertical component. From these components, the size and direction of the arrow are determined and the arrow is drawn.

Figure 17-1 shows how an arrow's size is determined before being drawn. The full area of the image is divided into equal-sized squares and an arrow is drawn in each square as shown. The center of the arrow always matches the center of the square. The length of the arrow is scaled proportionally to the magnitude of the vector formed from the horizontal and vertical components. The components also are used to form the tangent of the rotation angle for the arrow.

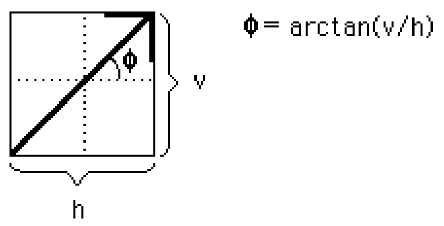


Figure 17-1: Arrow Components

Determine Origin

Transform assigns the positive direction for the arrows according to whether the scale labels ascend in value or descend in value. Figure 17-2, Part A shows the default orientation of the axes and the positive direction for the components of the vector arrows. The arrow shown has positive horizontal and vertical components of equal magnitude. Part B shows that if the row scales are descending values, a positive arrow points up instead of down. Part C shows the arrow pointing up and to the left when both scales have descending values.

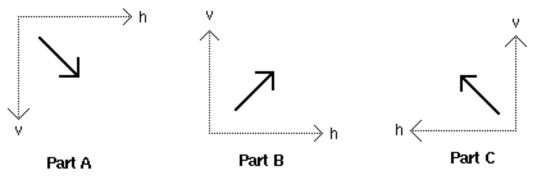


Figure 17-2: Arrow Direction Depends on Axes

Create a Vector Plot

Creating a vector plot starts with the **Vector Plot** command from the Image menu. A standard image window appears with a vector plot created from the current dataset. In this plot, all the arrows point at the same 45-degree angle because when vector plots are created, the same dataset is used for both the horizontal and vertical components of the arrows; therefore the tangent is always 45 degrees.

Choosing Components

When creating a vector plot, at least two datasets with the same dimensions must be open. One dataset is used for the Horizontal component (marked ‘H’ in Power Macintosh) and the other is used for the Vertical (‘V’).

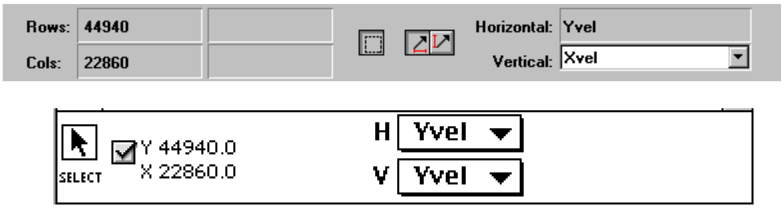


Figure 17-3: Vector Select Toolbar

The Windows toolbar and Macintosh image window have two primary component buttons/checkbox, one for horizontal and one for vertical. They set the component direction for the active dataset. The other component should be chosen from one of the pull-down menus. To choose a dataset, select the new component from the appropriate menu. The menu list only shows the names of open datasets with dimensions identical to the current dataset. As soon as you choose the new component, Transform redraws the vector plot with the new data.

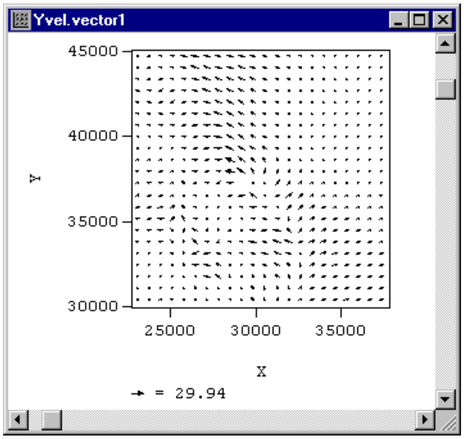
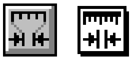


Figure 17-4: Vector Plot

MinMax Tool

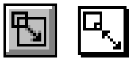


When you are working with vectors, the MinMax tool applies to the vector magnitude, not the value of one of the vector components. For each new vector plot or change of component datasets, the minimum and maximum values are calculated by surveying the magnitudes of the vectors. To eliminate vectors that are too big or too small, change the maximum or the minimum value. Any vectors with a magnitude outside the min/max range will not be drawn.

Figure 17-5: Vector MinMax toolbar

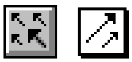
The **Reset** button in the MinMax tool surveys the chosen components, calculating the smallest and largest magnitudes. These results are placed into the minimum and maximum boxes and are used to redraw the plot.

Resize Tool



The number of arrows stays constant as the image size changes, so the size of the arrows stays proportional to the size of the image.

Vector Tool



Choose the Vector tool from the tool palette to bring up the vector-specific controls in the toolbar. Shown in Figure 17-6, the Vector toolbar contains text boxes for changing the number of arrows to plot in the image, a scaling factor for the arrows, and buttons (Windows) to select from six arrow types. Power Macintosh also contains a **Replot** button to make those values take effect, and the **Arrow Type...** button to open the Arrow Type dialog (described below).

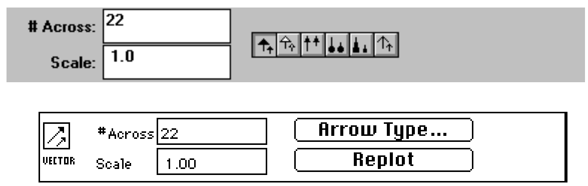


Figure 17-6: Vector Toolbar

Across

To change the density of arrows plotted (more or fewer), enter a new value for the number of arrows to space evenly across the plot and hit Enter. Each arrow field is square, so the number of arrows from top to bottom is determined from the aspect ratio of the image. In Power Macintosh, click the **Replot** button to draw the new graph.

The arrow density need not match the data resolution exactly. The arrows are drawn according to the closest data point to the center of the arrow. If there are fewer arrows than data points, some data points are skipped; if there are more arrows than data points the data points are used more than once. The row and column scales are mapped to the image in the same manner as for raster images.

Scale

Enter a new scale factor to change the multiplier for the size of the arrow, where 1.0 is a full-size arrow of full magnitude. Values greater than 1.0 can create overlapping arrows; values under 1.0 produce arrows that can never overlap or touch each other. Enter a smaller number to reduce the size of the arrows, or a larger number to make the arrows larger, even making them overlap. In Windows hit Enter to redraw the arrows or in Power Macintosh, click the **Replot** button to make the new value take effect.

Arrow Type

The appearance of each arrow is controllable with the arrow type buttons in the Windows tool bar or via the Power Macintosh Arrow Type dialog (available by clicking the **Arrow Type...** button. Each button represents a different kind of arrow display. Two sizes, a large arrow and a small arrow, are shown to indicate how the particular arrow is scaled according to magnitude. Some have the same size arrowhead independent of the scaled length of the arrow, while others have a scaled arrowhead that shrinks as the length of the arrow gets smaller. One type is only an arrow tail with a dot instead of an arrowhead. Pick any of the six arrow types by clicking the button which displays that arrow. In Power Macintosh, click the **OK** button to replot the image with the new arrows.

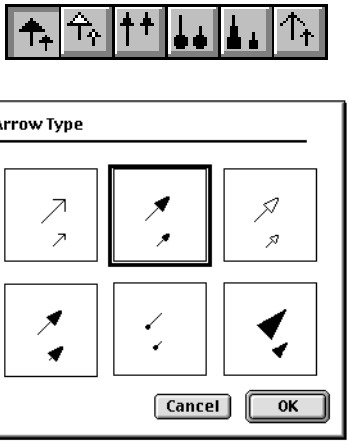


Figure 17-7: Arrow Types (Windows and Power Macintosh)

Chapter 18:

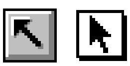
Surface Plots

Surface plots, sometimes referred to as three-dimensional plots because of their three-dimensional perspective appearance, are available in Transform as another way to view two-dimensional arrays of numbers. Two-dimensional data is represented by a three-dimensional flexible surface. The height of each point on the surface is determined from the data values in the array. Arbitrary orientation and size are supported. The surface can be filled with color or drawn as a wireframe, with or without hidden lines removed.

Create a Surface Plot

To create a surface plot, select **Surface Plot** from the Image menu. Initially, Transform draws a surface as a wireframe plot of approximately 10 lines per axis. The term ‘wireframe’ here means that all lines are visible, including ones that would be hidden by other parts of the surface if the surface were opaque.

Select Tool



When a surface plot is active, the Select tool lets you change the surface plot orientation, choose from six types of surfaces, and specify plot parameters.

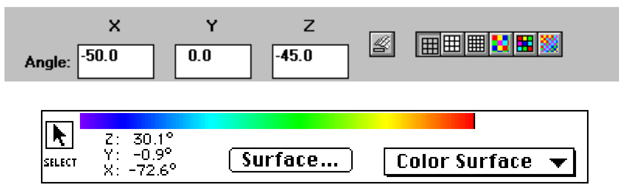


Figure 18-1: Toolbar for Select Tool

Surface Plot Orientation

Using the mouse, you can interactively change the orientation of the surface plot relative to you. When you depress the mouse button, the surface plot image is temporarily replaced by a wireframe box of equal size. To produce a new orientation, drag the mouse across the screen with the mouse button depressed. As you move the mouse, the wireframe box moves and the axes angles are displayed in the toolbar. After you release the mouse button, the new surface plot is drawn in the same orientation as the wireframe box.

In Power Macintosh, when orienting a surface plot, you can move the surface plot within the image area. To do this, hold down the Command key and click and drag the mouse to leave the plot in the new position.

Virtual Trackball

To get an idea of how the plot is rotated, imagine the three-dimensional surface plot encased in a large sphere, just large enough to hold the whole wireframe box. You click the mouse pointer on this sphere as you would touch a trackball and drag that point to a new rotation. This method of mouse-based rotation is known as a ‘virtual trackball'. If you drag from the bottom to the top, the plot rotates away from you. If you drag from left to right, the plot rotates counterclockwise in a horizontal plane. To spin the plot around an axes pointing out of the screen, drag the mouse around the perimeter of the virtual trackball.

Move Anchor

Sometimes you run out of room to complete a rotation with one mouse movement. Press and hold the Shift (Option on Macintosh) key to move the mouse pointer to a new anchor point. Release the Shift/Option key and continue rotating the plot.

Exact Values for Rotation Angles

You may also type exact values for the X, Y, and Z axes angles in the three text fields on the Windows toolbar or in the Power Macintosh Surface Plot Angles dialog (available by clicking the **Surface...** button in selection mode, then clicking the **Edit Angles** button).

The numbers shown in the text fields are the same as the angles of rotation for the surface plot coordinate system. To replot the image and return to the image window, type in your desired angles for the new orientation and press Enter on your keyboard (Windows) or **OK** in the Surface Plot Angles dialog (Power Macintosh).

Location of the Origin

The location of the origin depends on whether these values are positive or negative. When both independent axis scales are ascending, a right-handed dataset coordinate system is drawn, with the origin placed at the top left corner of the matrix. When both axis scales are descending, the origin is drawn at the bottom right corner, resulting again in a right-handed system.

If one or both of the scale values are descending, the origin of the coordinate system is drawn at a different corner. In the case of descending row axis values but ascending column axis values, the origin is drawn at the bottom left corner. This results in a left-handed system. If the column axis scale values are descending but row axis scale values are ascending, the origin is drawn at the top right corner, also resulting in a left-handed system.

Surface Parameters

As well as letting you change the surface plot orientation and choose from a variety of surfaces, the Select tool allows you to specify plot parameters. Clicking the **Surface...** button invokes the Surface Parameters dialog, where you specify grid, parallax, and color variables.

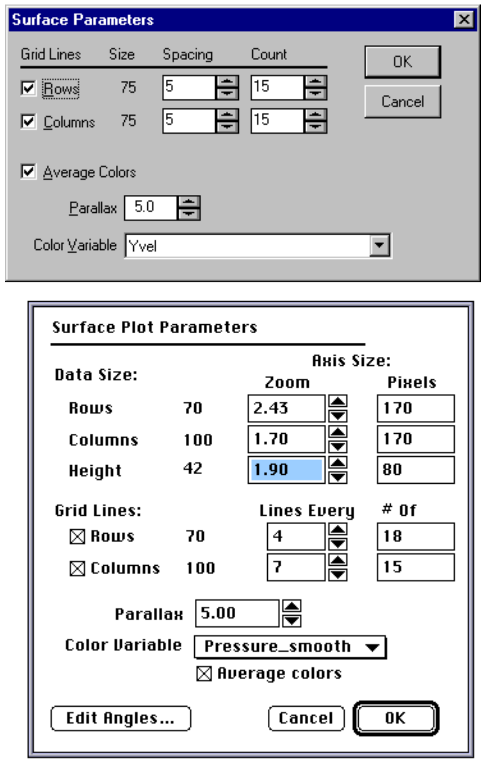


Figure 18-2: Surface Parameters Dialog (Windows and Power Macintosh)

Data Size (Power Macintosh)

To change the plot axis lengths, either select the up or down arrows or enter values in the text edit boxes. These values are specified in pixels, but the process of 3D transformation usually makes the axis lengths in the resulting 2D image come out smaller than the given size. The number of rows and columns in the source dataset are given under the header 'Data Size'. For the plot size, the number next to 'Height' is the z axis, the scaled height of the surface (in pixels) from its lowest point to its highest point. The 'Zoom' factors represent the ratios of the plot axis sizes to the dataset number of rows, number of columns, and default height, respectively.

Grid Lines

You can draw both row and column grid lines, or you can deselect one or the other (but not both) to draw lines in only one direction on the surface.

Note

The density of a surface plot is controlled by the number of lines chosen for BOTH rows and columns, even if you have chosen to display only row or column lines.

The values for Spacing and Count (Windows) or Lines Every and # Of (Power Macintosh) are relative to Size, by default. For example, if Size is 60 and Spacing/Lines Every is 4, Count/# Of will be 15. This defines the total number of lines to be drawn. You can specify how many lines to skip by changing the ratio between Spacing/Lines Every and Count/# Of. The number of lines to be drawn can never be greater than the total number of rows or columns in the dataset.

Parallax

The parallax factor lets you control the virtual distance from your eye to the data surface. The default value is 5.0, which represents a large distance away from the surface. The smaller the number, the closer you are, and the more distorted your view of the surface.

Average Colors

When you check the Average Colors box, each color-filled polygon is defined by the average of the surrounding four data points. When it is not checked, the color for each is set from one corner of the polygon.

Color Variable

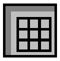
When your request a color surface, you can use the data values of an alternate dataset to supply the color information. Choose another dataset from the Color Variable list. Only eligible (equal-size) datasets are selectable from this menu. This feature provides a means to correlate data values from two different datasets into one graph. One dataset provides the shape of the surface; the other is a color image which is, in effect, draped over the surface.

Surface Types

With the Select tool active, Transform allows you to create six different surface plot types by clicking on the buttons in the toolbar. The types of plots that you can create are, from left to right, Wireframe, Black & White, Hi-res Black & White, Color Surface, Framed Color Surface, and Hi-res Color Surface. When you choose a type, the surface is redrawn in the new type without changing any other display parameters. Some types (e.g., the Hi-res Color Surface plot) take much longer to draw than other types.

All plot types, except Color Surface plot, contain black lines along the rows and columns. In the Surface Parameters dialog (discussed below), you can specify whether you want row lines, column lines, or both.

Wireframe



A wireframe surface drawn in perspective with black lines and no removal of hidden lines (Figure 18-3) is what you see when you initially generate a surface plot. This is the default type for all new surfaces because it is the fastest to draw. Data values are skipped if you draw fewer lines than there are data points.

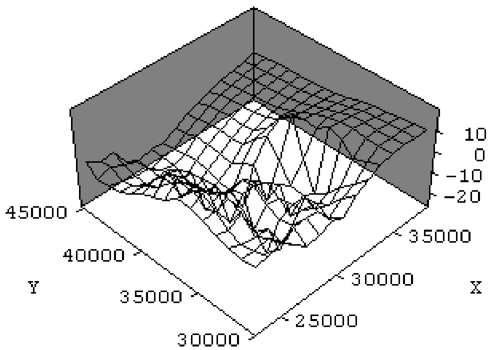


Figure 18-3: Wireframe Surface Plot

Black & White Surface Plot

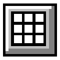


Figure 18-4 shows a black and white surface plot (B&W Surface) similar to the wireframe plot but filled in with white, resulting in the removal of hidden lines. Skipped data values between the lines are ignored.

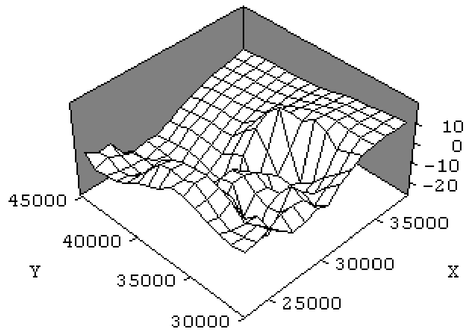


Figure 18-4: Black & White Surface Plot

Hi-res Black & White Surface Plot

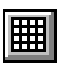


Figure 18-5 shows a high-resolution, black and white surface plot (Hi-Res B&W Surface). The same number of black lines are drawn as in the black and white surface plot. Between the lines, however, every data point is positioned and drawn as a white rectangle.

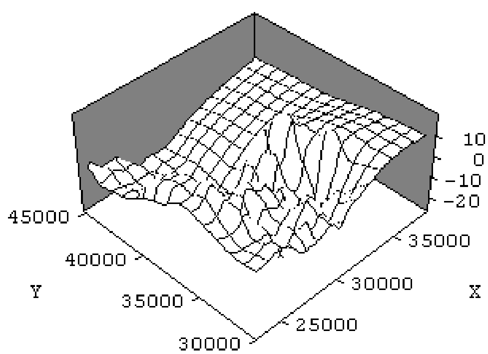


Figure 18-5: Hi-res Black & White Surface Plot

The lines appear to curve because they are drawn in several segments at a finer resolution. For some plots, the data between the lines may obscure portions of those lines. Using the Color Surface Plot or increasing the number of lines can help reduce the confusion this may cause.

When lines are requested along every row and column, no skipped data points fall between the black lines. In such a case, the Hi-res Black & White and Black & White Surface plots are identical.

Color Surface Plot

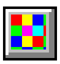


Figure 18-6 shows a color surface plot. Each rectangle patch has a color calculated from the average of the four corner values of the patch. When the image is drawn, the appropriate color from the current color table is used to fill each patch. Additionally, the color of each patch is taken from the same dataset that determines the shape of the surface. These defaults can be overridden with a different parameter setting. Color information can be taken from an alternate dataset.

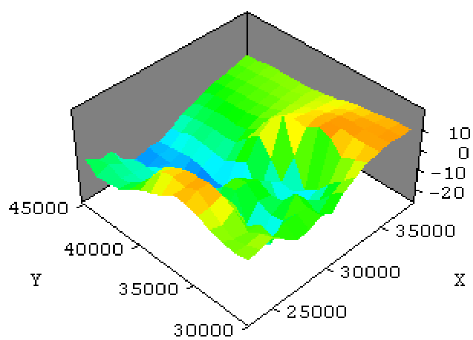


Figure 18-6: Color Surface Plot

Framed Color Surface Plot

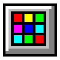


Figure 18-7 shows a framed color surface plot. This is similar to the color surface plot, but each rectangle of the surface is framed with lines. The lines help define the shape of the surface, while the color can be used to display data values from a different dataset or used to reinforce the surface shape.

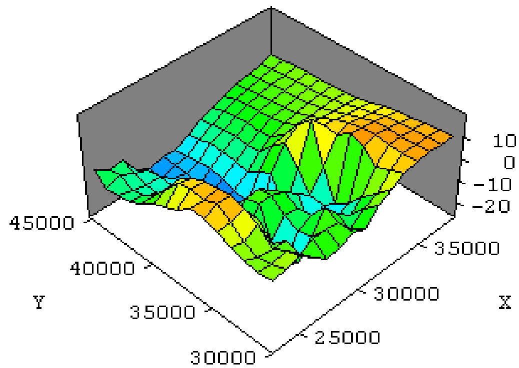


Figure 18-7: Framed Color Surface Plot

Hi-res (Framed) Color Surface Plot

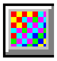


Figure 18-8 shows a high-resolution, framed, color surface plot, which takes a relatively long time to render. Like the high-resolution black and white surface plot, this plot has black lines. Black lines are drawn along only the lines requested, and they appear curved because they are drawn using every data point along the line. When lines are requested along every row and column, no skipped data points fall between the black lines. In such a case, the Hi-res Color Plot is identical to the Framed Color Surface Plot.

.

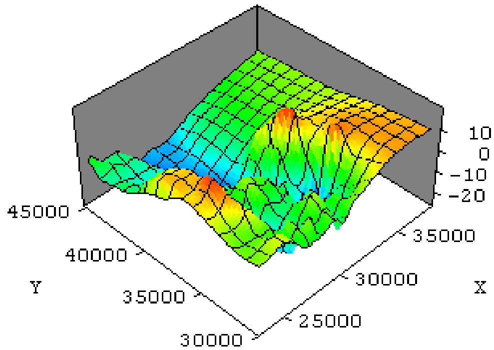
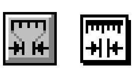


Figure 18-8: Hi-res Color Surface Plot

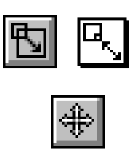
The rows and columns of data that do not fall on the black lines are represented by color. Inside each black frame, Transform draws a high-resolution color map of pixels with every data value represented. Each patch of color represents the average of the four adjacent data values (by default). Sometimes a central patch of color may obscure a segment from the framing lines.

MinMax Tool



In surface plots, the minimum and maximum data values are mapped to the lowest and highest points on the surface. Limiting the range of values is a common tool for analysis of surface plots. The effect is to 'chop off' the mountains and valleys so that detail in the middle range can be displayed.

Resize Tool



The Windows surface Resize tool allows you to translate (or move) the image within the plot window and to size images. To translate the image, simply click on the **Move Surface** radio button in the toolbar, then with the cursor back over the surface plot, depress the left mouse button, and move the image.

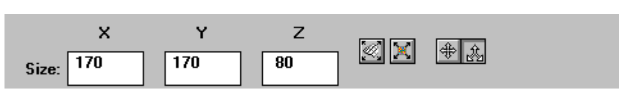
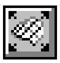


Figure 18-9: Surface Resize Toolbar

In Windows, you can also resize your active plot directly by typing values for the X, Y, and Z coordinates into the rectangles in the toolbar or use the Surface Scaling dialog described below.

In Power Macintosh, the initial plot rectangle is the same as for the other plots, controlled by the **Rectangle Size...** command, but to make it fit within the window, the size of the surface starts out smaller than the window.

Surface Scaling (Windows only)



Another option for sizing a surface plot is to click on the **Surface Scaling** button to bring up the Surface Scaling dialog. You can enter either zoom values or actual values (in pixels) for Rows, Columns, and Heights in this dialog.

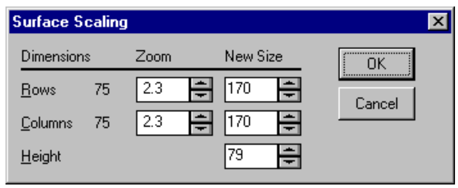
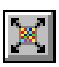


Figure 18-10: Surface Scaling Dialog

Fit to Window (Windows only)



Fit to Window makes Transform size the surface plot to fit reasonably within the image window. Transform calculates an average fit, so you may still need to fine tune the size afterward.

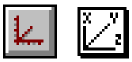
Move Surface (Windows only)

The **Move Surface** button lets you move the surface plot around on the canvas within the image window. When Move Surface is active, you can move the plot by clicking on the it, holding the mouse button down, and dragging the plot to the desired location.

Interactive Sizing (Windows only)

To interactively scale the plot in all three coordinates, click on the **Resize Surface** button in the toolbar and click on the plot. When you click, the plot is replaced with the outline box until you release the mouse. Moving the mouse up and down enlarges and reduces the Z (data) axes. Moving the mouse right and left reduces and enlarges the plot along the X and Y axes, respectively. As you resize the image, the current plot size is continuously updated in the toolbar. To scale the plot in only the X and Y axes, depress the Control (CTRL) key and maneuver the mouse as before.

Axis Tool



Clicking on the Axis button in Windows brings up the array of toolbar buttons that you've seen with other images. From left to right, the buttons activate the axes labels dialog for rows, columns, and data; the color bar; the color bar axes label dialog; the image title dialog; and the surface decorations dialog. The last five radio buttons specify the plot's axes origin. In Power Macintosh, the Axis tool provides interactive sizing and access to labeling. In the bottom margin of the image window, the left side of the margin displays the current plot size in pixel unit. These are the same axis size values found in the Surface Plot Parameters dialog, but the Surface Size tools allows you to interactively resize with the mouse.

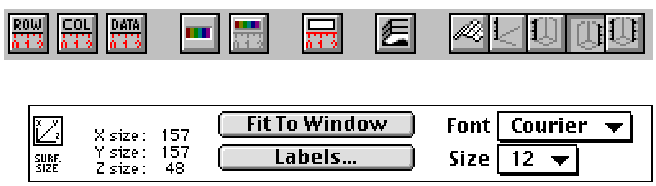
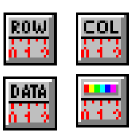


Figure 18-11: Surface Axes Toolbar

Row, Column, Data Axes, and Color bar Labels



In Windows, these four buttons invoke an Axes label dialog, in which you specify axes name, plot range, axes labels range, increments and intervals, tick mark settings, auto range calculation, and label format.

In Power Macintosh, clicking the **Labels...** button opens the Surface Labels dialog, which lets you control axis drawing, labels, backstops, skirts and tickmarks.

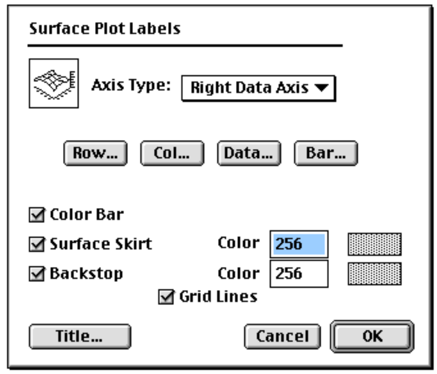
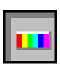


Figure 18-12: Surface Plot Labels

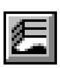
Enable Color Bar



In Windows, click this button to turn the color bar display on and off for a surface plot.

In Power Macintosh, this same option is available by selecting or deselecting the Color Bar checkbox in the Surface Plot Labels dialog.

Surface Skirt, Backstop and Grid Lines



In Windows, clicking the **Surface Decorations** button invokes a dialog with three checkboxes: skirt, backstop, and gridlines. If the box that corresponds with these features is checked, the feature is enabled. Click in the box to change the current setting. You can also type a color index for skirt and backstop.

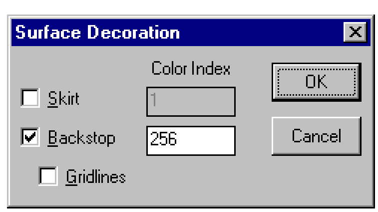


Figure 18-13: Surface Decorations Dialog

In Power Macintosh, these same features are available in the Surface Plot Labels dialog.

Skirt—The skirt is the base of the flexible surface; drawing one will obscure any view you may have had of the bottom of the surface. Skirts can be drawn in color. The color index box is active only when the skirt is enabled, allowing you to type in a color number to be used. This color number is on the same scale as the one produced by the color scaling equation used for raster images. Enter a number from 1 - 254 for a color, 0 for white, 255 for black, or 256 for gray.

Backstop—The backstop is made up of the two rear walls of the imaginary cube enclosing the surface plot. The backstop enhances the three-dimensional effect of the plot. Backstops can be drawn in color. The color index box is active only when the backstop is enabled, allowing you to type in a color number to be used for drawing. This color number is on the same scale as the one produced by the color scaling equation used for raster images. Enter a number from 1 - 254 for a color, 0 for white, 255 for black, or 256 for gray.

Gridlines—The backstop can also include horizontal gridlines at the major tick intervals. To turn on gridlines, click in its checkbox. You may have to enable Backstop first.

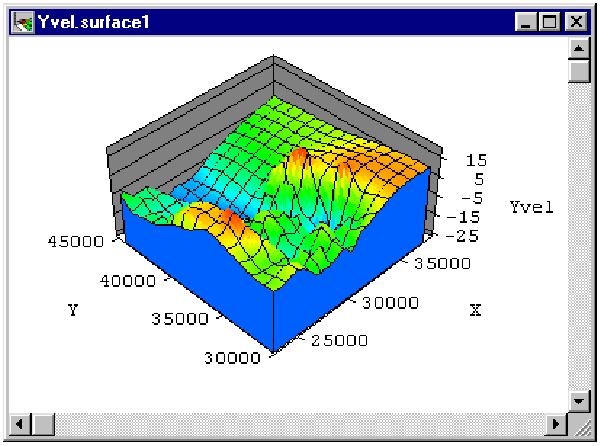
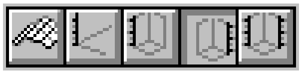


Figure 18-14: Finished Surface Plot with Skirt and Gridlines

Axes Types

The axes type controls the general appearance of the surface plot's axes. In Windows, the last five buttons let you specify from what perspective you wish to view the plot. In Power Macintosh, these same options are available from the Axis Type pop-up menu in the Surface Plot Labels dialog.



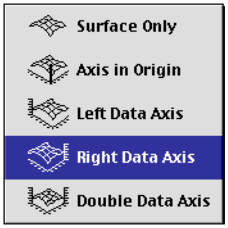


Figure 18-15: Axes Type Buttons (Windows Toolbar) and Macintosh Axis Type Pop-up Menu in the Surface Plot Labels Dialog

The available types are surface only (no axes), axes in origin (axes drawn to the origin point), left data axes (data axes drawn to the left of the plot), right data axes (data axes drawn to the right of the plot), and double axes (a combination of left and right).

Interactive Sizing (Power Macintosh Only)

Click the mouse anywhere within the window and drag to resize the plot. When you click, the plot is replaced with the outline box until you release the mouse. Moving the mouse up and down enlarges and reduces the Z (data) axis. Moving the mouse right and left reduces and enlarges, respectively, the plot along the X and Y axes simultaneously. The current plot size is continuously updated at the bottom of the window as you resize the plot.

Note

Interactively increasing the surface plot using the Axes tool does not automatically increase the image area. You may also need to increase the size of the image area using the Resize tool.

Adjust the X and Y Axes Independently (Power Macintosh Only)

Normally, the X and Y axes are kept at the same aspect ratio. To adjust the X and Y axes independently, hold the option key down. In this mode, the Z axis size is frozen and the mouse movement controls the X and Y axis sizes separately. Up-and-down movements of the mouse control the Y axis; left-to-right control movements control the X axis.

Font and Size Pop-up Menus (Power Macintosh Only)

Two pop-up menus at the right side of the window’s bottom margin control the font and size of the labels and numbers to be printed in the window. To change the plot’s annotations, select a new font from the font menu or a new size from the size menu.

Fit to Window Button (Power Macintosh Only)

Fit To Window makes Transform pick a reasonable size for the surface plot so it fits within that window. Transform calculates an average fit which does not always match the current plot, so you may want to fine-tune the size after using **Fit To Window**.

Chapter 19:

Histograms (Windows Only)

Histogram plots in Transform for Windows provide a graphical view of the distribution of the values in a dataset. The values are presented in the form of a vertical bar graph with one bar for each entry in the color table. The plot is drawn by calculating how many data values are mapped into each color table entry and setting the height of the vertical bars accordingly. In this chapter you will see how to use histogram plots both as an informative analysis plot and to facilitate control over the color scaling for a dataset.

Create a Histogram

With the dataset window active, select the **Histogram** command from the Image menu.

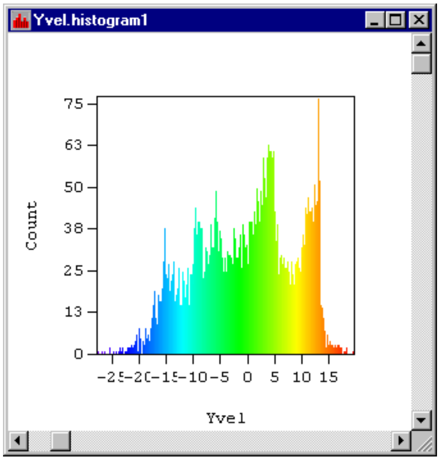


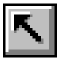
Figure 19-1: Histogram plot

The vertical axis of a histogram plot shows the count of how many data points are present in the dataset which map to each color. The horizontal axis covers the data range from the minimum data value to the maximum data value. Any points which fall outside of the current data range, called outliers, are omitted from the plot.

A histogram displays the same information you see in a color raster image of the same data, arranged differently. The same number of data points appear in the same colors in both types of plots. The raster image shows the data arranged in its original two-dimensional form while the histogram displays the points sorted in order by data value.

The histogram may be drawn in color, with each vertical bar drawn in the corresponding color from the current color table. Without color, each bar is drawn in black.

Select Tool



The Select tool allows you to interactively explore the histogram information. Click on one of the vertical bars and observe the numbers at the left side of the toolbar. The Value box shows an approximate data value (position on the horizontal axis) for that bar. The count box presents the number of data values which were mapped into that color cell (vertical axis).

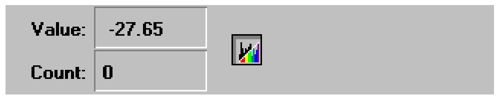
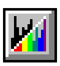


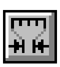
Figure 19-2: Histogram Select Toolbar

Draw Color Histogram



The **Draw Color Histogram** button on the Select toolbar controls the coloring of the vertical bars. If this setting is on, bars are drawn in color. If it is off, the bars are all black.

MinMax Tool



A histogram plot may be used to display and modify the data range, using the MinMax tool. When a suitable range is found, it can be propagated to all of the other image windows.

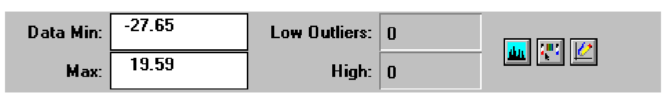


Figure 19-3: Histogram MinMax Toolbar

As with the MinMax tool for any other plot, you may directly edit the data range minimum and maximum using the text fields in the toolbar. You may also change the data range by clicking and dragging the black triangles at the bottom of the plot. As you drag the triangle, the data range values in the toolbar are updated. When you release the mouse, the data range is changed and the histogram is replotted.

To help you analyze the current data range, the number of low and high outliers in the dataset are shown in the toolbar.

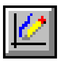
Reset Data MinMax

This button resets the data range for the histogram window to the absolute minimum and maximum of all data values in the array. After clicking this button, there will be no outliers.

Edit Outlier Colors

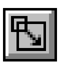
Outliers "fall off the edges" of a histogram plot. Using the Outlier Colors dialog does not change the appearance of the histogram. Since the histogram window is commonly used to set the data range for other image windows, you may still find this feature useful. After setting the data range and the outlier colors you may propagate these settings to the rest of the dataset's windows, as described below.

Update All Windows



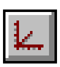
This button propagates the current data range and color scaling settings to the rest of the dataset. All existing plot and image windows are automatically redrawn. The settings become the primary settings for the dataset so that any new windows created inherit the new values.

Resize Tool



Histograms may be interactively sized with the Resize tool. Because the scales on histogram plots are not associated with the dataset row and column scales, this may be more useful than the **Image Size...** command.

Axis Tool



As explained above, the horizontal axis for histograms is associated with the data range. This means that the plot bounds for the horizontal axis is actually the data minimum and maximum. If you modify the plot bounds in the axis dialog for this axis, those changes modify the data range.

Chapter 20:

Window Options and Preferences

Here we discuss selecting data regions in windows, synchronizing windows, tiling windows, hiding windows and preferences.

The Selection Region

A fundamental concept in Transform is the selection region. Transform allows you to mark, or select, a region of data cells in a dataset window or on an image plot and have that region be marked in all windows associated with the dataset. With this feature you can point to an area of particular interest in one window, such as a raster image or contour plot, and automatically have the data values for that area visible and selected in the dataset window. If you also have a line plot open, it will display a horizontal or vertical profile of the dataset at that point. An example of this coordination is shown below.

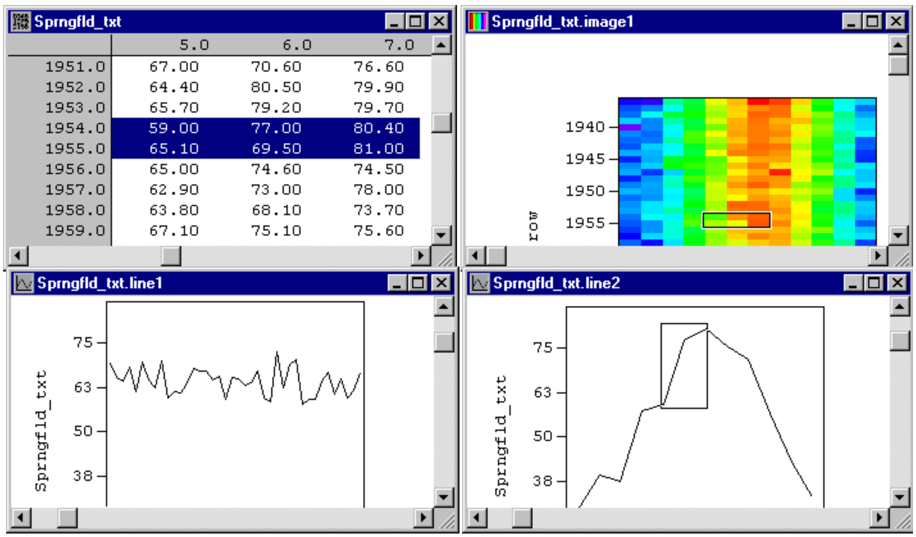


Figure 20-1: Multiple Image Windows Showing Same Selection Region

The Selection Region and Window Types

The selection region can be found on dataset windows, raster images, interpolated images, polar plots (Macintosh only), line plots, contour plots, and vector plots; the selection region is not available for surface plots or histograms (Windows only).

On dataset windows, the selection region is shown in inverted text. On the other window types the selection region is shown as a rectangle surrounding the pixels that correspond to the selected values. Note that if your row or column numerical scales are not uniformly spaced, the size of the selection region for the same number of data points may vary across the image (as it should).

On line plots, Transform not only draws a rectangle around the selection region, but also redraws the entire graph using the row (or column) of data from the top (left) edge of the selection. The top and bottom edges of the rectangle match the range of data values within the selection; the left and right edges match the columns (rows) in the selection region.

In Macintosh polar plots, Transform treats the selection region as a set of radius and angle measurements. An annular sector is drawn in the window, enclosing the corresponding selection region.

The Selection Region and the Tool Bar (Windows only)

The selection region is shown when either the Select tool or the MinMax tool is active. When the Select tool is active, Transform shows the coordinates of the selection region in the tool bar. This is done using the dataset's numerical scales. When the MinMax tool is active, Transform maintains the minimum and maximum of the selection region in the tool bar.

The Selection Region and the Mouse

The mouse may be used to change the selection region when either the Select tool or the MinMax tool is active.

- Click and drag the mouse across the window to start a new selection region.
- Click the mouse while the shift key is depressed to extend or modify the current selection region.
- On dataset windows, you may also click and drag across a series of column scale labels to select an entire column or series of columns and likewise for entire rows. If you click on the area left of and above the scale labels, Transform will set the selection region to the entire dataset.

The Selection Region and the Keyboard

Cursor motion keys may also be used to change the selection region when the Select tool is active and, with the exception of dataset windows, when the MinMax tool is active. The table below provides a list of keyboard functions available with Windows.

Key	Action	Action with CONTROL Key
Up-Arrow Down-Arrow Left-Arrow Right Arrow	Moves one cell in the indicated direction	Moves one screen-worth in the indicated direction
Page Up Page Down	Moves one screen-worth in the indicated direction	Moves to the top or bottom of the dataset
Home End	Moves to the left or right edge of the dataset	Moves to the top-left or bottom-right corner of the dataset

Table 20-1: Keyboard Functions (Windows only)

For each of the above key combinations, the SHIFT key may be used to extend the current selection region to the appropriate destination rather than move it.

Transform for Power Macintosh gives lets you use the Up-, Down-, Left- and Right-Arrow keys as described in the table above, plus the ability to extend a selection using the SHIFT key.

Synchronizing Selections Across Windows

When you change the selection region, Transform updates all windows for that particular dataset automatically. Sometimes, you may want Transform to also update the selection region for other datasets. For example, when you have temperature and humidity measurements for the same area and you want to compare them at corresponding locations.

To synchronize the selection region for multiple datasets, select **Synchronize** from the Edit menu. If it is active, there will be a checkmark next to it. Select it again to turn synchronization off.

With Synchronize on, an update to a selection region affects all datasets with the same number of rows and columns; datasets of a different size are unaffected. If you want to synchronize datasets with different sizes, first use **Resample Data...** from the Numbers menu to resample one dataset to the same size as the other.

The Windows Menu

The Windows menu helps you hide (Power Macintosh only) windows you do not want to see and find the windows you want to see. It can also rearrange the windows you have on the screen.

Window Names in the Windows Menu

The Windows menu represents a complete list of the windows that are available. On the Windows menu, a checkmark marks the current window—the frontmost window on the screen. The dataset associated with that window is the current dataset. Any image generated from the Image menu is always of the current dataset.

Window Naming

Window names are formed from the dataset name and a suffix depending on the window type. The suffix also contains a number to distinguish multiple windows of the same type.

Selecting Windows from the Menu

In addition to clicking in a window to select it, you can also select a window from the Windows menu, which brings the window to the front. This is useful for windows that are hidden or obscured.

Hide Window Command (Power Macintosh only)

The **Hide Window** command from the Windows menu makes the current window invisible: its entry in the Windows menu is italicized to indicate that the window cannot be seen. The only way to make the window visible again is to select the name from the Windows menu.

Tile Windows Command

Transform can arrange windows on the screen so that as many windows as possible may be seen at the same time. Transform will shrink or expand the open windows to make them fit on the screen.

You may want to 'hide' some of your windows before you select the **Tile Windows** command, so they do not clutter up your screen. You can always 'unhide' the hidden windows later.

If you have a lot of windows, Transform may be forced to layer the tiled windows (e.g., a screen of 640 x 480 pixels can hold eight windows). The first eight windows will be tiled normally, then the next eight will be tiled on top of the first eight in a second layer. This will continue with more layers if necessary. You can access the first layer by using the Windows menu or moving the second-layer windows.

Note

Transform does not tile color bar windows and hidden windows.

Preferences

Preferences are a collection of settings that specify the default behavior of Transform. When Transform first starts up, all of the preferences are set to the last settings saved. As you work with the program, Transform uses the preference settings to remember almost everything you enter in dialogs or set up when creating images.

Preference settings are associated with the reserved macro variables described in Chapter 23. For example, the default behavior of Transform is to create contour plots without color. This is because the default setting for the **contour_color** reserved variable is false. If you change the Preferences setting to **contour_color=true**, new contour plots will be created in color.

Preferences for Windows

This section describes how to set preferences in Transform for Windows.

Font Preferences

To set the default font for Transform, select the Preferences submenu under the Edit menu, and choose the **Font...** command. This opens a standard font browser displaying the current default font. Select a new default font and size and click **OK**. The default font is used for all new data and notebook windows. To make this setting permanent, save your preferences with the Preferences Settings dialog, described below.

Display Preferences Dialog

To modify your display preferences, select the Preferences submenu under the Edit menu, and choose the **Display**... command.

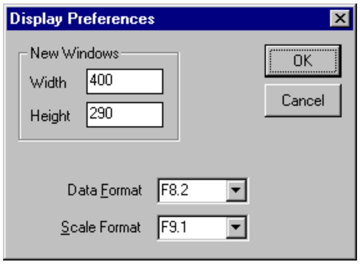


Figure 20-2: Display Preferences Dialog

New Windows: Width and Height

These values determine the size of all newly created windows. You may want adjust these to suit your particular display configuration.

Data Format and Scale Format

Newly created datasets will use these display formats for data and scales. However, in some cases, these defaults may be overridden. For example, in file import, display formats saved with the dataset override the data and scale format defaults.

Preferences Settings Dialog

To view and modify the complete list of preference settings, select the Preferences submenu under the Edit menu, and choose the **Settings...** command. The dialog shown in Figure 20-3 appears.

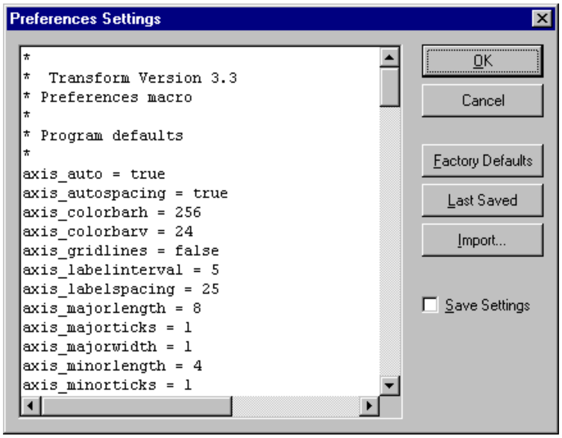


Figure 20-3: Preferences Settings Dialog (Windows)

This dialog consists of a text editing area which lists all the preference settings. When the dialog first appears, the values shown reflect the current state of the preference settings in Transform. Note that these settings may change during your use of the program.

Using the text editing area and the various options in this dialog, you may modify the current settings and, optionally, save those settings. The options in this dialog are discussed below.

Factory Defaults

Click this button to restore your Preferences settings to the default state they had when Transform first installed from the CD. All settings and reserved variables are reset to their initial values.

Last Saved

Click this button to reset all the current settings to the last settings you saved.

Import

Click this button to import the preferences settings and custom macros from previous versions of Transform. Used when upgrading Transform.

Save Settings

If this box is checked at the time you click **OK**, the settings shown in the dialog will be saved on disk, and will be active the next time you use Transform as well.

Preferences for Power Macintosh

To set Transform preferences, select **Preferences...** from the Edit menu. The dialog is shown in Figure 20-4. The options in this dialog are discussed below. Parameters that you save in this dialog remain in effect from session to session.

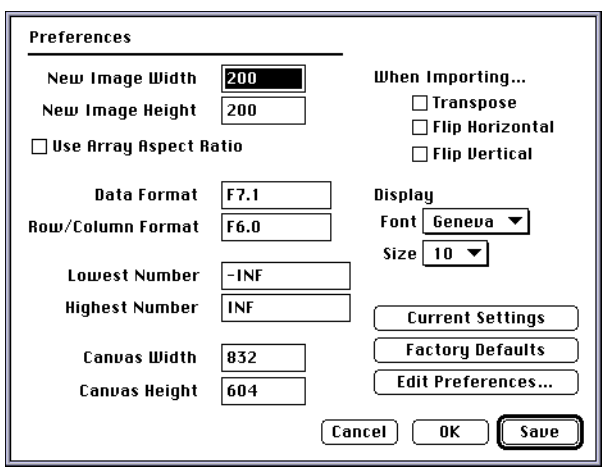


Figure 20-4: Preferences Dialog (Power Macintosh)

New Image Width, New Image Height

Enter the width and height in pixels of new images. Images you generate in this session will be of this size. Changing this does not affect images currently open.

Use Data Aspect Ratio

When checked, the image sizes are changed so that the aspect ratio is maintained. For example, if the current image size is 200 x 200 pixels, an image generated from a 70 x 100 array will be 140 x 200 pixels in size. Note that in this option, the aspect ratio is defined as the ratio of the column size to row size, not the column range to the row range.

Data Format, Row/Column Format

Enter display formats for the data values and column/row labels in the data window. The format specifications are detailed in Chapter 10.

These defaults are used when you create a new dataset, import a non-HDF dataset, or paste data from the clipboard. In other cases, Transform uses the formats stored with the data. For example, if you use the **Extract Selection** and **Calculate From Notes** commands, Transform copies the formats from the datasets used in the calculation.

Lowest Number, Highest Number

All data values outside this low/high range are excluded when Transform does any calculations. This feature affects the color bar window, imported data, and notebook-derived datasets. To reset the largest possible range, set these fields to -INF and INF.

This feature is most useful when working with missing data. For example, if your missing data value is -9999 but your true data goes from around 0 to around 300, you can calculate the true min/max of the data by setting the Lowest Number to -9000 and then opening the Color Bar.

Canvas Width, Canvas Height

Use to set the default background for image windows. The canvas corresponds to the paper your image is printed on; the scroll bars control your view of that canvas. Other controls allow you to resize and move the image anywhere on the canvas.

The canvas always starts out at least the size of your screen and is enlarged when you create any image larger than your screen. When the image prints, the upper left corner of the canvas is mapped to the upper left corner of the paper. If you want more room to scroll or more room to print, increase the size of the canvas.

When Importing..., Transpose, Flip Horizontal, Flip Vertical

When importing non-HDF datasets, you can have Transform automatically transpose your date (swap rows for columns), flip the columns horizontally left for right, or flip the rows vertically top for bottom. Transform does not apply these operations to its own datasets.

Display Font, Display Size

Choose your desired Font and Size selections from the pop-up menus in the dialog. The choices you make affect all data and notebook windows. For most windows, you will see the change as soon as you click **OK** or **Save**. These settings can be overridden for axis labels and for labels in contour plots.

Current Settings

To save your current settings from session to session, click **Current Settings** and then **Save**. Transform remembers almost everything you enter in dialogs or set up when creating images.

Factory Defaults

Click this button to restore your Preferences settings to the default state they had when Transform was installed from the CD. All settings and reserved variables are reset to their initial values.

Edit Preferences...

Click this button to edit the complete list of saved Preferences variables, saved as a Transform Macro. When you are done editing, click on **OK** to return to the Preferences dialog.

Note that the Preferences macro is different from a normal Transform macro in that the list of variables and commands is fixed: you cannot add or delete macro commands or variables, you can only change the values assigned to the variables. Refer to Chapters 22 and 23 for more information on macros and reserved variables.

Cancel

Select **Cancel** to return to Transform without making any Preferences changes.

OK

Click on **OK** to store your Preferences changes. Your changes will only be active for the current session.

Save

Click on **Save** to store your Preferences changes. All changes except for New Image Width and New Image Height will be stored on disk in the Transform application, and will be active in this and all future sessions.

Chapter 21:

Data Exchange and File Export

Transform provides several commands under the Edit menu that allow you to exchange data using the clipboard. In addition, you can export datasets and images using the **Save As** command from the File menu.

Data Exchange Commands

This section describes the variety of copy and paste commands available to both the Windows and Power Macintosh versions of Transform.

Copy Command

The **Copy** command makes choices about what to copy, and about copy format, according to the context in which it is selected.

Using Copy in Dataset and Notebook Windows

When selected from a dataset window, the **Copy** command places whatever is currently selected on the clipboard as text. Data in the selection region is copied as lines of numbers separated by tabs. In Windows, the scales are not included. In Power Macintosh, you may copy an entire row and column with its scales by clicking the individual row or column header, then press and hold the Shift key while clicking the lower triangle above the column of labels. To copy the entire table with scales, click the region of the two triangles at the upper left corner of the data window, then hold the Shift key and click again in the triangle region. This is suitable for pasting into spreadsheet programs. In addition, text on the clipboard in this format may be pasted into another Transform dataset window.

When selected from a notebook window, the **Copy** command places the currently selected text on the clipboard. The text can be pasted into other applications, or into other Transform notebook windows.

Using Copy in Image and Plot Windows

When selected from any plot or image window, the **Copy** command places the plot or image on the Clipboard as a Picture (a Windows Metafile) or as a bitmap (Power Macintosh). This is suitable for pasting into presentation graphics programs. In addition, graphic objects on the clipboard in this format may be pasted onto any Transform plot or image as an overlay.

In Power Macintosh, if an overlay is present and the Select tool active, it is copied along with the image. In this case, the drawing in the Clipboard consists of the base image plus the overlay drawing.

Note

Power Macintosh: if you copy a contour plot with the Option key selected, your overlay will have white lines instead of black.

Copy As... Command (Windows Only)

Copy As... allows more flexibility by bringing up a dialog with options. When selected from a dataset window, the **Copy As...** command produces the dialog shown in Figure 21-1.

This dialog always consists of two lists. The left list displays the choice of objects that may be copied. The right list contains the choices of what formats to use when storing the information on the clipboard.

The contents of the lists vary depending on what type of window is active, and the contents of that window.

Using Copy As... from Dataset Windows

From a dataset window, the selection region is copied as text, just as it would be using the **Copy** command. **Copy As...** provides the option of copying with or without scales, as shown in Figure 21-1.

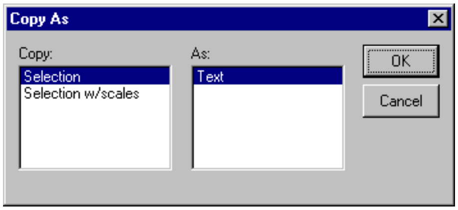


Figure 21-1: Using Copy As... from a Dataset Window

Note

From a notebook window, **Copy As...** works identically to **Copy**; the selection region is simply copied as text.

Using Copy As... from Image and Plot Windows

Figure 21-2 shows the Copy As dialog when invoked from an image window that contains a raster image and an overlay.

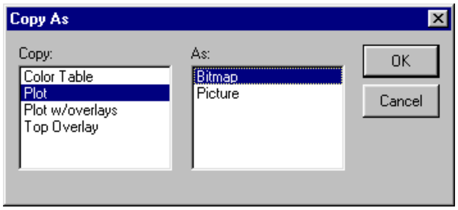


Figure 21-2: Copy As Dialog

From a plot or image window, Transform allows you to copy either the color table, the image or plot, the image with an overlay, or just the overlay. Format options listed on the right depend on the component selected on the left:

- A color table may only be copied as a Palette object.
- The plot itself may be copied as either a Picture (Windows Metafile) or a bitmap.
- The plot may be copied with its overlay(s) as a Bitmap.
- The top overlay may be copied as a Picture.

Copy Color Tables (Power Macintosh Only)

Color tables, technically 'clut' resources, can be copied and pasted between windows in Transform and the View Utility. Only the Color Bar window in Transform can be used to copy a color table. Choose **Color Bar** from the Image menu and then pick **Copy Colors** from the Edit menu. Whatever colors are being used in the color bar window are stored in the Clipboard.

Copy 2x, Copy 4x (Power Macintosh Only)

These copy commands from the Edit menu multiply the size of the copied image as it is drawn for the Clipboard. The reason for this operation is to increase the fine detail in the plots by using more dots. PICT drawings in the Clipboard are always assumed to be based on 72 dots per inch (dpi) of resolution, limiting the amount of detail which can be shown. Copying a plot at actual size is always limited to 72 dpi resolution. We get around this limitation by copying an image at a much larger size than intended and shrinking the plot to its intended size later.

For example, copy a contour plot with the **Copy 2x** command and then paste the result into another application such as MacDraw. In MacDraw, print the image at reduced size, by using the **Page Setup...** command to set the percent reduction to 50%. Now do the same procedure with no percent reduction and the regular **Copy** command. Compare the results. Figure 21-3 shows a similar set of results obtained by resizing the plot in our documentation formatting process. The plot on the left came from **Copy**, the one on the right was produced with **Copy 4x**, then resized to match.

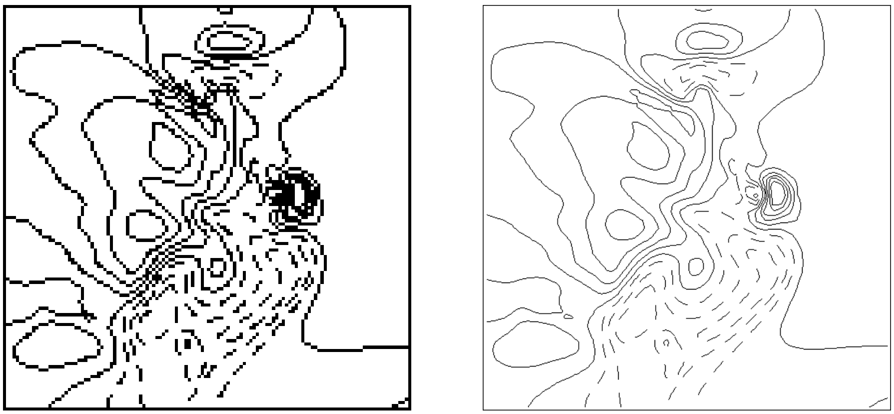


Figure 21-3: High Resolution Copy Results

With **Copy 2x**, twice as many dots of resolution are used for the line calculations. **Copy 4x** uses four times the resolution.

Copy Bitmap (Power Macintosh Only)

Each time a line graph, contour, surface, or vector plot is drawn to the screen, Transform creates a bitmap representation of the lines that make up the plot. In this process, the lines are drawn as dots and polygons are filled with dots of color against a white background. When the drawing is done, the final image is a rectangle of pixels, each set to the appropriate color. **Copy Bitmap** puts the bitmap version of the plot in the Clipboard. All information of lines, polygons, and text labels are forgotten and only the complete image remains.

Raster images are always bitmap representations, so **Copy** should be used; **Copy Bitmap** is disabled.

Paste Command

The **Paste** command, like the **Copy** command, is simple and allows no options. In fact, if the clipboard does not contain a format which is valid for the active window, the **Paste** command may be grayed.

Using Paste in Dataset and Notebook Windows

For dataset windows, the **Paste** command requires a text object on the clipboard. The text must be text numbers, separated by tabs or spaces. Each row must be separated by a carriage return and linefeed.

The numbers on the clipboard are pasted into the dataset starting at the upper left corner of the selection region. There must be sufficient space to the right and down from the selection point to accommodate the size of the data array on the clipboard. If there is sufficient space and the text on the clipboard is in the correct format, the numbers in the dataset are replaced by the numbers on the clipboard.

One convenient use for **Paste** is to copy portions of one array to another. If you have two arrays that represent similar or overlapping data but you need to compare or combine the two, using Copy and Paste may be the easiest way to accomplish the merger.

Using the Notebook window with the clipboard can provide data entry capabilities. If you need to enter a series of numbers into a dataset, enter them into the notebook window first, separated by tabs with carriage returns at the end of each row. Copy the text numbers to the Clipboard, select a data window and use Paste to enter the numbers into the dataset (or in Power Macintosh, use **Paste New** to create a new dataset)

Using Paste in Image and Plot Windows

For plot or image windows, the **Paste** command requires a graphic that can be pasted as an overlay. The preferred format is a Picture (Windows Metafile) or PICT (Power Macintosh) but a bitmap may be used.

Paste As... Command (Windows Only)

The **Paste As...** command allows more flexibility in pasting objects from the Clipboard. The Windows Clipboard is capable of containing more than one kind of object at a time. Using the **Paste As...** command, it is possible to choose which item to paste and how to paste it. Shown below is the Paste As dialog. This dialog always displays two lists, described below.

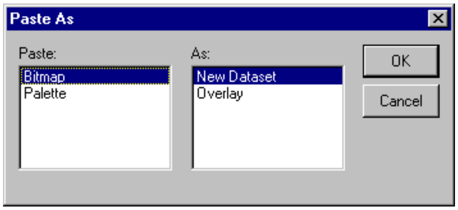


Figure 21-4: Paste As dialog

Paste:

This list shows the objects, by format, which are available on the Clipboard for pasting. In the example above, an object has been copied to the clipboard in two formats: Bitmap and Palette.

As:

This list shows the available ways in which the object might be pasted into Transform, depending on which item is highlighted under Paste:, as well as on the type of the active window. Only formats appropriate for the active window are shown.

For example, a Bitmap object listed under Paste: can be pasted into an image window as on overly, or pasted into a data window as a new dataset (Transform converts the pixel values to data values).

A Palette object can only be pasted into an image window as a new color table.

Paste Overlay and Paste New for Images (Macintosh Only)

Two forms of pasting images are supported, creating a PICT overlay and creating a new dataset. Use the **Paste Overlay** command from the Edit menu to paste the contents of the Clipboard as an overlay. The current window must be an image window. The overlay is drawn into the window, aligned with the image which is already present (note that the drawing in the Clipboard needs to be transparent so that the image beneath it is still visible). If this PICT is not the desired overlay, delete it with the **Clear Overlay** command (make sure the Overlay tool is selected to do this).

Select **Paste New** to create a new dataset and a new image window based on the current Clipboard. The Clipboard PICT is drawn as a Transform image and converted to pixel values in the range 0 to 255. The data window that appears shows the data values derived from this image, one data value for every pixel in the PICT image.

Paste Color Tables (Power Macintosh only)

Any color table, technically a 'clut' resource, in the Clipboard can be pasted into Transform's image windows. This resource contains a full set of 256 color entries just like the color tables in the Color Tables menu.

After using **Copy Colors** to place a color table into the Clipboard, use the **Paste Colors** command from the Edit menu to apply that color table to the current image. Only images and plots that use color can be affected by this process, limiting its use to raster images, color contour lines, and color surface plots.

File Export

Transform lets you to save your data and/or images in one of several formats. You may also export through the Clipboard as discussed in the previous chapter.

Using Save As... to Export Data

Under the File menu, the **Save As...** command brings up the dialog shown below when chosen with a dataset window active. The drop-down menu/radio buttons in the bottom of the dialog allows you to choose the file format to be used. The formats available depend on the active window type.

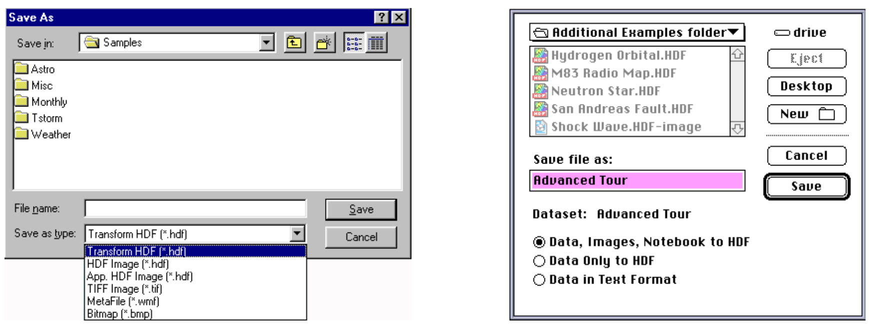


Figure 21-5: Save As Dialog (Windows and Macintosh)

Transform HDF/Data, Images, Notebook to HDF

By default, Transform saves a dataset to an HDF file with an identifier that marks the file as having been created in Transform.

The Notebook contents are saved as an annotation record. All plot or image windows are saved as 8-bit raster images. All custom color tables are saved as 8-bit color palettes. All overlays are stored as Windows Metafile/PICT records.

In addition, Transform saves extra information to allow all windows to be reconstructed as they were when they were saved. Transform files may be used by any other program which supports the HDF format.

HDF Data/Data Only to HDF

With this option, Transform writes the dataset and the Notebook contents (Windows only) into an HDF file, but does not save anything else. Transform will import the data in this file, but it will not recognize the file as a Transform HDF file.

Append to HDF Data File (Windows only)

This is equivalent to the HDF Data File described above, except that the dataset is appended to the existing HDF file.

If you have a Transform HDF File and you append datasets to it using "Append to HDF Data File", when you reopen the file in Transform it will only open the dataset that was originally saved using the Transform HDF option. If you want the Multiple Records dialog to appear (allowing you to open all of the datasets in a file), you must first save the file using HDF Data File, and then append the datasets to that file.

Note

Power Macintosh: you can append datasets and images with the macro language using `saveas()`.

ASCII Matrix/Data in Text Format

With this option, Transform writes the data and scales to an ASCII text file as a scaled matrix. Each row is output on one line with each number separated by tabs. The numbers are formatted exactly as they are in the data window. The first line contains the column headers, and each line starts with the row header for that row. Useful for exporting the data to spreadsheets or word processors. Transform can import this file format as an ASCII matrix file, with the Table Format option selected.

Raw Binary Float File (Windows only)

With this option, Transform writes the data only to a raw binary file. The numbers are saved in IEEE floating point format, 4 bytes per data value. No header information is retained, so a dataset saved in this format with X columns and Y rows will be exactly (X * Y * 4) bytes long.

Transform can import this file as a binary matrix file, but you will have to specify the number of rows and columns and number type at import time, as that information is not stored in the file.

Save As... from Image Windows

The following options are available in the Save As dialog when the command is chosen from a plot or image window:

Transform HDF/Data, Images, Notebook to HDF

Same as described above for dataset windows.

HDF Image/Image to HDF

In Windows, this option in Transform creates an 8-bit raster image which contains the contents of the current window and saves it into an HDF file.

In Power Macintosh, the image record is a compressed 8-bit image with a color lookup table. Useful for exporting to programs that display HDF images. Transform can import this file type, but it converts the 8-bit image into data values in the range 0-255. Note that saving an image to HDF Image saves the overlay but not the axes.

Append to HDF Image

In Windows, this is the same as the HDF Image File described above, except that the image is appended to the existing HDF file.

If you have a Transform HDF File and you append images to it using "Append to HDF Image", when you reopen the file in Transform it will only open the dataset, notebook and image that were originally saved using the Transform HDF option. If you want the Multiple Records dialog to appear allowing you to open all of the images in a file, you must first save the file using "HDF Image File", and then append the images to that file.

Note

Power Macintosh: you can append datasets and images with the macro language using `saveas()`.

Image to PICT (Macintosh only)

Saves the image and any overlay in a PICT file as one combined drawing. Macintosh publishing, drawing, and word processing programs often can import PICT files directly.

Transform can import this file type, but first converts the drawing into an 8-bit representation with data values in the range 0-255. Note that saving an image to PICT saves both the overlay and the axes.

TIFF File/Image to TIFF

With this option, Transform creates an 8-bit raster image which contains the contents of the current window and saves it into a TIFF (Tagged Image File Format) file, one of the most commonly used standard formats for image data. Note that in Power Macintosh, saving an image to TIFF does not save overlays or axes.

Note

In full Transform HDF save files, the dataset is stored as an HDF Scientific Data Set (SDS) record with 32-bit floating point values. The scales and printing formats are stored with the data in the SDS record.

Images, including the raster plots, contour plots, surface plots, vector plots, and line graphs are stored as compressed raster images in their own Raster Image Group (RIG) records, one per image window. The notebook is an annotation record for the dataset. Overlays in Power Macintosh are PICT records which are restored when reading the file.

Finally, the master index is stored in a custom ASCII description record unique to Transform. This master record is not published, but the individual image and data records match the HDF specification exactly and are available for extraction with programs that use the HDF libraries.

Windows Metafile (Windows only)

With this option, Transform saves the plot or image (but not the overlays) into a Windows Metafile, which can be imported into page layout or presentation applications.

Save Command

Save only works for complete files which were saved previously with the 'Transform HDF/Data, Images and Notebook to HDF' option of the **Save As...** command. The Save command rewrites the file with the data, notebook, and all image windows. Usually Transform saves a single dataset per file. However, when you save a file containing vector plots or surface plots with a second color variable, both datasets used to create those image plots are written to the file. When you open the file again, both datasets will appear. Note that if a surface plot and a vector plot both use the same second variable, two copies of the second variable are saved.

Close Commands

To close an image or dataset and free its memory, click on the close box or select **Close** on the File menu.

Selecting **Close All** from the File menu closes all the windows and datasets currently open, hidden or not. It is the same as using the **Close** command for every open window.

When you close a data window you will see the warning dialog if you have made any changes to images, data, or the notebook associated with that data window. Click **Yes** in the dialog to save the changes or to save a new data file and close the window. Click **No** to ignore any changes and close the window. The **Cancel** button cancels the **Close** command.

A file can only have one dataset window and one Notebook window, but may have multiple image windows. When a dataset is removed from memory, all image windows that are dependent on that dataset are also closed. However, image windows may be closed without affecting their associated datasets.

Note that unlike image windows, you may close the Notebook window at any time without losing its contents. The contents are saved with the dataset at all times. Choosing **See Notebook** from the Numbers menu restores an old notebook window to the screen with all of its previous contents.

Note

Vector plots (and some surface plots) are made from two datasets. If any of their supporting datasets are removed from memory, then these image windows are also removed.

Chapter 22:

Using Macros

Transform has a built-in macro language that is very similar to Fortran. The macro language is procedural and consists of three basic elements: functions (i.e., y=3*x), subroutine calls (i.e., call open("filename")) and reserved variables (i.e., axis_auto=true). In Transform, the function and subroutine elements are primarily used to perform calculations, manipulate data, and generate images. The reserved variables give the capability of setting the preferences used by the subroutine elements.

Most of the following example macros are included in the 'Transform/Macros' folder. You can avoid typing in these macros in two ways.

1. Copy the text into an empty Notebook window, highlight the text and choose **Calculate from Notes** from the Numbers menu.
2. From the Macros menu, select **Edit Macros** and then **Import...** from the dialog to add the macro to the Macros menu. Then execute the macro by selecting it from the Macros menu.

The Notebook Window

All of the three basic elements mentioned above contain a variety of macros which can be accessed through the Notebook window. To open the Notebook window select **See Notebook** under the Numbers menu. With the Notebook window active in Windows, the toolbar consists of two popup menus; Datasets and Scalars; in Power Macintosh, the Notebook window contains four pop-up menus: Math Fns, Functions, Datasets and Externals. These menus are designed to expedite entries into the Notebook. Select an item from the pop-up menu (in Windows you have to also press the Enter key) to place it into the Notebook at the location of the cursor. We recommend that you use the pop-up menus as much as possible to avoid typos and syntax errors.

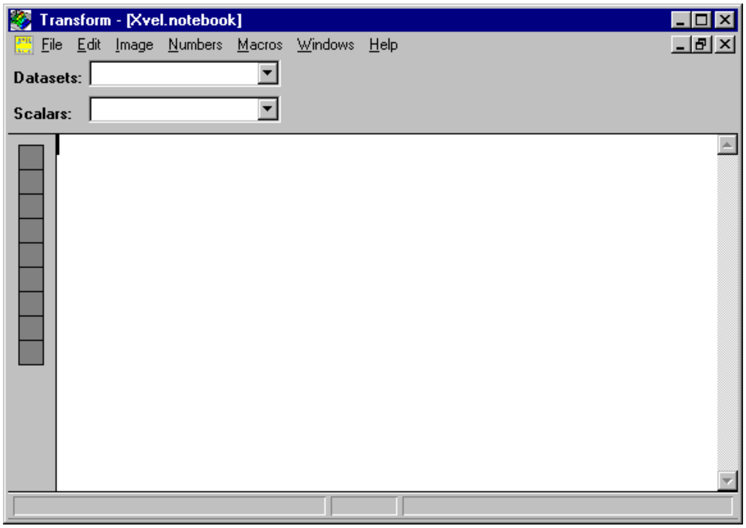


Figure 22-1: The Notebook Window

Datasets Menu

The Datasets menu contains a listing of all available dataset arrays currently in memory. For example, the following datasets were imported into Transform and are currently available.

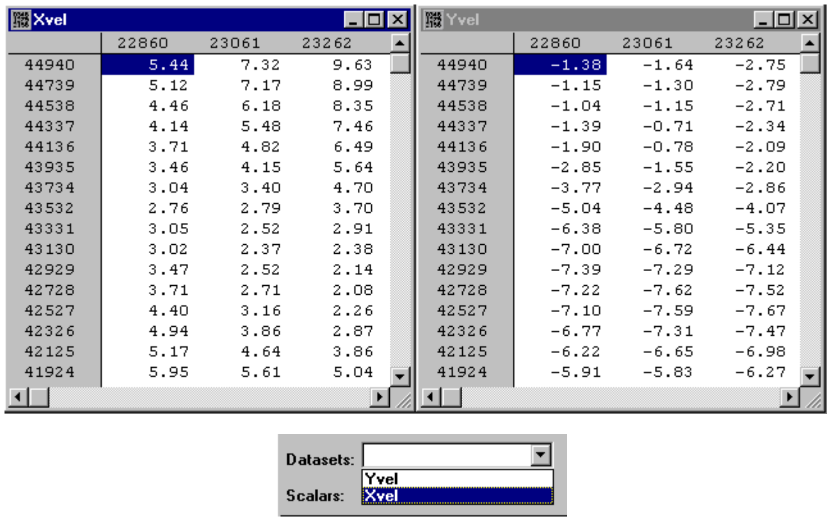


Figure 22-2: The Datasets Menu

Selecting a name and then pressing Enter will paste it into the Notebook at the current cursor location. Transform automatically converts any illegal characters into proper macro syntax (i.e., My File is converted to My_File).

Scalars Menu (Windows only)

The Scalars Functions menu lists all of the reserved variables that are used by Transform. Selecting a name and then pressing the Enter key will paste it into the Notebook at the current cursor location.

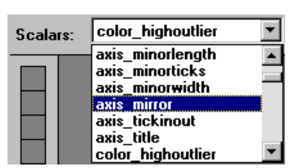


Figure 22-3: The Scalars Menu

In addition to the chapters in this manual, you can consult Transform's on-line help by selecting **Macro Language Reference...** command from the Help menu.

Math FNS Menu (Macintosh only)

The Math Fns menu contains a listing of the mathematical macro functions built into Transform. When you select a function, the name is pasted into the Notebook, followed by a set of parentheses (). The cursor is placed inside the parentheses, ready for you to type the parameter for the function to call. The function name that is initially highlighted is always the last function selected from the menu.

Note

Each individual mathematical macro is described in greater detail in Chapter 23.

Functions Menu (Macintosh only)

The Functions menu contains a listing of the data manipulation macro functions built into Transform.When you select a function, the name is pasted into the Notebook, followed by a set of parentheses (). The cursor is placed inside the parentheses, ready for you to type the parameter for the function to call. The function name that is initially highlighted is always the last function selected from the menu.

Note

Each individual mathematical macro is described in greater detail in Chapter 23.

Externals Menu (Macintosh only)

The External Functions menu lists all of the external functions that have been loaded into Transform. Selecting a name will paste it into the Notebook at the current cursor location. See Appendix F for information about how to create you own external functions.

Executing Macro Commands in the Notebook

Once the Notebook window is open, it takes three steps to execute a macro command. The steps are:

1. Type in a macro command or a series of macro commands.

2. Highlight either a single macro command or series of macro commands.

3. Select **Calculate from Notes** from the Numbers menu.

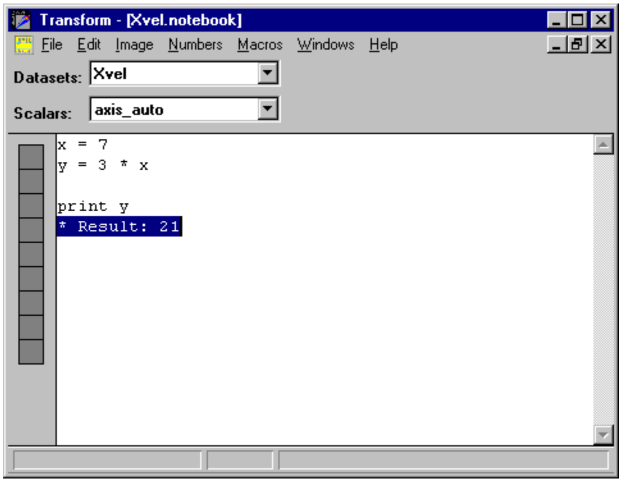


Figure 22-4: Commands Executed in the Notebook (Fig22_04.txt)

Note

Each macro command must occupy one and only one line. However, if your macro wraps to the next line and you did not use a carriage return, Transform will execute the macro as if it occupied only one line.

Tip

If you only want to execute one macro command you do not have to highlight it. You can simply put the cursor at the end of the macro you want to execute and select **Calculate from Notes** from the Numbers menu anywhere in the line.

By selecting **Calculate from Notes**, Transform will execute every highlighted macro command in the Notebook window starting from the top and ending at the bottom. This is very convenient for entering a series of commands and executing them in a specific sequence. For example you may want to open a file, perform some data manipulation, and then save the file. To illustrate this example, we will use the file Yvel.hdf which is included with a typical installation of Transform.

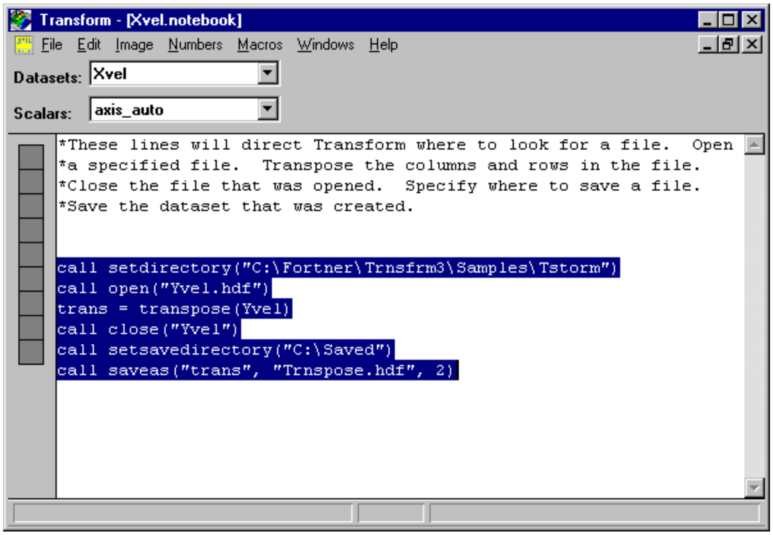


Figure 22-5: Sample Macro Run in the Notebook (Fig22_05.txt)

In this example, the directory "Saved" was created before the macro was executed.

Note

Power Macintosh: the name of the hard drive in these macro examples will not always be the same as your system. Be sure and substitute the name of your hard drive whenever you have to give path information.

In addition, in the example above, instead of ‘call setdirectory’ and ‘call setsavedirectory’ you would use the line ‘call setfolder’ and ‘call setsavefolder’. See the example in the ‘Macros’ directory for more information.

Besides entering macro commands in the Notebook window, it is also possible to enter comments. Comments will enable you to document your work so that it is easily understood at a later date. To enter a comment that starts at the beginning of the line, place an asterisk (*) symbol before your text. If you are entering a comment that follows a macro command that you want to execute, place an exclamation (!) symbol before your text. Any characters entered after an asterisk or exclamation symbol will be ignored by Transform when **Calculate from Notes** is selected.

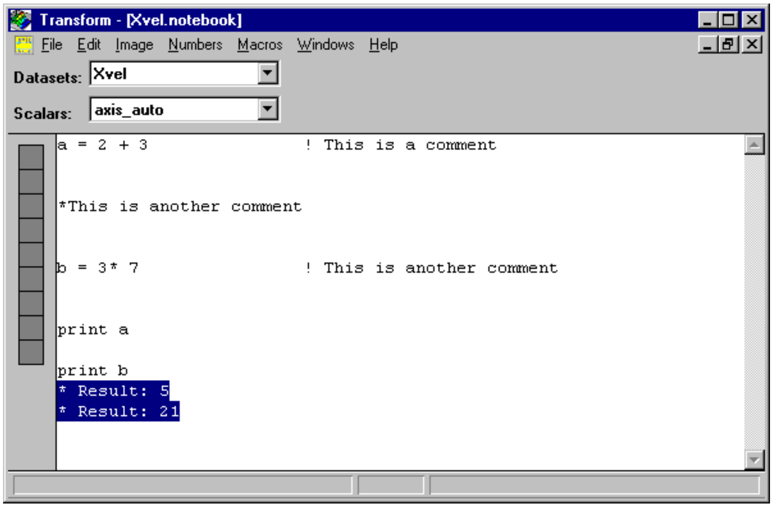


Figure 22-6: Comments in the Notebook (Fig22_06.txt)

There is one last command that can be executed in the notebook window, and it is the **Print** command. The **Print** command will display either scalar variables or the results of scalar calculations in the Notebook window. It is important to point out that values displayed from the **Print** command start with an asterisk and are treated as comments.

The **Print** command will only work for scalar variables and scalar calculations. Any attempt to use the **Print** command for an array will cause an error dialog to appear. In this instance, the error dialog is, "Warning, Incorrect or misplaced parameters in the macro function". This error message makes sense because an array is an incorrect parameter for the **Print** command.

Error Messages

Transform displays an error dialog box any time an illegal operation is attempted. An illegal operation can be caused by typing in the macro wrong or attempting to perform a procedure that is not possible. The dialog box displays the macro in question and gives general information about the problem. Below is a list of some common error messages you may encounter. Keep in mind that this list is not all inclusive.

Windows Errors

- Warning, Function not found
- Warning, Syntax error in line: "___" _ characters from beginning of line
- Warning, Identifier not found. Undefined symbol or array
- Warning, Incorrect or misplaced parameters in macro function
- Warning, This expression requires an array

Macintosh Errors

- Error, incorrect or misplaced parameters in the macro function
- Error, Window not found
- Error, Identifier not found. Undefined symbol or array
- Syntax Error, ## characters from the beginning of the line
- Error, Function not found

Tips for Avoiding Errors

Some tips for avoiding error messages are to ensure:

- Macros are typed correctly.
- CAPITAL letters are used when they refer to a name with capital letters.
- Double quotes (“”) are used around names when using subroutine macros.
- The word **call** is placed in front of subroutine macros.
- You use **var(string)** when you want a string name to point to the dataset itself (i.e., b = "Filename", m = cols(var(b)) returns the number of columns in the dataset called “Filename”).

Components of Macro Commands

Before discussing how to use the macro commands, it is important to identify the components of a command. A macro command consists of Mathematical Operators, Constants, Scalar Variables, Dataset Arrays, and Transform's internal macros. A legal macro command can consist of any number of operators, constants, scalar variables, arrays, and internal macros. Each component is defined below.

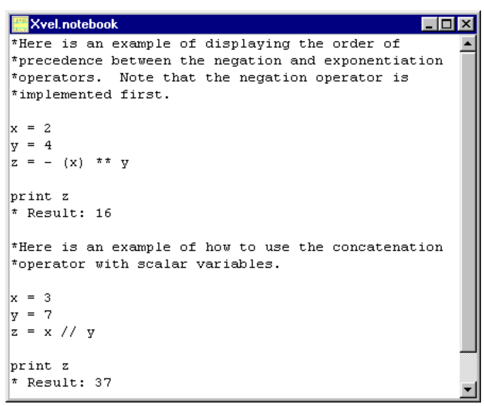
Mathematical Operators

Mathematical operators are listed below in order of precedence:

-	negation (unary minus)
**	exponentiation
* and /	multiplication and division
+	addition
-	subtraction
//	concatenation (of strings)

Table 22-1: Mathematical Operators

To override the precedence, use parentheses ().



Xvel.notebook

```
*Here is an example of displaying the order of
*precedence between the negation and exponentiation
*operators.  Note that the negation operator is
*implemented first.

x = 2
y = 4
z = - (x) ** y

print z
* Result: 16

*Here is an example of how to use the concatenation
*operator with scalar variables.

x = 3
y = 7
z = x // y

print z
* Result: 37
```

Figure 22-7: Mathematical Precedence in the Notebook (Fig22_07.txt)

Constants

Numerical constants are entered with periods (.) for decimal points, and with e representing exponentiation. Integers are converted to floating point values before being used in calculations. Examples of numerical constants are:

```
3.14159
1.0669e+23
```

String constants are always delimited with double quotes ("). Some examples are:

```
"xvel.hdf"
"3.14159"
```

Scalar Variables

Scalar variables are created from assignment statements. For example,

```
pi = 3.14159
```

creates the scalar variable pi which has the value "3.14159". All variables are global and may be referred to any time after they are created.

The values of all variables are stored as ASCII strings, so they may represent names as well as numbers. The maximum length of a variable value is 255 characters. The examples below are all legal expressions:

```
pi = "3.14159"
myfile = "xvel.hdf"
print 1 + "3.14159"
* Result: 4.14159
```

In particular, the `pi` assignment statement given above and the one given previously are exactly equivalent.

The legal characters for a variable name include all of the alphanumerics and the underscore (_). Names cannot start with a number. Periods, slashes, asterisks, and plus or minus signs (. / * + –) are not allowed in variable names. For example, pressure/dens, ke.hdf, and v*u are invalid variable names; but xv102 and K_Energy are valid.

Whenever a variable is used in an expression which requires a number, it is converted to a number before being used. A value of zero is returned if the field does not start with a number.

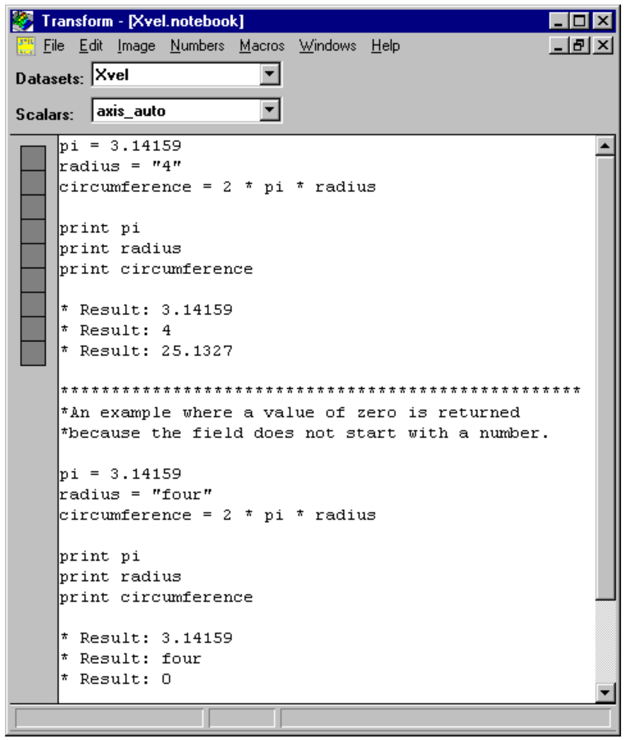


Figure 22-8: The Use of Constants and Strings in the Notebook (Fig22_08.txt)

Dataset Arrays

Any open dataset array may be used in a calculation as you would a scalar variable. For example, if A is an array, then executing the Macro command:

```
B=log(A)
```

will generate a new array called B, where each element is equal to the log of the corresponding element in A.

The following expressions are also legal:

```
B=log(A)+1
C=A*A+2
```

In the first example above, each element in B is set to the log of the corresponding element of A, plus 1. Note that arrays and scalar variables can be combined in the same expression: the scalar variables are treated as if they were arrays with every element set to the scalar value.

In the second example, each element in C is set to the square of the corresponding element in A, plus 2. Note that here, this is not a cross product. If you use more than one array in an expression, they must all have the same row and column dimensions.

The newly created array will have the dimensions, the numerical column and row labels, and the attributes of the first array in the expression. In addition, the minimum and maximum values for the dataset are automatically calculated. A data window is created and appears on the screen with the name from the formula.

Legal characters for dataset names used in expressions are the same as those for scalar variables.

Function Macro Commands

Function commands are used to calculate mathematical expressions and assign the results to either scalar variables or dataset arrays. If the function returns a scalar value or a scalar string, it is defined as a scalar variable. The best way to display the value of a scalar variable is to use the **Print** command. In most calculations, all of the arrays must have the same dimensions. Note that a mathematical expression can contain any number of these functions.

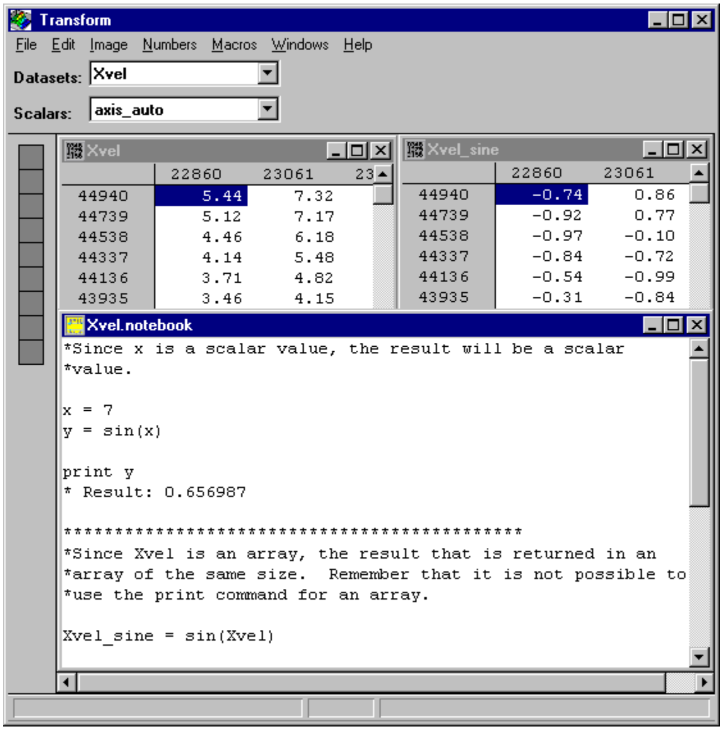


Figure 22-9: Functions in the Notebook that Return Scalars or Arrays (Fig22_09.txt)

If the function returns an array, then a new dataset is created with the variable name. In the examples above, you can see that since “x” is a scalar variable then “y” is also a scalar variable. Since “Xvel” is an array, then a new array called “Xvel_sine” is created.

Function Macro Examples

Mathematical Trigonometric Functions That Create Arrays Or Return Scalars

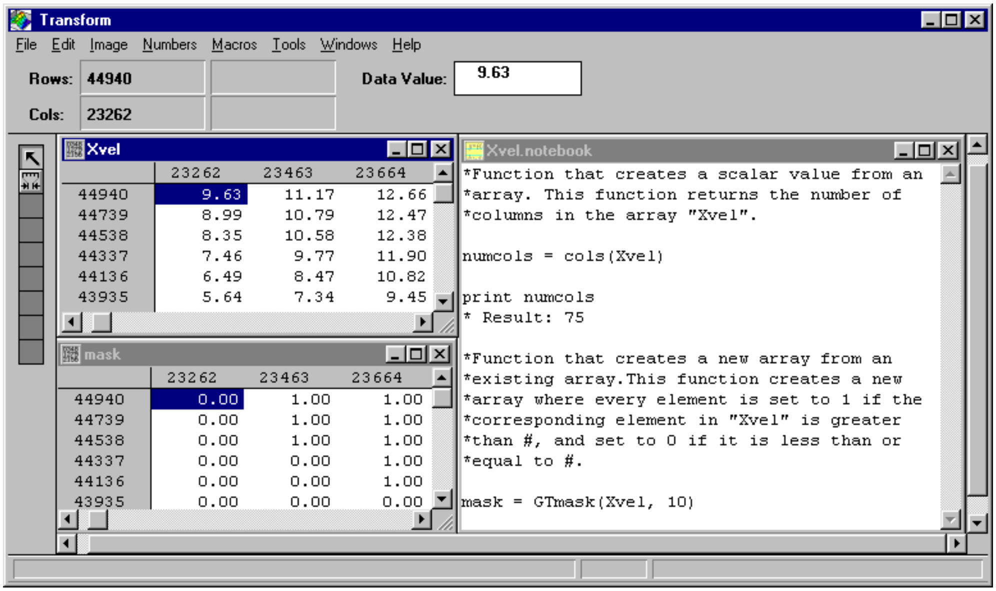


Figure 22-10: Mathematical Trigonometric Functions that Return Scalars or Arrays (Fig22_10.txt)

The **GTmask** function creates a new array from an existing array. In the function in the above figure, every element in the new array is set to 1 if the corresponding element in “Xvel” is greater than 10, and set to 0 if it is less than or equal to 10.

The dataset “Xvel” is the file “Xvel.hdf” which can be found in the ‘Samples\Tstorm’ directory.

In the following figure, the random 10 x 10 dataset "q" was created in the first line of the macro. The x dataset was created with truncated decimals, the y dataset rounded down, and the z dataset rounded up.

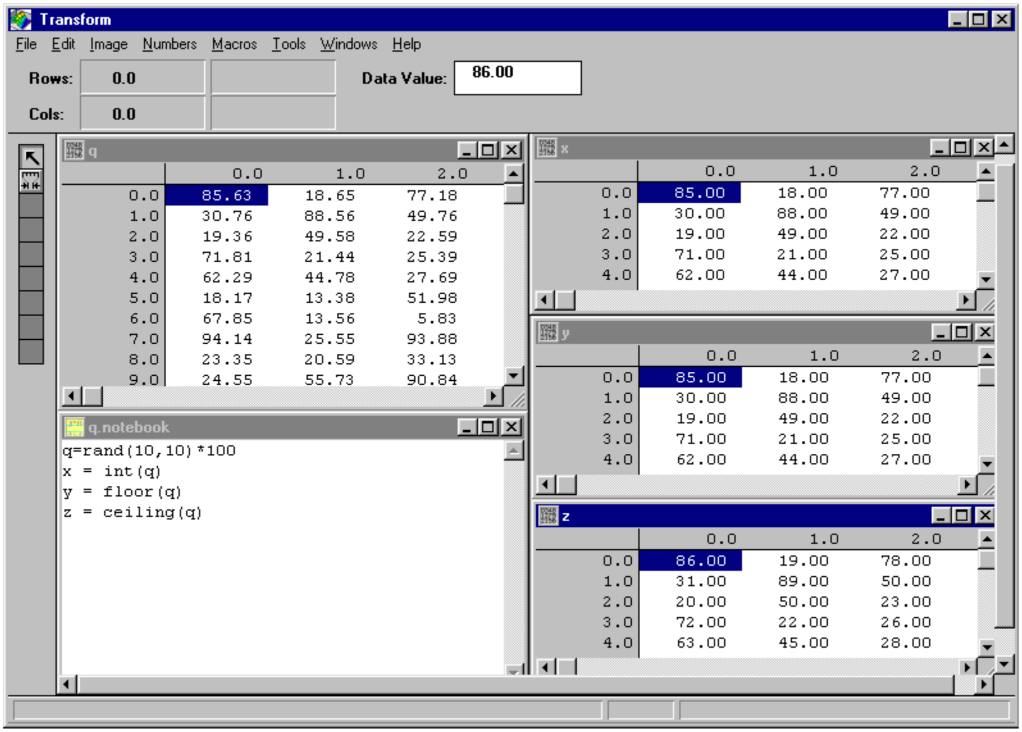


Figure 22-11: Mathematical Functions that Create Arrays or Return Scalars (Fig22_11.txt)

Mathematical Functions that Require an Array and Return a Scalar Value

In the example below, we find the mean, standard deviation, minimum, maximum and sum values for the dataset "Xvel". The **Print** command is used to display the results.

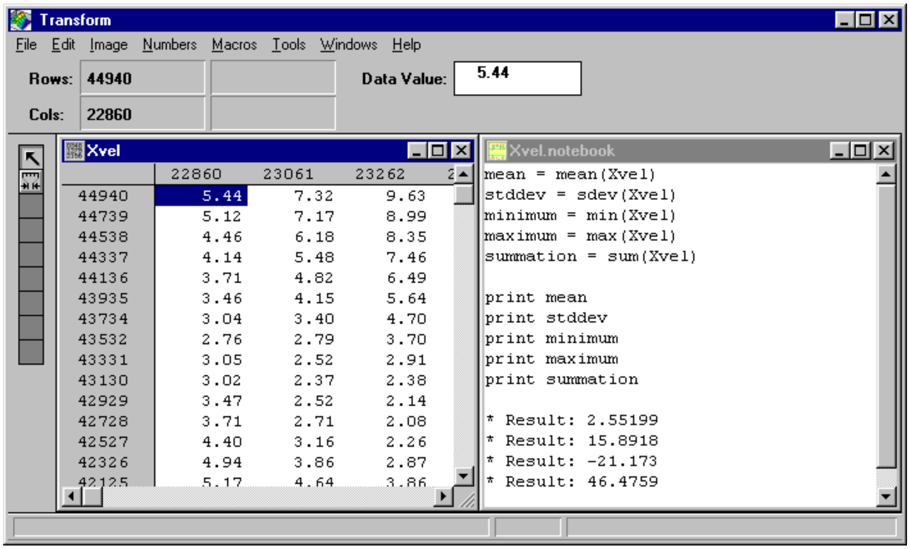


Figure 22-12: Mathematical Functions that Require an Array and Return a Scalar Value (Fig22_12.txt)

The dataset “Xvel” is the file “Xvel.hdf” which can be found in the ‘Samples\Tstorm’ directory.

Data Manipulation Functions

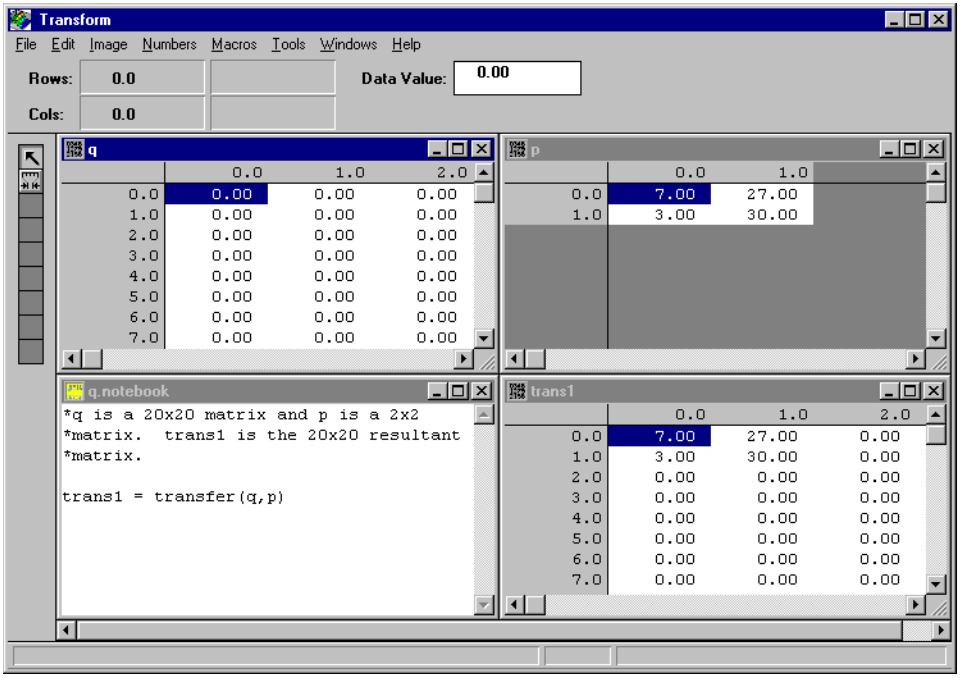


Figure 22-13: Data Manipulation Functions in the Notebook (Fig22_13.txt)

The **transfer(q,p)** function copies the array “q” into a new array “trans1”, then copies array “p” into “trans1” for every cell that they have in common. This is one of the few functions where the arguments p and q can be different sizes.

Data Manipulation Functions that Require an Array and Return a Scalar Value

This example proves that the matrix “trans1” has the same dimensions as matrix “q”.

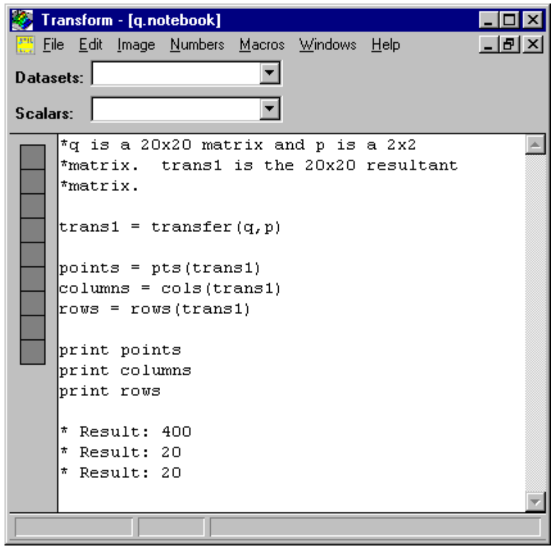


Figure 22-14: Data Manipulation Functions in the Notebook (Fig22_14.txt)

Data Manipulation Functions that Generate Arrays Based on Analytic Expressions

This example creates a 10x10 matrix where every data location has the value of one.

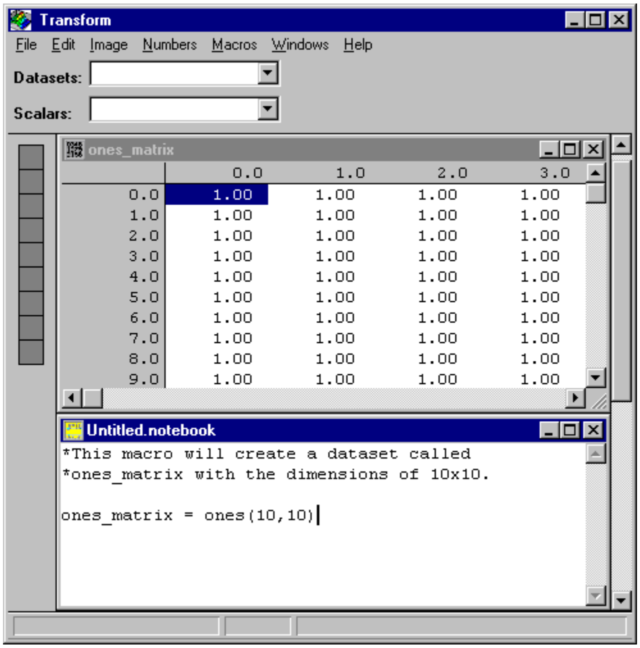


Figure 22-15: Data Manipulation Functions that Generate Arrays Based on Analytic Expressions (Fig22_15.txt)

Advanced Functions

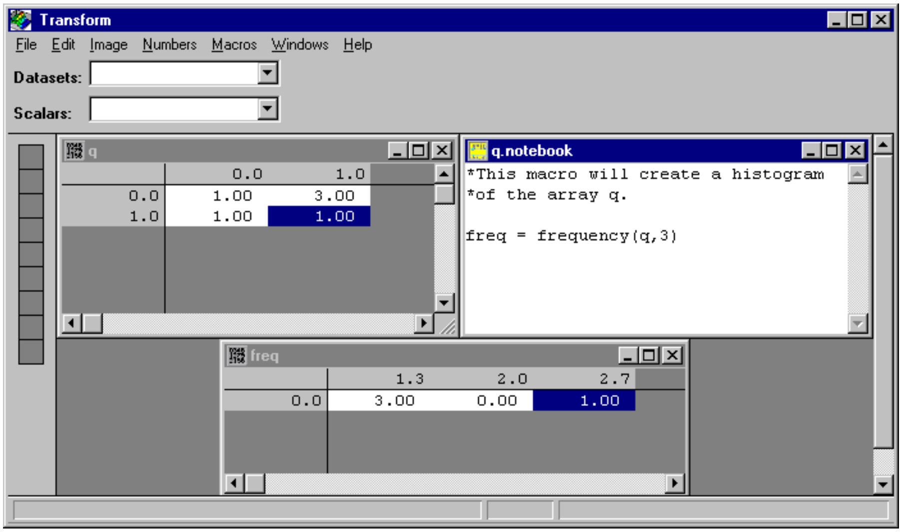


Figure 22-16: Advanced Functions in the Notebook(Fig22_16.txt)

As you can see, Transform inserts the frequency of the matrix data values into the appropriate data location.

Fourier Transform

You can use Transform to take 2D FFTs of images. Using the 'Xvel.hdf' dataset in the ‘Samples\Tstorm’ directory, we generated the image below.

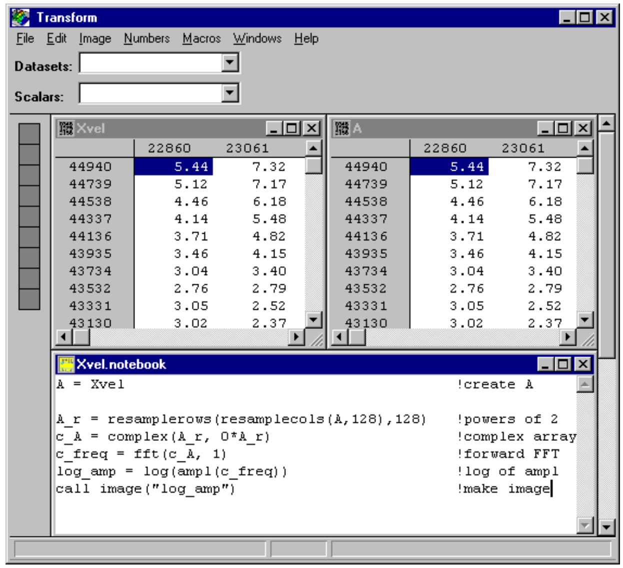


Figure 22-17: 2D Fast Fourier Transform in the Notebook (Fig22_17.txt)

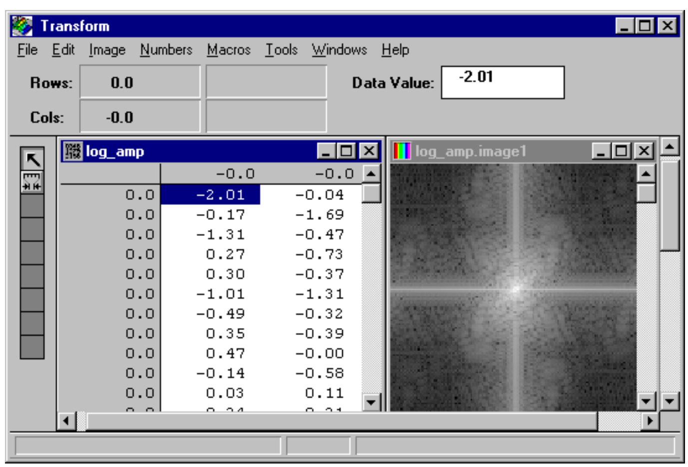


Figure 22-18: Image Created from the Log of the Amplitude

The **resamplerows()** and **resamplecols()** functions resample arrays to the specified size. They are needed here because Transform's FFT routine only works on complex arrays with row and column sizes that are a power of 2 (64,128,256,512...); they need not be the same power of 2. The **complex()** function takes a real array and an imaginary array (here just zeros), and creates a complex array, with each set of two columns representing the real and imaginary parts.

The **fft()** function does the 2D FFT in the forward (+1) direction. The **ampl()** function calculates the real amplitude of a complex array, here the complex frequency array. We took the log of the amplitude so we could make a better image.

Subroutine Macro Commands

Transform’s macro language includes subroutines for performing various tasks, such as creating images, processing data, and running custom macros. In order to use most subroutines, the macro command requires the word “call” to be placed in front of the macro. However, some of the subroutines have the form of functions. These subroutines DO NOT require the word “call” to be placed in front of the macro.

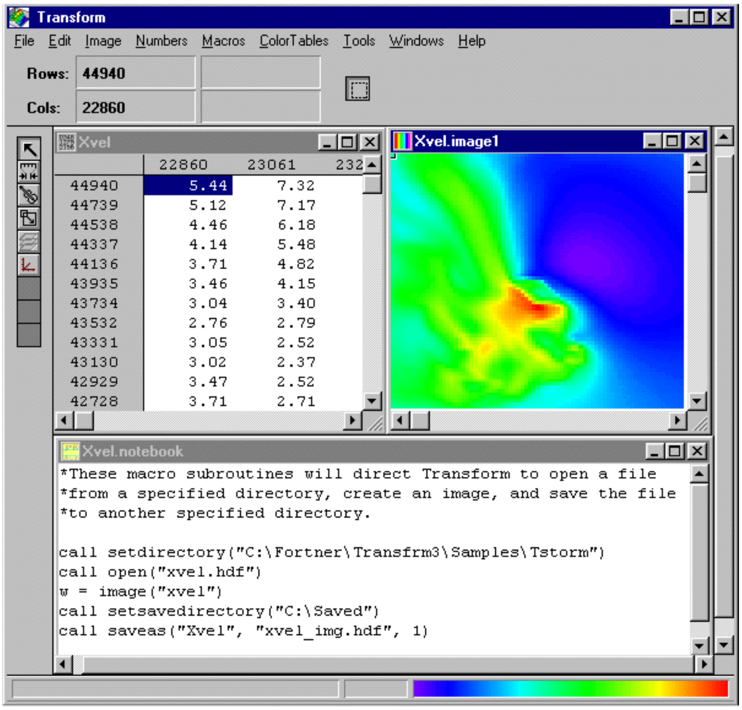


Figure 22-19: Subroutine Macros in the Notebook (Fig22_19.txt)

The dataset 'Xvel' is the file 'Xvel.hdf' located in the ‘Samples\Tstorm’ directory. In the example, the directory 'Saved' was created before the macro was executed.

Tip

You can run custom macros saved on the Macros menu from the Notebook window by using the “call” statement (e.g., call ImageMacro).

Note

Power Macintosh: Be sure and substitute the name of your hard drive whenever you have to give path information.

In addition, in the example above, instead of 'call setdirectory' and 'call setsavedirectory' you would use the line 'call setfolder' and 'call setsavefolder'. See the example in the 'Macros' directory for more information.

It is also important to note that most subroutines require either the datasetname, windowname, or filename to be entered after the macro. Transform uses the following definitions:

datasetname

Name of the dataset window currently active in Transform.

windowname

Name of an image window currently active in Transform.

filename

Name of a file as it is saved on disk.

macroname

Name of macro listed in the Macros menu.

Each of these four names must be enclosed in quotation marks when used in macros. Instead of implicitly stating the dataset (e.g., Xvel), you can also use the reserved variable `currentdataset` which is defined by Transform to be the most recently opened or created dataset.

Tip

In Transform, you will notice that your datasetname does not necessarily correspond to your filename. If you want them to correspond, simply select **Attributes...** under the Numbers menu. There you can change the name of your dataset to correspond to the filename.

Note

In Transform, images that are created from your datasets have the datasetname with the image type and number added to it. The number specifies the order in which the image is created. This is helpful if you create several images from one dataset.

Subroutine Macro Example

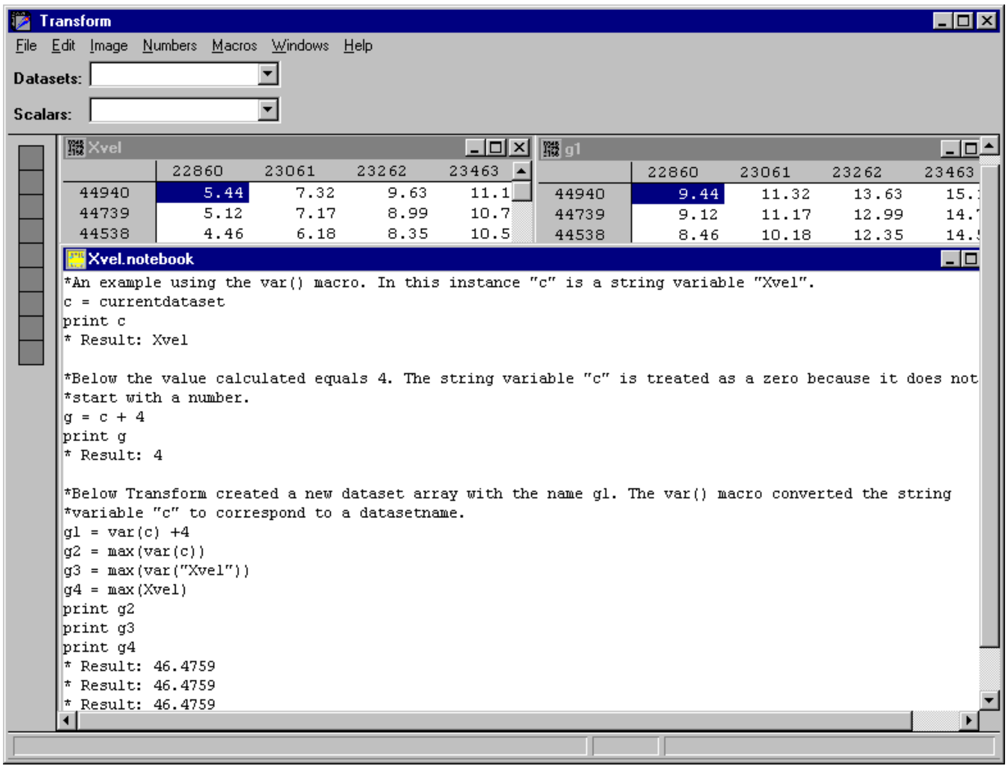


Figure 22-20: Subroutine Macro using var( ) in the Notebook (Fig22_20.txt)

The subroutine **var()** will return the actual dataset (the numbers) from a string representing the datasetname. The subroutine **var()** is most commonly used with **currentdataset**. Using **var()** allows you to use currentdataset within a function that expects a numerical argument. Without the use of **var()**, the function will return an answer of zero, because the value of the string represented by **currentdataset** is zero, or an error saying this function requires an array.

Reserved Variable Macro Commands

In Transform, the reserved variable macros are primarily used to set the preferences of the subroutines. In other words, reserved variables provide the details necessary for a subroutine to carry out it's function. For example, if you attempt to create an image using the subroutine:

```
w=image(datasetname)
```

all of the details about how to build that image are controlled by the reserved variable macros. The reserved variable macros define everything from image size to adding axes. The reserved variable macros can be used from the Notebook window, but are primarily used when editing custom macros.

In this example, the reserved variables defined the surface plot's orientation angle and surface type. Type six is a high resolution, framed color surface plot.

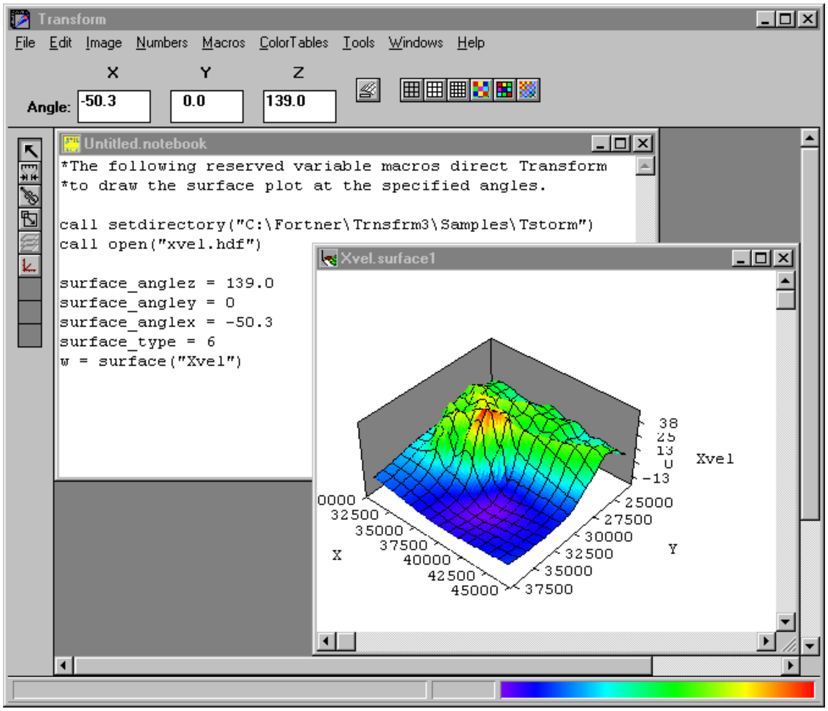


Figure 22-21: Reserved Variable Macros in the Notebook (Fig22_21.txt)

Custom Macros And External Macros

In addition to executing macros from the Notebook, they can also be stored as custom macros on the Macros menu or they can be executed remotely from Mathematica.

Transform may be connected to Mathematica from Wolfram Research via MathLink. You can also exchange data with Transform over MathLink. See Appendix B for details and examples.

Custom macros can be created in one of three ways, depending on whether a notebook window, a dataset array window, or an image window is currently active.

Transform also accepts macro commands from other applications via AppleEvents under System 7 (Power Macintosh only). One convenient scripting environment is AppleScript from Apple computer. See Appendix C for details and examples.

Custom Macros: The Notebook Window

To save macro commands that you have entered in the notebook as a custom macro, make the notebook window the active window on your screen and select **Create Macro** from the Macros menu. Enter a name for the macro, then click **OK**. The custom macro will be added and stored in the Macros menu for future use. If you choose the same name as an existing macro, the new macro will replace it. To use the custom macro at a later date, you can either select it from the Macros menu, type **call** in front of it in the Notebook window, or you can call it from another custom macro using **call** and the macro name.

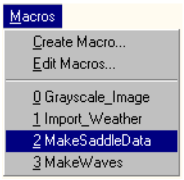


Figure 22-22: Running a Macro from the Macros Menu

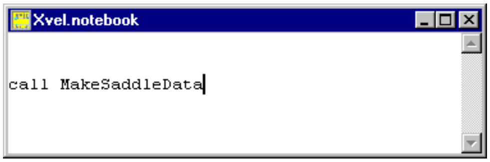


Figure 22-23: Running a Macro from the Notebook (Fig22_23.txt)

Note

Custom macros never accept parameters. All variables in Transform are global, so any information needed by a macro should be stored in variables accessed by the macro before calling the custom macro.

The saved custom macro will work exactly as it did in the Notebook. Note that every line of the Notebook is brought into the custom macro, so make sure that any comments either begin with an asterisk (*) or an exclamation (!).

Tip

If you are executing a custom macro, or several custom macros, and you get an error message, you may want to copy and paste the macro(s) into the Notebook window and run the macros line by line to determine exactly which line is causing the error.

Tip

When creating macros that perform many different functions, it is advisable to split them up into several smaller macros and call them from a main macro. This makes it cleaner to read your macro, and easier to find errors during execution. Also, adding many comment lines will help you remember later on how you created your macros and for what purpose.

Creating A Startup Macro

It is also possible to run a custom macro each time Transform starts. If you save a macro to the Macros menu with the name 'Startup_Macro', it will be executed each time the program launches. To avoid running the macro at startup, either rename or delete the Startup_Macro.

1. Plot Size

This Startup_Macro sets the initial plot size at 400 x 400 points. Save these lines in a macro called Startup_Macro. Exit Transform and then restart the program. Open a file and create an image. The image will be 400 x 400 pixels.

```
image_h = 400
image_v = 400
```

2. Setting Dataset Window Attributes

This Startup_Macro sets the initial characteristics for data windows. It sets type size to 12 points, font to Times, and column width to six characters.

```
pref_textsize = 12
pref_textfont = "Times"
pref_columnwidth = 6
```

3. Setting Default Print Options (Windows only)

This example sets initial printer and print options. It sets the page margins.

```
print_margintop = .5
print_marginbottom = .5
print_marginleft = .5
print_marginright = .5
```

4. Setting Default Directories (Windows only)

This macro sets initial directories for opening and saving files. The saved directory was created before starting Transform.

```
call setdirectory("C:\Fortner\Trnsfrm3\Samples") !where to open files
call setsavedirectory("C:\Saved")  !where to save files
```

It is recommended that you use the **setsavedirectory()** subroutine directly in front of any **saveas()** commands.

5. Setting Default Folders (Macintosh only)

This Startup_Macro sets the initial folders for opening and saving files

```
call setfolder ("drive:Transform") ! where to open files
call setsavefolder ("drive:Saved") ! where to save files
```

Note, it is recommended that you use the **setsavefolder()** subroutine directly in front of any **saveas()** commands.

Custom Macros: Dataset Import Macro

Custom import macros can also be saved for any non-HDF files that you open. To save an import macro, simply open a file and select **Create Macro** from the Macros menu. Enter a name for the macro, then click **OK**. Creating an import macro is a good way to automate the loading of files that have exactly the same format.

Note

The dataset window must be the active window before selecting **Create Macro** from the Macros menu.

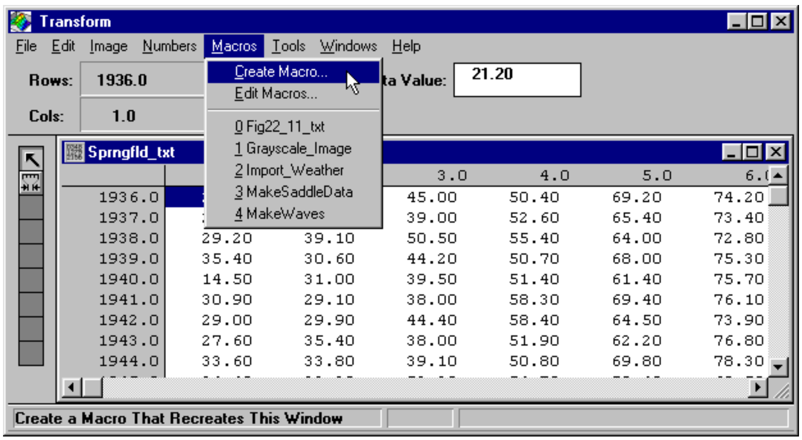


Figure 22-24: Creating a custom import macro

In this example, the imported file is “Sprngfld.txt”. It can be found in the ‘Samples\Monthly’ directory.

Note

If your dataset window was either an imported HDF file or created inside Transform, then the **Create Macro...** option in the Macros menu will be grayed out.

Custom Macros: Image Window

Custom image macros can also be saved for any images that you create. To save an image macro, simply create an image and select **Create Macro** from the Macros menu. Enter a name for the macro, then click **OK**.

Note

The image window must be active before selecting **Create Macro...** from the Macros menu.

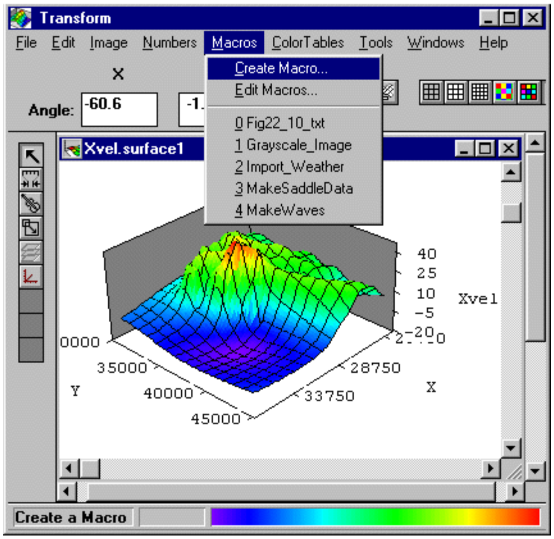


Figure 22-25: Creating a Custom Macro to Generate Surface Plots

In this example, the imported file is 'Xvel.hdf' which can be found in the ‘Samples\Tstorm’ directory.

Editing Custom Macros

Selecting **Edit Macro...** from the Macros menu brings up a dialog that lists all existing macros. The Edit Macros dialog lets you select a macro from a scrolling list and edit, rename, delete, or export it. You can also import macros from text files or create a new macro. Each of these options is described below. When you are finished click **Done** to return.

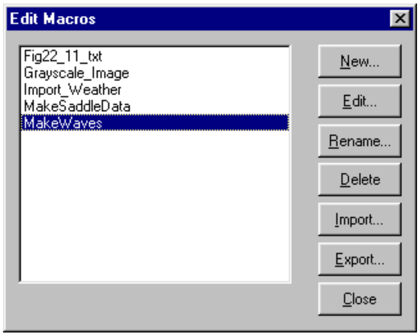


Figure 22-26: Edit Macros Dialog

Tip

It is always a good idea to make a back-up of all your custom macros, using the **Export...** button.

New Macro

Click **New...** to script a new macro from scratch. When you click **New**..., you will be prompted for a name. Enter a name for the new macro and click **OK**. The Macro Editor dialog, described next, will appear, blank, allowing you to enter a new macro.

Edit Macro

To edit an existing macro, select its name from the list in the Edit Macros dialog, then click **Edit...** to display the Macro Editor dialog (you also can double click on the macro name.)

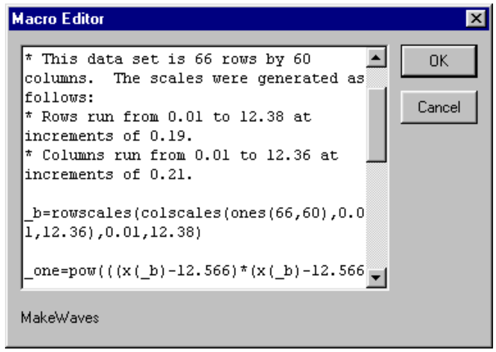


Figure 22-27: Macro Editor Dialog

The selected macro will appear, line by line, in the Macro Editor dialog. This dialog lets you enter new text, or edit existing text. The Edit menu is disabled while the Macro Editor window is active; you can use Control-X, Control-C, Control-V (in Power Macintosh use the Command key instead of the Control key) to cut, copy and paste text in the window. You may find it easier to copy the entire macro into the Notebook window to edit it, and then copy it back into the Macro Editor dialog. When finished, click **OK** to save your changes, or **Cancel** to leave your macro unchanged.

Rename Macro

Click **Rename**... to rename the selected macro. Enter the new name and click **OK**; otherwise click **Cancel**.

Delete Macro

Click **Delete** to remove the selected macro. As soon as you confirm the deletion, the macro is removed from the list permanently.

Import Macro

Click **Import**... to invoke a file browser, which allows you to select a text file and save it in Transform as a custom macro. Imported macros will appear on the Macros menu and in the dialog list.

Note

If you import a file as a macro, or create a new macro from the Edit dialog or the Notebook, any macro on your menu with the same name is overwritten automatically.

Export Macro

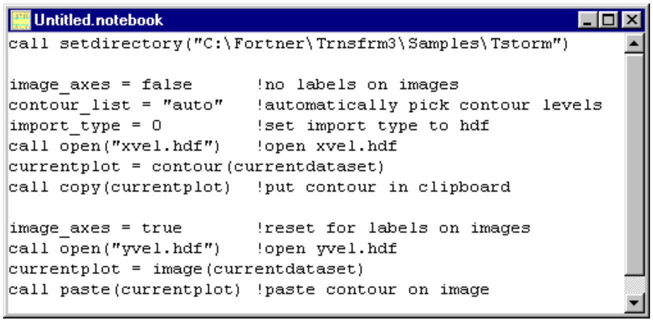
Click **Export**... to save the selected macro as a text file. You can edit this file with any text editor or word processor. Use **Import**... to read the file back in.

Custom Macro Examples

The following provides examples of how to use custom macros.

Contour Overlay

You can use Transform to automate your visualization tasks. The following example opens a data file and creates a contour plot, opens another data file and creates a raster image, then pastes the contour overlay on the raster image.


Untitled.notebook

```
call setdirectory("C:\Fortner\Trnsfrm3\Samples\Tstorm")

image_axes = false      !no labels on images
contour_list = "auto"   !automatically pick contour levels
import_type = 0         !set import type to hdf
call open("xvel.hdf")   !open xvel.hdf
currentplot = contour(currentdataset)
call copy(currentplot)  !put contour in clipboard

image_axes = true       !reset for labels on images
call open("yvel.hdf")   !open yvel.hdf
currentplot = image(currentdataset)
call paste(currentplot) !paste contour on image
```

Figure 22-28: Contour Overlay Macro in the Notebook (Fig22_28.txt)

To execute this macro, highlight every line and select **Calculate from Notes** from the Numbers menu.

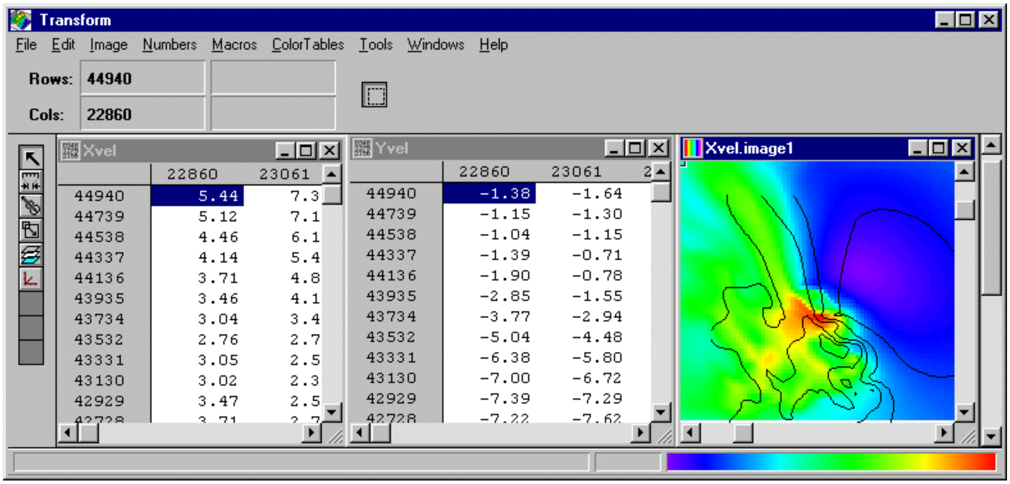


Figure 22-29: Contour Overlay Pasted onto the Raster Image

The first line sets the directory to the location of the data files. Note you may have to change the path to the location of the Transform directory or folder.

The variable **currentdataset** is set by Transform and is always the active dataset. The variable **currentplot** is a reserved variable used to store the name of the current image window.

We also set the reserved variable **image_axes** to false so no labels appear on the contour image, and we set **contour_list="auto"** so that Transform will pick contour levels.

Calculate Area of a Surface Plot

This next macro calculates and displays the surface area of a surface plot. It calculates a numerical approximation of the following integral for a function f(x,y).

$$\int_{x_1}^{x_2}\int_{y_1}^{y_2}\sqrt{1+\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2}\,dx\,dy$$

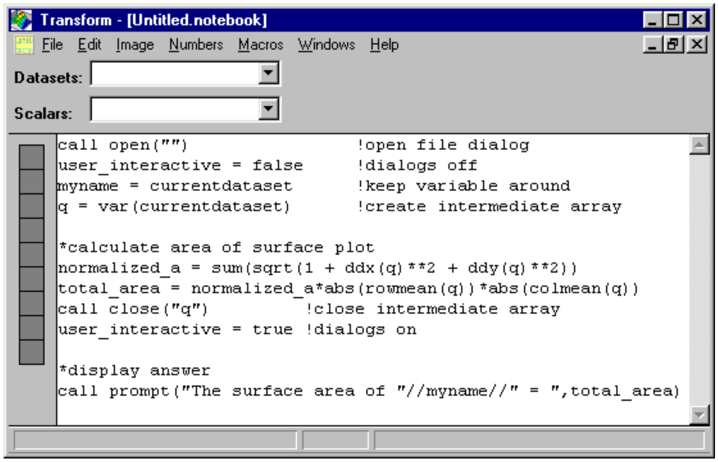


Figure 22-30: Calculate Surface Area Macro in the Notebook (Fig22_30.txt)

The first line calls the open file subroutine; since no path or file name is listed, the file browser will appear on screen, prompting you to select a data file. Open 'Xvel.hdf'.

Note that line 4 creates an array named "q" identical to **currentdataset; var()** is used because **currentdataset** contains a string, while mathematical functions require either an array or scalar as their arguments.

Creating a new array or calling a macro from a dataset will modify currentdataset. In line 3 we stored the name of the array 'Xvel' to use later in other macro commands.

Surface Plot Animation (Windows)

In this example, we create a rotating surface plot animation that can be played back with the Windows Media Player. For more information on the Media Player and its capabilities, see the Microsoft Windows User’s Guide.

First, you need to create a surface plot for the first frame of the animation. Use any sample dataset. Set up the surface plot the way you want it to appear, but for this example, we recommend a framed color surface plot with the following settings: Under the Axes tool, set the Axis Type to **Left Data Axis**, and turn off the Color Bar. Under the Select tool, set the orientation to X=-50, Y=0, and Z=0.

From your new surface plot window, choose **Create Macro...** and use the name "surface_anim" for the macro. Now select **Edit Macros...** from the Macros menu, edit the macro named "surface_anim" and make the changes described below.

Scroll through the macro to find the line for **surface_anglez** and change it to follow the **loop_index. Loop_index** is a special reserved variable set up by the loop command which we will be using next.

```
surface_anglez = loop_index
```

Now scroll to the end of the macro and add the following lines **before** the **user_interactive = true** line. Every time this macro is run, this line appends the resulting image to an HDF file.

```
call setsavedirectory("c:\saved")
call saveas(currentplot, “anim.hdf”, 13)
call close(currentplot)
```

Those are the only two changes to the "surface_anim" macro. This macro is now capable of creating one frame of our animation. We now need to call it multiple times, once for each frame of the animation. You must create the directory 'c:\saved' before you can run this macro.

The **loop** command allows us do just that. It calls any custom macro multiple times. Enter the following command in the Notebook window for the dataset you want to animate. In this example, it counts 0 to 180 by 20, which are the steps we want to use for **surface_anglez**.

```
call loop(0, 180, 20, “surface_anim”)
```

Now the basic animation is complete. In order to play these frames back with the Microsoft Windows Media Player, one more format conversion is needed.

To convert from an HDF image sequence to an AVI file, use the following commands.

```
call makeavi(“anim.hdf”, “anim.avi”)
```

After the macro command., we change the current directory to make sure we can find the file which was just created.

```
call setdirectory("c:\saved")
```

Your animation file is now ready to be played. Launch the Media Player application and open the file "anim.avi" to play it back.

Surface Plot Animation (Macintosh)

In this example, we create a rotating surface plot animation that can be played back with the View Utility or saved as a PICS file to use with Macintosh's QuickTime MoviePlayer.

First, you need to create a surface plot for the first frame of the animation; use any sample dataset. Set up the surface plot the way you want it to appear; for this example, we recommend a framed color surface plot with the following settings: under the Axes tool choose **Surface...**, set the Axis Type to **Left Data Axis**, and turn off the Color Bar. Under the Select tool, choose **Surface...** and **Edit Angles...** and set the orientation to X=-50, Y=0, and Z=0.

From your new surface plot window, choose **Create Macro...** and use the name "Surface_anim" for the macro. Now go to **Edit Macro** from the Macro menu, and edit the macro named "Surface_anim" and make the changes described below.

Scroll through the macro to find the line "surface_anglez = 0" and make it equal to "loop_index". **Loop_index** is a special reserved variable set up by the **loop()** command which we will be using to run the animation.

```
surface_anglez = loop_index
```

Now scroll to the end of the macro and add the following lines before the "user_interactive = true" line. Note in the **call setsavefolder()** line you need to change “Harddrive” to the name of your hard drive. Before you can run this macro you must create a folder on your hard drive called “Saved”. This is where the HDF files will be saved by the macro.

```
out = "anim"//loop//".hdf"
call setsavefolder("Harddrive:Saved")
call saveas(currentplot, out, 4)
loop = loop + 1
call close(currentplot)
```

Every time this macro is run, these lines save each surface plot as an image in an HDF file.

This macro is capable of creating one frame of our animation. We now need to call it multiple times, once for each frame of the animation. The loop command allows us do just that. It calls any custom macro multiple times. Open the Notebook window and make sure it is empty. Enter the following two commands in the Notebook and choose **Create Macro** from the Macro menu. Call this “RunLoop”.

```
loop = 0
call loop(0, 180, 20, "Surface_anim")
```

In this example, it counts 0 to 180 in increments of 20, which are the steps we want to use for surface_anglez. Because the loop_index will be 0, 20, 40, ...180 and we want to save the files out as surface0.hdf, surface1.hdf, surface2.hdf... we use another variable “loop” that we increment by one, not twenty, to create the file names. The RunLoop macro initializes this variable and executes the loop commands. To execute the macro, choose **RunLoop** off the Macros menu.

Now the animation is complete and you have 10 HDF files in the “Saved” folder on your hard drive. In order to play these frames back launch the View Utility, found in the “Transform:Extras” folder, and choose **Animate From Disk** off the File menu. Browse to the “Saved” folder and choose **Current Folder**. The View Utility will open the first file and have the animation buttons at the bottom of the window.

If you want to save the animation to a file format that can be viewed by QuickTime MoviePlayer, choose **Save Animation As...** from the File Menu and enter a file name. Your animation has now been saved as a PICS file. This PICS file can be opened in MoviePlayer and converted to a standard movie file.

Advanced Macros

The following provides a summary of the macros located in the ‘Macros\ Advanced' directory. Note that there are lines of code in each of these macros specific to both Macintosh and Windows platforms. Remove the lines that do not pertain to your platform.

In addition, each macro requires the user to perform several steps to run the macros. It is recommended that you open these text files in a word processor program and read all of the file before you use them in Transform.

FileLoop.txt

This macro is designed to loop through one directory, and open all of the files that are in the directory. Then we perform a function on each file, in this case it's a contour plot, then save each file into a specified directory.

DirLoop.txt

This macro is designed to loop through a series of directories, and open all of the files that are in each directory. After each file is opened, we perform a function on each file, in this case it's a contour plot, then save each file into a specified directory.

Tiff to HDF.txt

This macro is designed to loop through one folder and open all of the TIFF files in it. The image from the TIFF data is then saved into an HDF file in the specified folder.

OpenSaveLoop.txt

This macro is designed to open all of the files in a directory. Then save each file into a specified directory. This macro is useful for converting non-HDF files to HDF.

ImportMacro.txt

Import macros are useful for importing a series of files that are not automatically imported into Transform. You may want to create an import macro to use manually or to incorporate in a loop. All files to be imported must be of the exact same dimension and type. For example, if you have 30 text column files with the same number of rows, columns, header lines, etc... you can automate their import into Transform.

Matrix-Column.txt

This macro will convert your 2D matrix data into column data. Save your new dataset using **Save As...** from the File menu and choose "Data in Text Format". This saves your new data file as a scaled matrix. You must then open it in another program (like Excel), and remove the scales. What is left is text column data. If your data has scales, you would need to add lines of code for the scale values to be used. This is what it does:

Figure A:

	1	2	3	
1	10	20	30	This is your matrix in Transform
2	40	50	60	
3	70	80	90	

Figure B:

	1	2	3	These correspond to x y and z columns
1	1	1	10	
2	1	2	40	With the macro language you could create this matrix
3	1	3	70	
....				
8	3	2	60	
9	3	3	90	

In Noesys, save your new Figure B dataset as an HDF dataset. Open that file in Noesys. You can now highlight your dataset in Noesys and choose **Export Object...** from the File menu. This exports only the data and not the scales and you would not have to bring your exported text into another program to remove the scales.

OrderedPairs.txt

These macros allow you to create ordered pairs from two matrices, a and b. The matrices a and b must be the same size.

matrix a	matrix b
1 2 3	1 2 3
4 5 6	4 5 6
7 8 9	7 8 9

ordered pairs, matrix c

1, 1

2, 2

3, 3

4, 4

5, 5 etc....

WinAnimation.txt (Windows only)

This macro is designed to loop through one directory, and open all of the files in the directory. Then we perform a function on each file; in this case it's create an image from the data. Then each file is saved as an HDF file into a specified directory. The HDF file is then converted to an AVI file that can be used with Microsoft Media Player.

PICSAnimation.txt (Macintosh only)

This macro creates a series of HDF image files that can then be animated using the View Utility, found in the 'Transform:Extras' folder on the Noesys CD. The documentation for the View Utility is also available on the CD in the ‘Manuals’ folder in PDF format. These files can be read with Adobe Acrobat Reader, available in the 'Extras:Adobe' folder on the CD or from Adobe Systems Incorporated at www.adobe.com

In the View Utility, use **Animate From Disk** from the File menu to animate a series of HDF files and then choose **Save Animation As...** to save the animation as a PICS file. The PICS file can then be read into Apple's QuickTime MoviePlayer as a movie file.

Anim3DData.txt (Macintosh only)

This macro creates a series of PICT image files from a 3D dataset. The files can then be animated using the View Utility, found in the 'Transform:Extras' folder on the Noesys CD. In View, use **Animate From Disk...** from the File menu to animate a series of PICT files and then choose **Save Animation As...** to save the animation as a PICS file. The PICS file can then be read into Apple's QuickTime MoviePlayer as a movie file.

Printing.txt

These macros can be used to print an image to be 4"x2" and centered on an 11"x8.5" page. Type **PrintPortrait** and **PrintLandscape** into a Transform Notebook window and save them as macros. Depending on whether you would like to print portrait or landscape, simply execute the appropriate macro before creating your image.

Note

If you create your image using an image macro, it is possible that some of the image macro settings may override the PrintLandscape or PrintPortrait macro settings. To avoid this, make sure that the image macro does not use the “image_h”, “image_v”, “image_marginleft”, and “image_margintop” reserved variables. If it does you can comment them out by adding an asterisk (*) at the beginning of the relevant lines.

AutoTran.txt

In Windows, this macro describes how to drive Transform from outside the program using standard Visual Basic calls. This example launches Transform, and opens two files and makes a vector plot from those two files.

In Power Macintosh, this file describes how to send commands to Transform using AppleEvents. See Appendix C for more information.

Chapter 23:
Macro Reference

This chapter lists and describes the functions, subroutines, and variables that can be used to construct macros.

Macro Functions Reference

Transform has mathematical and data manipulation functions which return data arrays and/or scalar values. If the mathematical expression contains an array, the result will be an array of the same size. If there are no arrays in the mathematical expression, the result will be a scalar value. In most calculations, all of the arrays must have the same dimensions. Note that an expression can contain any number of these functions.

For all functions, the proper syntax to use in the macro language is:

newdataset = function() for functions that create a new dataset.

newvariable = function() for functions that return a scalar variable.

If a function returns an error stating the function requires an array, you may need to use the subroutine **var()**.

The subroutine **var()** returns the actual dataset (the numbers) from a string representing the datasetname. The subroutine **var()** is most commonly used with **currentdataset**. Using **var()** allows you to use **currentdataset** within a function that expects a numerical argument. Without the use of **var()**, the function will return an answer of zero, because the value of the string represented by **currentdataset** is zero, or an error stating this function requires an array.

Mathematical Operators

The mathematical operators, in order of precedence, are:

-	negation (unary minus)
**	exponentiation
* and /	multiplication and division
+	addition
-	subtraction
//	concatenation (of strings)

To override the precedence, use parentheses ().

Variables Set By Transform

These variables are set by Transform. They are very useful in macros, but should not be changed by your macros.

loop_index	Current loop counter. See the **loop()** function call for details.
currentplot	Name of the most recently created plot. In macros saved with **Create Macro...**, this variable is used to store the name of the new plot.
currentdataset	Name of the most recently created dataset. Macros often use this to refer to the file which was just opened.
currentfile	Name of the file being opened. Used internally for opening files. It contains the filename during the open process.

Literals

true and **false**	These values are used to set other keywords to true or false.
INF	The value for infinity, the largest number which can be represented in Transform.

Mathematical Functions

This section provides a list of mathematical functions available in Transform.

Mathematical Trigonometric Functions that Create Arrays or Return Scalars

The following trigonometric functions are available. They will either create a new data array or return a scalar value, depending on the argument in the mathematical expression.

sin(q)	Sine of the argument (q) in radians.
cos(q)	Cosine of the argument (q) in radians.
tan(q)	Tangent of the argument (q) in radians.
asin(q)	Arcsine of the argument (q), returns radians.
acos(q)	Arccosine of the argument (q), returns radians.
atan(q)	Arctangent of the argument (q), returns radians between - /2 and /2.
atan2(q,p)	Arctangent of the argument (q/p), returns radians between - and .
sinh(q)	Hyperbolic sine of the argument (q), in radians.
cosh(q)	Hyperbolic cosine of the argument (q), in radians.
tanh(q)	Hyperbolic tangent of the argument (q), in radians.
dtor(q)	Degrees to radians of the argument (q).
rtod(q)	Radians to degrees of the argument (q).

Mathematical Functions that Create Arrays or Return Scalars

These functions will either create a new data array or return a scalar value, depending on the argument in the mathematical expression.

mod(q,p)	Integer remainder of the argument (q/p).
log(q)	Natural logarithm of the argument (q).
exp(q)	Exponential function of the argument (q).
log10(q)	Base 10 logarithm of the argument (q).
pow(q,p)	Take the argument (q) to the power (p).
sqrt(q)	Square root of the argument (q).
abs(q)	Absolute value of the argument (q).
int(q)	Integer truncation of the argument (q).
floor(q)	Round the argument (q) down to next integer.
ceiling(q)	Round the argument (q) up to next integer.

Mathematical Functions that Require an Array and Return Scalars

The following mathematical functions require a single data array as an argument and always return a scalar value.

mean(q)	Arithmetic mean of the argument (q).
sdev(q)	Standard deviation of the argument (q).
min(q)	Minimum data value of the argument (q).
max(q)	Maximum data value of the argument (q).
sum(q)	Sum of all data values of the argument (q).

Data Manipulation Functions

These functions require at least one data array as an argument, and create a new data array. In most cases, the resultant array always has the same dimensions as the first array used in the argument.

transpose(q)	Transposes (flips diagonally) the data array (q) and exchanges the row and column numerical scales.
rowflip(q)	Exchanges the rows (flips top for bottom), and the corresponding row labels, of the data array (q).
colflip(q)	Exchanges the columns (flips left for right), and the corresponding column labels, of the data array (q).
rowdup(q,row#)	Creates a new data array where every row is equal to the value in row number row# of the array (q), where row numbers begin with 0. The row and column labels in the new array will be identical to those in array (q).
coldup(q,col#)	Creates a new data array where every column is equal to the value in column number col# of the array (q), where column numbers begin with 0. The row and column labels in the new array will be identical to those in array (q).
shl(q)	Creates a new array where the data in array (q) has been shifted to the left one column. The rightmost column will be set equal to the previous rightmost column.
shr(q)	Creates a new array where the data in array (q) has been shifted to the right one column. The leftmost column will be set equal to the previous leftmost column.
shu(q)	Creates a new array where the data in array (q) has been shifted up one row. The bottom row will be set equal to the previous bottom row.
shd(q)	Creates a new array where the data in array (q) has been shifted down one row. The top row will be set equal to the previous top row.
rotrows(q,#rows)	Rotates the rows in array (q) from top to bottom by #rows. Copy the bottom rows to the top as they fall off. Negative numbers move the data up, positive numbers move the data down.

rotcols(q,#cols)	Rotates the columns in array (q) from left to right by #cols. Copy the right side rows to the left side as they fall off. Negative numbers move the data left, positive numbers move the data right.
transfer(q,p)	Copies array (p) into a copy of array (q) for every cell that they have in common. The resulting array is the same size as array (q). This is one of the few functions where the arguments can be arrays of different size.
colsum(q)	Sums each column in array (q) and replaces every element of the column with that sum. Can be used to approximate an integral in the vertical direction (dy).
colavg(q)	Same as colsum, except that the column sums are divided by the number of elements summed.
rowsum(q)	Sums each row in array (q) and replaces every element of the row with that sum. Can be used to approximate an integral in the horizontal direction (dx).
rowavg(q)	Same as rowsum, except that the column sums are divided by the number of elements summed.
recode(q,n)	Replaces all zero values in array (q) with floating point value n.
zapnan(q,n)	Replaces all NAN (Not-A-Number) values in array (q) with floating point value n.
LEmask(q,#)	Returns an array where every element is set to 1 if the corresponding element in array (q) is less than or equal to #, and set to 0 if it is greater than #.
LTmask(q,#)	Same as **LEmask()**, but with less than.
GEmask(q,#)	Same as **LEmask()**, but with greater than or equal.
GTmask(q,#)	Same as **LEmask()**, but with greater than.
EQmask(q,#)	Same as **LEmask()**, but with equal to.
NEmask(q,#)	Same as **LEmask()**, but with not equal to.

Data Manipulation Functions that Require an Array and Return a Scalar

These functions require an array as an argument, and always return a scalar value.

pts(q)	Number of points in array (q).
cols(q)	Number of columns in array (q).
rows(q)	Number of rows in array (q).
colrange(q)	Range of column scale values in array (q).
rowrange(q)	Range of row scale values in array (q).
colmean(q)	Mean of distance between columns in array (q).
rowmean(q)	Mean of distance between rows in array (q).
colsdev(q)	Standard deviation of distance between columns in array (q).
rowsdev(q)	Standard deviation of distance between rows in array (q).

Data Manipulation Functions that Generate Arrays Based on Analytic Expressions

These functions are useful for generating new arrays based on analytic expressions.

c(q)	Returns an array where every number in a column is equal to the numerical label from array (q) for that column.
r(q)	Returns an array where every number in a row is equal to the numerical label from array (q) for that row.
x(q)	Synonym for the function c(q).
y(q)	Synonym for the function r(q).
zeros(a,b)	Creates an array of (a) columns and (b) rows, where every element in the array is equal to 0.
rand(a,b)	Creates an array of (a) columns and (b) rows, where every element is a random number between 0 and 1.
ones(a,b)	Creates an array of (a) columns and (b) rows, where every element in the array is equal to 1.

Advanced Functions

Transform allows you to use these more complex data manipulation features in macros, most of which are available on menus.

colscales(q,startval,endval)	Returns a dataset of the same size as array (q), with the same data, except that column scales are changed to go from startval to endval. The new scale values are evenly spaced. This macro is the same as the **Generate Scales** command from the Numbers menu.
fillmissing(q)	Does a Fill Missing Data operation on array (q). Make sure all of the fill_ reserved variables are set appropriately before calling this function. Same as the **Fill Missing Data...** command from the Numbers menu.
frequency(q,n)	Creates a histogram of array (q). The function first divides the number range into n equally spaced bins, and then counts the number of data points in the array (q) which fall in each bin. Returns an array with a single row and n columns containing the counts.
resamplecols(q,n)	Resamples array (q) in the horizontal direction, stretching the array to n columns. To control resampling options, it uses the current value of keyword variables resample_type and resample_scalecol. Same as the **Resample Data...** command from the Numbers menu.
resamplerows(q,n)	Resamples array (q) in the vertical direction, stretching the array to n rows. To control resampling options, it uses the current value of reserved variables resample_type and resample_scalerow. Same as the **Resample Data...** command from the Numbers menu.

rowscales(q,startval,endval) Returns a dataset of the same size as array (q), with the same data, except that the row scales are changed to go from startval to endval. The new scale values are evenly spaced. This macro will operate the same as the **Generate Scales** command from the Numbers menu.

smooth(q,n) Smooths array (q) using n passes. Same as the **Smooth Data...** command from the Numbers menu.

streamfunction(h,v) Calculates the non-divergent stream function for velocity components h (horizontal) and v (vertical). A contour plot of the result gives you streamlines for the velocity field. This function works only for 2D, irrotational (no source or sine), incompressible flows.

Fast Fourier Transforms

Transform provides a set of functions for dealing with complex variables and Fast Fourier Transforms.

Complex Numbers

You can create complex arrays using the **complex(r,i)** function. Here **r** is an array containing real values, and **i** is an array containing imaginary values. The resulting complex array has twice as many columns as either the real or imaginary arrays (which must be the same size). The columns alternate real and imaginary numbers. For instance, the real array:

```
r1  r2  r3
r4  r5  r6
```

and the imaginary array

```
i1  i2  i3
i4  i5  i6
```

becomes the combined complex array:

```
r1  i1  r2  i2  rr3  i3
r4  i4  r5  i5  r6   i6
```

In the list below, **c** refers to complex arrays (that is, double width arrays that alternate real and imaginary components).

complex(r,i)	Combines the real array (r) and the imaginary array (i), both of the same dimensions, into a complex array.
real(c)	Extracts the real part of complex array (c) and makes a new array with only the real values.
imag(c)	Extracts the imaginary part of complex array (c) and makes a new array of only the imaginary values.

complexap(a,p) Forms a complex array, given two arrays representing the amplitude and phase of those complex numbers. The phase is assumed to be in radians.

ampl(c) Calculates the amplitude of each complex number in the complex array (c), and returns a real array with just the amplitude values.

phase(c) Calculates the phase of each complex number in the complex array (c), and returns a real array with just the phase values, in radians.

fft(c,n) Performs a 2D Fast Fourier Transform on the complex array (c). The parameter n=1 for a forward transform and =-1 for an inverse transform. Datasets must be square and the number of rows must be a power of 2. A square complex dataset has twice the number of columns as it does rows because every two columns represents one complex number.

fftrows(c,n) Performs a 1D Fast Fourier Transform on every row of complex array (c). The parameter n=1 for a forward transform and =-1 for an inverse transform. The number of columns must be a power of 2.

fftcols(c,n) Performs a 1D Fast Fourier Transform on every column of complex array (c). The parameter n=1 for a forward transform and =-1 for an inverse transform. The number of rows must be a power of 2. Every pair of columns is processed as one column of complex numbers.

If your array does not have dimensions which are a power of 2, you will need to use the **Resample Data...** command to create a new array of the proper size from your current data.

Kernel Functions

The following built-in functions approximate the various derivatives. The x derivative is the slope from left to right, the y derivative is the slope from top to bottom. For details on their exact operation, see Appendix E.

ddx(q) dq/dx of array (q) - difference from left to right

ddy(q) dq/dy of array (q) - difference from top to bottom

d2dx(q) d^2q/dx^2 of array (q) - second derivative from left to right

d2dy(q) d^2q/dy^2 of array (q) - second derivative from top to bottom

lap(q) 5 point laplacian of array (q)

lap5(q) same as lap(q)

lap9(q) 9 point laplacian of array (q)

Generic Kernel Functions

Generic kernel operations are implemented with the kernel function:

kernel(q,k) Performs a generic 3x3 kernel convolution on array (q). Array (k) is the convolution kernel and it must be 3x3.

Listing Of Macro Subroutines

In the listing of macro subroutines, the following conventions are used; each of these four names must be enclosed in quotation marks when used in macros:

datasetname	Name of a dataset currently open in Transform (for example, "Xvel").
windowname	Name of a currently open window (for example, "Xvel.contour1").
filename	Name of a disk file (for example, "XVEL.HDF").
macroname	Name of macro listed under the Macros menu.

Most subroutines require the use of the word "call" in front of the subroutine. Those that do not are presented here with syntax variable = **subroutine()** where the subroutine returns information into the variable. For all the other subroutines the proper syntax is **call subroutine()**. This is also the proper syntax for calling custom macros that are saved on the Macros menu.

w=image(datasetname)
w=interpolated(datasetname)
w=line(datasetname)
w=contour(datasetname)
w=surface(datasetname)
w=vector(datasetname, datasetname, 0) **(Power Macintosh only)**
w=vector(datasetname, datasetname, Flag) **(Windows only)**

```
if Flag = 0
   1st parameter is horizontal axis
   2nd parameter is vertical axis
if Flag = 1
   1st parameter is vertical axis
   2nd parameter is horizontal axis
```

Creates an image window from the dataset **datasetname**, using the current reserved variable values and returns the name of the new window. Executing one these calls is exactly the same as selecting the command from the Image menu.

w=histogram(datasetname) **(Windows only)**

Creates an image window from the dataset **datasetname**, using the current reserved variable values and returns the name of the new window. Executing one of these calls is exactly the same as selecting the command from the Image menu.

call addaxis(windowname, whichaxis)

Adds axis type whichaxis to window **windowname**. The parameter whichaxis=1 for the X axis, =2 for the Y axis, =3 for the Z (data) axis, and =4 for the Color Bar.

call addlabel(windowname, x, y) **(Power Macintosh only)**

Adds a contour label at x pixels in and y pixels over from the lower left corner of the image, to window **windowname** (which must be a contour window). Used internally for saving contour plot labels.

call addscalelabel(windowname, x, y) (Windows only)

Adds a contour label to the contour level closest to the coordinate (x,y) on the contour plot **windowname**. The x and the y locations for the label are given in scale coordinates rather than pixels.

call beep

Beeps the speaker once.

call close(windowname)

Closes window **windowname**. If you close the dataset window, all windows associated with that dataset are closed. An error will occur if you try to close a dataset from the Notebook window associated with that dataset.

call closeall **(Windows only)**

Same as **Close All** from the File menu. Can only be executed if saved as a custom macro and run from the Macros menu. If user_interactive=false, every window will be closed without being saved.

call copy(windowname or datasetname)

Copies the data or image in window **windowname** or the selected data in the dataset **datasetname** to the Clipboard. Same as **Copy** from the Edit menu. Use with **Paste** to paste data or create an overlay.

call createcolortable(datasetname, tablename) **(Windows only)**

Converts the dataset **datasetname** to a color table and adds it to the list of custom color tables with the name **tablename**. The dataset must have 256 rows and 3 columns. Columns 1, 2 and 3 contain the red, green, and blue components of the color table respectively, with values between 0 and 256.

call deletemacro(macroname)

Deletes the macro macroname from the Macros menu. Does nothing if the named macro is not found.

call exechdfmacro(filename, macroname)

Executes the macro macroname from the HDF file filename.

call execmacrofile(filename)

Executes a macro from the text file **filename**. The entire contents of the text file are executed as a single macro. Use with **setdirectory()** (Windows) or **setfolder()** (Power Macintosh) to specify where the file is located.

call extractselection(datasetname)

The currently selected region of the dataset is extracted to create a new dataset. The new datasetname has the same base name with the suffix "_x". Works the same as the **Extract Selection** command from the Numbers menu. Use **setselection()** to specify which region of data to extract.

s = getdatasetname(windowname)

Returns the dataset name for the dataset window.

n = getfilecount()

Counts the number of files in the current folder. Used for processing sequences of files with the loop command, or in scripts. This subroutine should be directly preceded by the **call setdirectory()** (Windows) or **callsetfolder()** (Macintosh) subroutine. It takes no argument in the ()'s and returns the number of files in the variable "n" which can then be used in a macro. Syntax example:

```
call setdirectory("d:\myfolder")  !Windows syntax  or
call setfolder("HardDrive:Folder")  !Macintosh syntax
n = getfilecount()
```

f = getnthfilename(filenumber)

Returns the filename for file number filenumber in the current directory/folder. The parameter **filenumber** is the number of the file, starting with 1, when they are arranged in the folder in alphabetical order. Useful for importing a sequence of files from a folder.

s = getvalidname(datasetname) (Windows only)

Returns a unique name for the given datasetname. Useful in macros to avoid conflicting with existing dataset names.

call loadhdfdataset(filename, reference#)

Loads the dataset record with reference number reference# from the HDF file filename. Used internally for opening HDF files.

call loadhdfnotes(filename, dataset#, reference#) (Windows only)

Given filename, a dataset number, and an HDF reference number, reads an annotation record as Notebook contents to the given dataset. Used internally for opening Notebook records.

call loadhdfoverlay(windowname, filename, reference#)

Loads the overlay record with reference number **reference#** from the HDF file filename and places it on top of the image in window **windowname**. Used internally for opening saved overlays.

call loadpictoverlay(windowname, filename) **(Power Macintosh only)**

Loads an overlay from the PICT file **filename** and places it on top of the image in window **windowname**.

call loop(start, stop, increment, macroname)

Executes the macro **macroname** as the variable **loop_index** goes from the value **start** to the value **stop** in increments of **increment**. The macro macroname is executed (stop-start)/increment+1 times. The **macroname** must be in double quotes. The variable **loop_index** can be used to reference files to be opened or saved.

call makeavi(filename, aviname) (Windows only)

Given the name of an HDF file and a name for a new AVI file, read every 8-bit image from the HDF file and create an AVI file with those images. AVI files contain animations that may be played back with the Windows Media Player. To use these files with Windows Media Player, the files must end in “.avi”.

call open(filename)

Opens the disk file **filename**, bringing up any datasets and windows which were stored in the file. Used both for Transform-saved files and for imported files. Looks for the file in the current directory/folder. If no file name is given, a dialog prompts you for the required information.

call paste(windowname or datasetname)

Pastes contents from the clipboard onto the image in the window **windowname** or the data from the clipboard into the selected region of the data window **datasetname**. Same as **Paste** from the Edit menu.

call pastenew(windowname or datasetname) **(Power Macintosh only)**

Pastes an image from the clipboard to create a new dataset called "Untitled" and a new image from that untitled dataset. For a dataset, pastes data from the clipboard into a new dataset called "Untitled". Same as **Paste New**.

s = prompt(promptstring, defaultstring)

Prompts you with the text string **promptstring** and waits for text input. Returns the string typed in into the variable **s**. If no changes made, it returns the string **defaultstring**. Unless they are variables, **promptstring** and **defaultstring** need to be in double quotes.

call saveas(windowname, filename, save_type)

Saves the window windowname to the disk file filename in the current save directory/folder. If user_interactive=false, it will overwrite any file of the same name without prompting. For types 4, 5, 6, 8, (and additionally 13, 14, and 15 in Windows), the **windowname** must be the name of the image window (or you can use currentplot) and not the dataset window. The parameter **save_type** is:

=0	Prompts the user
=1	Saves data, images, and notebook to HDF file
=2	Saves data only to HDF file
=3	Saves data to text file
=4	Saves image to HDF file
=5	**(Power Macintosh only)** Saves image to PICT file
=6	Saves image to TIFF file
=7	Appends data, notebook to HDF file

=8 Appends image to HDF file

=13 **(Windows only)** Appends image to HDF file, same size as image(s) already in file—useful for animations.

=14 **(Windows only)** Saves image to Windows bitmap

=15 **(Windows only)** Saves image to Windows metafile

=16 **(Windows only)** Saves data to raw binary IEEE float file

If you have a Transform HDF File (equivalent to using **saveas** option 1, or "Data, Images, Notebook to HDF" from the File menu's Save As dialog) and you append datasets or images to it using **saveas** option 7, 8, or 13 (Windows), when you reopen the file in Transform it will only open the dataset, notebook and image that were originally saved using the Transform HDF option. If you want the Multiple Records dialog to appear (allowing you to open all of the datasets or images in a file), you must first save the file using HDF Data File (**saveas** option 2 or 4), and then append the datasets or images to that file.

call savecolortable(datasetname, tablename) **(Power Macintosh only)**

Converts the dataset **datasetname** to a color table and adds it to the Color Tables menu with the name **tablename**. The dataset must have 256 rows and 3 columns. Columns 1, 2 and 3 contain the red, green, and blue components of the color table respectively (1-255). **tablename** must be in double quotes.

call savemacro(windowname, macroname)

Creates macro **macroname** from the image window **windowname** and adds it to the Macros menu. Same as **Create Macro...** on the Macros menu.

call setdirectory(directoryname) **(Windows only)**

Specifies the directory name and path in **directoryname** for opening files. You can use a full or partial path name with backslash separators.

call setfolder(foldername) **(Power Macintosh only)**

Specifies the folder name and path in foldername for opening files.You can use a full or partial path name with colon separators, e.g. “HD80:myfiles:myfolder”. If no folder name is given, a dialog prompts you for the required information.

call setformats(datasetname, string, string) **(Windows only)**

Given a dataset name and two Fortran-style format strings, set the data format to the first and the scale format to the second.

call setoverlayrect(windowname, left, top, right, bottom)

Modifies the location of the overlay in the image window **windowname**. The parameters left, top, right, and bottom refer to the pixel locations of the overlay rectangle relative to the upper left corner of the canvas. Normally this call is not needed, as executing a **paste** call will automatically set the overlayrect to match the image. Used internally.

call setnames(datasetname, newdataname, newxname, newyname)

Changes the name of dataset **datasetname** to **newdataname**, and changes the X axis and Y axis scale names to **newxname** and **newyname** respectively. If an empty string ("") is passed a parameter, that name is not changed.

call setsavedirectory(directoryname) **(Windows only)**

Specifies the directory name and path in **directoryname** for files saved with **saveas**. You can use a full or partial path name with backslash separators.

call setsavefolder(foldername) **(Power Macintosh only)**

Specifies the folder name and path in foldername for files saved with saveas(). You can use a full or partial path name with colon separators. For example, "HD80:myfiles:myfolder". If no folder name is given, a dialog prompts you for the required information.

call setselection(datasetname, left, top, right, bottom)

Sets the selection region for dataset datasetname. The parameters left, top, right, and bottom refer to the row and column index numbers of the dataset, starting with 0,0 for the upper left corner. All windows for this dataset are updated, and if **Synchronize** is turned on, all matching datasets also update.

d = var(windowname)

Returns the dataset from a string representing the **datasetname** or a **windowname**. The subroutine **var( )** is most commonly used with **currentdataset**. Consider the following example:

```
sinx = sin(var(currentdataset))
```

Since the function sin expects a numerical argument, **var ( )** is used to convert the string in currentdataset to the actual data. A new dataset named 'sinx' is created such that each element in 'sinx' is the sine of the corresponding element in **currentdataset**. Without the use of **var ( )**, the **sin** function would return an answer of zero, since the value of the string represented by currentdataset is zero.

Macro Variables Reference

Listed here are the reserved variable names for use in Transform macros. These reserved variables are used by the macro subroutine calls, listed earlier in this chapter.

Values for all variables are stored as ASCII strings which may be up to 255 characters in length. When a numeric value is required by Transform, the string is converted into a number. A value of 0 is returned if the field does not start with a number. Many variables need to be set to the string "true" or the string "false". You can instead use the reserved variables **true** and **false**.

Transform Reserved Variables

axis_auto=true

When =true, Transform automatically calculates the axis labels, format, and spacing, and overrides **axis_labelspacing, axis_min, axis_max, axis_labelmin, axis_labelmax and text_format**. When =false, those same parameters are used to calculate the axis labels.

axis_autospacing=true

When =true, Transform automatically calculates the label spacing (increment), and overrides axis_labelspacing. When =false, **axis_labelspacing** is used to calculate the label spacing.

axis_autotextformat=true

When =true, Transform automatically calculates the printing format for axis labels, and overrides **text_format**. When =false, **text_format** is used to set the printing format.

axis_colorbarh=254 **(Windows only)**

Specifies the width, in pixels, of the color bar on color images and plots.

axis_colorbarv=24 **(Windows only)**

Specifies the height, in pixels, of the color bar on color images and plots.

axis_gridlines=true

When =true, Transform draws gridlines on the plot. Gridlines are drawn from each major tick to the opposite side of the plot. When =false, no gridlines are drawn.

axis_labelinterval=5

Number of intervals between labels on the axis. Labels are spaced to form **axis_labelinterval** intervals between **axis_labelmin** and **axis_labelmax**.

axis_labelmax=3.14

End of label range, in data or scale units. Labels are drawn starting at **axis_labelmin** and ending at **axis_labelmax**. They may be clipped by **axis_min** and **axis_max**. If **axis_labelmax** is less than **axis_labelmin** the labels decrease in value along the axis.

axis_labelmin=0.0

Start of label range, in data or scale units. Labels are drawn starting at **axis_labelmin** and ending at **axis_labelmax**. They may be clipped by **axis_min** and **axis_max**. If **axis_labelmax** is less than **axis_labelmin** the labels decrease in value along the axis.

axis_labelspacing=0.5

Distance between axis labels, in data units.

axis_majorlength=8

Length of a major tickmark, in pixels.

axis_majorticks=2

Number of major tickmarks per label interval. If **axis_majorticks=1**, then a major tick is placed at every axis label.

axis_majorwidth=1

Width of a major tickmark, in pixels.

axis_max=3.14

End of the axis in scale units. No tickmarks and labels appear beyond this value.

axis_min=0.0

Start of the axis in scale units. No tickmarks and labels appear before this value.

axis_minorlength=8

Length of a minor tickmark, in pixels.

axis_minorticks=2

Number of minor tickmark intervals per major interval. If **axis_minorticks**=5, then four minor ticks are visible between each two major tickmarks. If **axis_minorticks**=1, then the minor ticks are each obscured by a major tickmark.

axis_minorwidth=1

Width of a minor tickmark, in pixels.

axis_mirror=true

When =true, two sets of axes are drawn. The axes are drawn on both sides of the image when vertical axes are being drawn, or above and below the image when horizontal axes are being drawn.

axis_tickinout=2

Direction to draw both major and minor tickmarks:

=1	Inward ticks
=2	Outward ticks
=3	Both inward and outward ticks

axis_title="X axis"

Title to display on the axis.

color_lowoutlier=1 **(Windows only)**
color_highoutlier=254 **(Windows only)**

These variables specify the color index number (0-255) used for outliers. Use 0 for black; 255 for white.

color_max=254
color_min=1

Lowest and highest color index number (0-255). When you are creating color images, data is scaled to the colors between the color indices **color_min** to **color_max**. Usually, **color_min**=1 and **color_max**=254; 0 and 255 are normally reserved for black and white, respectively.

color_scalemax=3.14 **(Power Macintosh only)**
color_scalemin=0.0 **(Power Macintosh only)**

The highest and lowest data values; used for converting data values to colors. Usually overridden by **data_min** and **data_max**, except in special plots (surface plots with a separate color variable).

color_scaling=0 **(Windows only)**

Specifies method of scaling values to colors:

=0	Automatic selection
=1	Linear (appropriate for scientific data)
=2	No scaling (appropriate for image data)

color_table="Rainbow"

Name of the color table to use from the ColorTables menu.

color_variable="Density"

Name of the dataset to base colors on. When you are using a primary dataset for a surface plot (e.g., surface height) and a secondary dataset (e.g., surface color), indicates the name of the secondary dataset.

contour_autolevels=7

Number of automatic, equally spaced contour levels. Used when **contour_list="auto"** and in the Contours dialog.

contour_color=true

When =true, Transform draws contour lines in a color that depends on the data value for each line. When =false, all lines draw in black. Same as the color lines setting in the Contour Levels dialog.

contour_dashlines=true

When =true, Transform draws some contours as dashed lines. Lines representing data values less than **contour_dashsplit** are dashed, others are solid. When =false, all lines are solid. Same as the dashed lines setting in the Contour Levels dialog.

contour_dashsplit=0.0

If **contour_dashlines**=true, then contour lines below the value **contour_dashsplit** are drawn as dashed lines. Same as the dashed line value for Contours.

contour_highres=true

When =true, Transform draws contours in higher resolution. Same as the high resolution setting in the Contours dialog.

contour_list="0:100@10"

Explicit list of contours to draw. The list may contain a comma-separated list of numbers or contour ranges. When **contour_list**="auto", then Transform picks the contours, using **contour_autolevels** to set the number of contours. See the contour chapter for details on the format. Same as the contour list in the Contour Levels dialog.

contour_smoothline=true

When =true, Transform draws contours with Bezier-smoothed curves. Same as the smooth lines for Contours.

contour_textfont="Courier" **(Power Macintosh only)**

Font for contour labels.

contour_textformat="F8.2"

Printing format for contour labels.

contour_textsize=10 **(Power Macintosh only)**

Text size for contour labels, in points.

contour_textw32font="Courier New,NoBold,10" **(Windows only)**

Font for contour labels.

data_autominmax=true

When =true, Transform calculates the data minimum and maximum automatically, and overrides the **data_min** and **data_max** settings.

data_max=3.14
data_min=0.0

The highest and lowest data values; used for converting data values to colors.

data_xname="x axis" **(Windows only)**
data_yname="y axis" **(Windows only)**

Names of X and Y axes; identical to the rows and cols fields in the Attributes dialog.

file_directory **(Windows only)**

Name of the current directory for opening files. See the **setdirectory** subroutine for details.

file_savedirectory **(Windows only)**

Name of the current directory for saving files. See the **setsavedirectory** subroutine for details.

file_folder **(Power Macintosh only)**

Name of the current folder for opening files. See the **setfolder** subroutine for details.

file_savefolder **(Power Macintosh only)**

Name of the current folder for saving files. See the **setsavefolder** subroutine for details.

fill_equal=1

Parameter used for defining missing data values and also for defining a background fill value when converting column files to matrix files:

=1	if missing data values are equal to fill_value
=2	if missing data values are greater than fill_value
=3	if missing data values are less than fill_value

fill_function=2

Parameter designating data value range of influence. Used by the Kriging and weighted fill interpolation methods:

=1	for linear (1/r) distribution function
=2	for spherical (1/r^2) distribution function
=3	for exponential distribution function
=4	for Gaussian distribution function

fill_krignugget=0.0

The Kriging Nugget value. See the Kriging section for more information.

fill_krigcolrange=1

The Kriging column range of influence, in data units. See the Kriging section for more information. This reserved variable cannot be set to zero. If so, it will be changed to 1 before being executed.

fill_krigrowrange=1

The Kriging row range of influence, in data units. See the Kriging section for more information. This reserved variable cannot be set to zero. If so, it will be changed to 1 before being executed.

fill_krigsamples=50

Maximum number of samples per Kriging operation.

fill_krigsill=1.0

Kriging sill factor.

fill_krigvarmatrix=true

When =true, Transform produces a variance matrix in addition to the filled matrix after Kriging.

fill_linear=1

Parameter for linear interpolation. Same as the linear interpolation option in the Fill Missing Data dialog.

=1	for interpolate along rows and columns
=2	for interpolate along rows only
=3	for interpolate along columns only

fill_method=1

Parameter for selecting fill missing data method. Matches the selections from the Fill Missing Data dialog. Used for the **fillmissing()** notebook function.

=0	Do not interpolate missing data
=1	for interpolation using nearest neighbor
=2	for interpolation using Kernel smoothing
=3	for interpolation using linear interpolation
=4	for interpolation using weighted fill
=5	for interpolation using Kriging

fill_preserve=false

When =true, Transform preserves valid data values while filling missing data. To get smoother output, set **fill_preserve=false**. Only used with weighted fills.

fill_range=4

Size of the region of influence for filling missing data, in row and column units. If **fill_range**=0, a default value is calculated and used. When using the Kriging method, the variables **fill_krigcolrange** and **fill_krigrowrange** are used instead.

fill_value=-99

Missing data value. Along with **fill_equal**, specifies the range of missing data values (and therefore the range of valid data values).

histogram_color=true **(Windows only)**

When =true, a histogram appears in color; when =false histogram appears in black-and-white.

image_axes=true

When =true, Transform adds axes to a plot or image; when =false axes are not included.

image_canvash=640
image_canvasv=480

Size of the window's background canvas in pixels. The default value is the size of the main display. Also found in the Preferences dialog.

image_frame=true

When =true, Transform draws a frame around the plot.

image_v=300
image_h=400

The image size, in pixels. Same as the Image Size dialog.

image_marginleft=0
image_margintop=0

The position of the upper left corner of the image, relative to the upper left corner of the canvas, in pixels.

image_title="Plot title"

Title for the plot. This title appears centered, above the plot.

import_3d=true

When =true, the file being imported is flagged as a 3D file. When =false, the file is flagged as a 2D file.

import_3daxis=1

Parameter used when importing 3D files to specify axis direction. Ignored unless **import_3d**=true.

=1	for importing slice in XY plane
=2	for slice in XZ plane
=3	for slice in YZ plane

import_3dslices="20,22,23"

List of 2D slices to import from a 3D file. Ignored unless **import_3d**=true. See the section on the Select Slices dialog for information on the format.

import_byteswap=true

When =true, Transform swaps bytes when reading data. See the data import section for more information.

import_colautoscale=true

When =true, Transform calculates the column minimum and maximum values automatically, overriding the parameters **import_colxmin, import_colxmax, import_colymin, import_colymax, data_min and data_min**. When =false, those same parameters are used to calculate the column x, y, and data minimums and maximums.

import_coldatacols="3,4,5"

Indicates which columns to import as data columns. See the Data Import section for more information.

import_coldelim=false

When =true, Transform treats every delimiter as a new column. Same as choosing Strict Delimiters during import. When =false, you can have multiple delimiters (spaces, commas, tabs) between columns.

import_colfixedchar="1:4,5:10,11:15"

Specifies the character positions of the columns when import_delimiter=7 (fixed character positions). Each column is specified by a starting character and ending character position, separated by a colon. Same as choosing Fixed Fields during import.

import_colplacement=3

If two import data values will be placed in the same location in the target matrix, this parameter specifies how to handle the situation:

=1	for replacement of the previous value at the matrix location
=2	for summing all values at the same matrix location
=3	for averaging all values at the same matrix location

import_coltargetx=61
import_coltargety=41

Size of the target matrix when converting column data to matrix data. The parameter **import_coltargetx** is the number of columns in the target matrix, **import_coltargety** is the number of rows.

import_coltitles=true

When =true, the last line of the header contains the column names (for text column import only).

import_colx=1

Specifies the column number that contains the X locations.

import_colxmin=0.0
import_colxmax=3.14

Specifies minimum and maximum values in the column used for the X locations. Data values with X locations outside this range are ignored. Points inside this range are positioned in the target matrix between **import_colxmin** on the left and **import_colxmax** on the right. If **import_colautoscale**=true, these parameters are ignored.

import_coly=1

Specifies the column number that contains the Y locations.

import_colymin=0.0
import_colymax=3.14

Specifies minimum and maximum values in the column used for the Y locations. Data values with Y locations outside this range are ignored. Points inside this range are positioned in the target matrix between **import_colymin** on the top and **import_colymax** on the bottom. If **import_colautoscale**=true, these parameters are ignored.

import_delimiter=3

Specifies how text columns are delimited:

=0 for columns separated by tabs only

=1 for columns separated by spaces only

=2 for columns separated by commas only

=3 for columns separated by tabs or spaces

=4 for columns separated by tabs or commas

=5 for columns separated by spaces or commas

=6 for columns separated by tabs, spaces, or commas

=7 for columns in fixed locations specified by import_colfixedchar

import_dimcolumns=100
import_dimrows=80
import_dimlayers=70

Specifies the size of the import 2D or 3D matrix or column file. The parameter **import_dimlayers** is ignored unless **import_3d=true**.

import_filetype=1

This macro is only necessary for **user_interactive = false**. Parameter specifying the type of import file:

=0	for file type not specified (the default case)
=1	for HDF files
=2	for HDF files saved with Transform
=3	for HDF VSET files
=4	**(Power Macintosh)** for PICT files
=5	**(Power Macintosh)** for PICS files
=6	for TIFF files
=7	for FITS files
=8	for binary matrix files
=9	for binary column files
=10	for binary PBM files
=12	for MATLAB files
=13	**(Power Macintosh)** for XWD files
=14	for GIF files
=15	for ASCII Text matrix files
=16	for ASCII Special files
=17	for ASCII Text matrix files with scale information
=18	for ASCII column files
=19	**(Windows)** for ASCII PBM files
=20	**(Windows)** for Bitmap files

import_flipcols=false

When =true, Transform flips the columns left for right when importing files. Same as using the **colflip()** function on your imported dataset.

import_fliprows=false

When =true, Transform flips the rows up for down when importing files. Same as using the **rowflip()** function on your imported dataset.

import_transpose=false

When =true, Transform flips the data array diagonally when importing files. Same as using the **transpose()** function on your imported dataset.

import_numtype=1

Parameter specifying the number type in a binary file:

=1	for signed 8 bit byte
=2	for unsigned 8 bit byte
=3	for signed 16 bit integer
=4	for unsigned 16 bit integer
=5	for signed 32 bit integer
=6	for unsigned 32 bit integer
=7	for IEEE 32 bit floating point
=8	for IEEE 64 bit floating point
=9	for VAX/VMS 32 bit floating point

import_record="5,6,7,10"

Specifies which records to read from a multi-record file.

import_skip=0

For text files, specifies the number of lines of text to skip before reading data. For binary files, specifies the number of bytes to skip before reading data.

lineplot_lock=false

When =true, Transform will not update the lineplot when the selection region changes. When =false, the lineplot will be updated when the selection region changes.

lineplot_number=4

Specifies which row or column index number to plot on a lineplot.

lineplot_vertical=false

When =true, the lineplot is drawn from a column of the 2D data array. When =false, the lineplot is drawn from a row of the 2D data array.

line_dashlength=5

Length of dashes in a dashed line, in pixels. Only used when drawing dashed lines.

line_width=1

Width of drawn lines, in pixels. Used for line graphs, contours, axes, and the plot frame. Note that you can set **line_width** to fractional values such as 0.4, to produce very fine lines when printing to high-resolution output devices.

polar_radius=150 **(Power Macintosh only)**

Size of polar plot contents, from center to edge, in pixels. The largest radius value (row scale) is scaled to this size when plotted on the screen. The parameters **image_v** and **image_h** control the overall size of the plot, but the contents are drawn to the scale specified in **polar_radius**.

polar_view="200,-200,200,200" **(Power Macintosh only)**

Defines the viewing area of a polar plot in pixels, where 0,0 is the center of the polar coordinates.

polar_zeroangle=0 **(Power Macintosh only)**

Parameter specifying the coordinate system for polar plots.
=0 for east, =1 for south, =2 for west; =3 for north being the direction of the angle 0.0.

pref_aspectratio=false

When =true, images are initially generated with the same aspect ratio as the size of the dataset array. When =false, images are initially generated with the size **image_h** by **image_v**.

pref_maxnumber=INF
pref_minnumber= -INF

Specifies the largest and smallest legal numbers. All values outside this range are assumed invalid, so they are left out of max/min calculations, etc. Note: **INF** represents infinity.

pref_textformatdata="F8.1"

Specifies the printing format used for the data values in all new data windows. Also found in the Preferences dialog.

pref_textformatscales="F5.1"

Specifies the printing format used for the scales in all new data windows. Also found in the Preferences dialog.

pref_textfont="Courier" **(Power Macintosh only)**

Specifies the default font for all new windows; also found in the Preferences dialog.

pref_textsize=10 **(Power Macintosh only)**

Specifies the text size, in points, for all new dataset, image and Notebook windows. Also found in the Preferences dialog.

pref_textw32font="Courier New,NoBold,10" **(Windows only)**

The preferred default font for all new windows; may also be set using the **Font...** command under the Preferences submenu of the Edit menu.

pref_windowh=200

Width of new windows.

pref_windowv=200

Height of new windows.

print_cropmarks=true **(Windows only)**

When true, Transform places cropmarks on multi-page printouts for alignment assistance.

print_marginleft=1 **(Windows only)**
print_marginright=1 **(Windows only)**
print_margintop=1 **(Windows only)**
print_marginbottom=1 **(Windows only)**

Specify page margins, in inches, as measured from border of printable area of page.

print_pageheaderleft=" " **(Windows only)**
print_pageheaderright="&n" **(Windows only)**
print_pagefooterleft="&d" **(Windows only)**
print_pagefooterright="Page &p of &P" **(Windows only)**

Parameter specifying text page headers and page footers to be printed. The following special symbols will be expanded as each page is printed:

&d	date MMM DD YYYY
&D	date DD MMM YYY
&n or &N	dataset name
&p	page number
&P	total number of pages
&t or &T	the total number of pages
&w or &W	name of the window
&&	&

resample_scalecol=0

When =0, columns are resampled to a linear scale. When =1, columns are resampled on a logarithmic scale.

resample_scalerow=0

When =0, rows are resampled to a linear scale. When =1, rows are resampled on a logarithmic scale.

resample_type=0

Parameter specifying the resampling method when resampling to a larger size. Parameter is ignored when resampling to a smaller size.

=0 for bilinear interpolation

=1 for biquadratic interpolation

=2 for bicubic interpolation

surface_anglex=-50.0
surface_angley=0.0
surface_anglez=45.0

Rotation angles, in degrees, specifying the orientation of the surface plot. When both **surface_anglex=0** and **surface_angley=0**, then the surface plot Z axis (the axis perpendicular to the plane of the surface plot) points toward you out of the screen. A positive **surface_anglex** tilts the Z axis downward, a positive **surface_angley** tilts the Z axis to the right. The parameter **surface_anglez** specifies the rotation of the surface plot around the Z axis.

surface_averagecolors=true

When =true, Transform averages the four corner data points to calculate the color used on color surface plots. When =false, the upper left corner data point is used to calculate the color.

surface_axistype=4

Parameter specifying surface plot axis configuration.

=1 for surface plot only, no axes

=2 for row, column and data axes originate at origin

=3 for row, column axes at front, data axes on left

=4 for row, column axes at front, data axes on right

=5 for row, column axes at front, data axes on left and right

surface_backstop=255

Parameter specifying the color of the surface backstop:

= -1	for no backstop at all
= 0	for a white backstop
= 255	for a black backstop
= 1 to 254	for the color index from current color table
= 256	for gray

surface_backstopline=false

When =true, Transform draws lines from each data axis major tick mark along surface backstop walls. Ignored if **surface_backstop=-1**.

surface_linemaxcol=25
surface_linemaxrow=25

Parameters specifying the maximum number of lines to draw for the row and column axes on the surface plot. The lines are always spaced an integer skip between data rows and columns, so the actual number of lines drawn is often less than the maximum.

surface_linescol=true
surface_linesrow=true

When =true, Transform draws lines along the columns and/or rows. At least one of the two must be =true. If both are =true, you will see a 'mesh' surface plot.

surface_offseth=50
surface_offsetv=50

Specifies the horizontal and vertical offset of the center of the surface plot from upper left corner of the window, in pixels.

surface_parallax=5.0

Parameter specifying the surface plot parallax.

surface_sizex=110
surface_sizey=110
surface_sizez=85

Specifies the size of the X, Y, and Z axes of a surface plot, in pixel units. The actual lengths of the axes are smaller due to the 3D projection.

surface_skirt=0

Parameter specifying the color of the surface skirt:

= -1	for no skirt at all
= 0	for a white skirt
= 255	for a black skirt
= 1 to 254	for the color index from the current color table
= 256	for gray

surface_type=1

Parameter specifying the type of surface plot. Same as the surface option menu in the surface plot selection tool.

=1	for a wireframe surface plot (no hidden line removal)
=2	for a black&white surface plot (hidden line removal)
=3	for a high resolution black&white surface plot
=4	for a color surface plot (no grid lines)
=5	for a framed color surface plot (grid lines)
=6	for a high resolution framed color surface plot

text_format="F8.1"

Specifies the number printing format. Used for numbers printed as axis labels.

text_font="Courier" **(Power Macintosh only)**

Specifies the font name used for displaying text. Used when producing images to specify the font for labels and titles.

text_size=12 **(Power Macintosh only)**

Specifies the font size, in points. Used when producing images to specify the text size for labels and titles.

text_w32font="Courier New,NoBold,10" **(Windows only)**

Default font used for all windows created when **user_interactive=false**, such as during the execution of a macro.

user_interactive=true

When =true, dialogs are allowed. For example, executing a **call contour('Xvel')** when **user_interactive=true** will cause the contour dialog to appear. When =false, no dialogs are displayed, and the keywords defined here are used to create the plots. The parameter **user_interactive** is usually set to false for complex macros. When using calculate from notes, note that **user_interactive** is set to true before the macro commands execute. In fact, each time you execute a menu command or use a command key combination, Transform sets **user_interactive=true.**

vector_arrowtype=1

Parameter between 1 to 6 specifying the type of arrow for vector plots. The following represents Windows parameters:

=1	for scaled arrow with black head
=2	for scaled arrow with white head
=3	for fixed black head with scaled tail
=4	for fixed circle with scaled tail
=5	for scaled wedge
=6	for plain scaled arrow

vector_max=3.14
vector_min=0.0

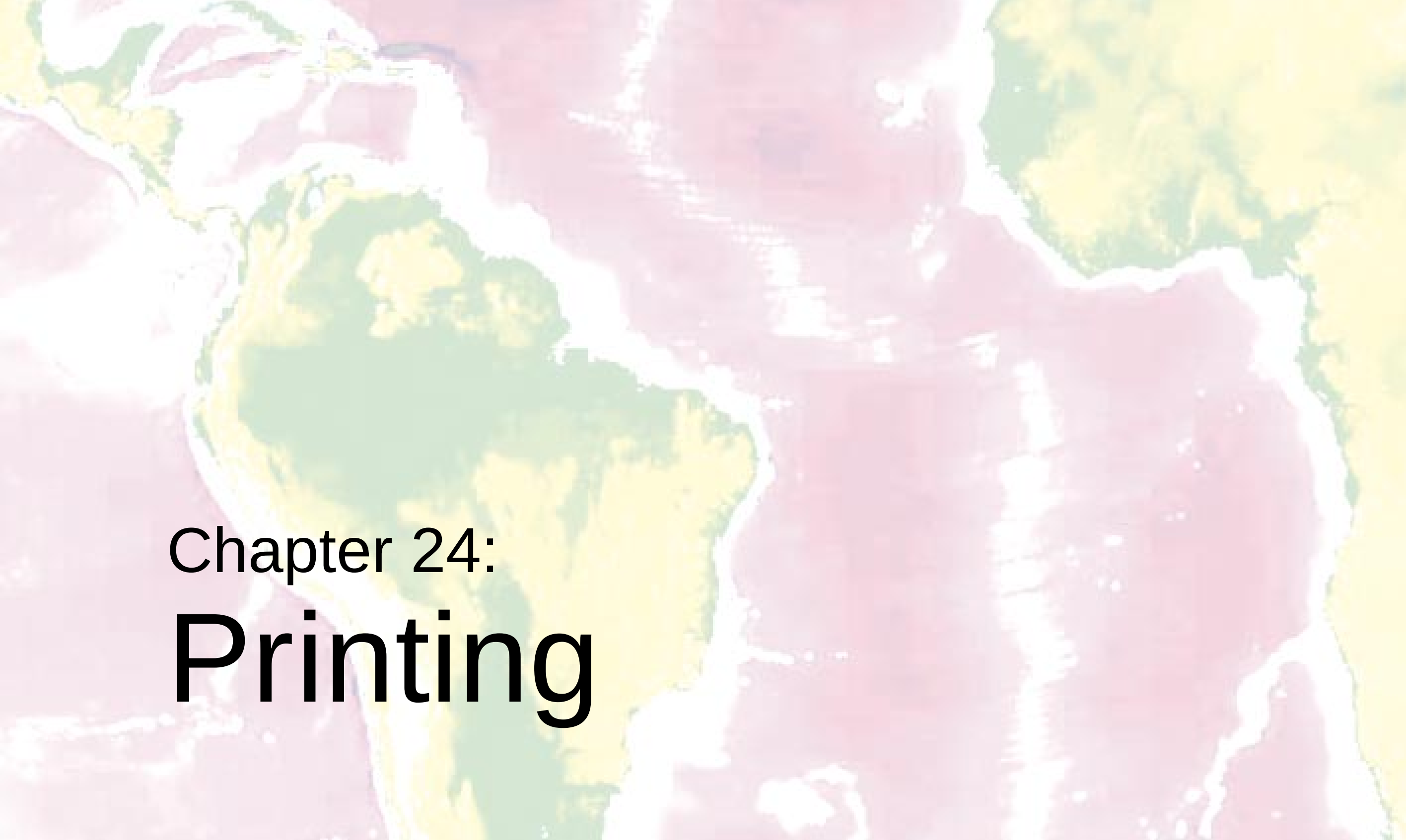
The minimum and maximum vector magnitudes in a vector plot, in data units. Vectors with a magnitude less than vector_min or greater than **vector_max** are omitted from the vector plot. A vector with a magnitude of **vector_max** will be exactly **image_h/vector_number** pixels long, if **vector_scale=1.0**.

vector_number=22

Parameter specifying the number of arrows across for a vector plot. The number of arrows vertically is calculated automatically so that the horizontal and vertical arrow spacing is identical.

vector_scale=1.0

Parameter specifying the maximum length of vectors. The maximum vector length is **vector_scale*(image_h/vector_number)** pixels long.



Chapter 24:

Printing

Transform provides extensive support for printing dataset windows, Notebook windows, and the image and plot graphic windows. Image windows are printed as they are displayed; the entire contents of a window, from image to annotation to overlaid plots, are printed just as they are displayed on the screen.

Printing in Transform for Windows

Before printing, you must select and configure a printer from the list of printers installed on your Windows system.

To do so, select the **Printer Setup...** command from the File menu. You will see a dialog similar to the one shown below.

Print Setup Dialog

Figure 24-1: Print Dialog

Printer

The Printer field lists the currently selected printer, its status, type and location. Open the drop-down list to choose from other printers configured on your system. If this list is empty or does not list a printer you wish to use, install or configure the printer using the Printers program in the Windows Control Panel; please refer to the your Windows documentation for additional information. You may also select the range of pages you would like to print and the number of copies to be made.

To change printers or paper attributes, press the **Properties** button. You will see a Print Properties dialog similar to the one shown below.

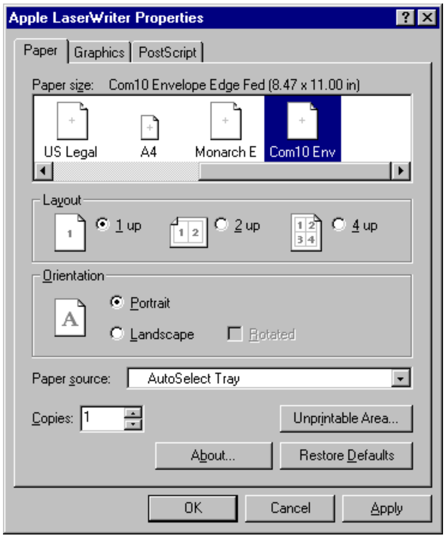


Figure 24-2: Print Properties Dialog

This dialog is tailored to the currently selected printer and to your version of Windows. Therefore, its content may vary considerably from that shown above.

Paper

Use the settings in this area to select the paper size on which you want to print and set the paper source.

Orientation

Lets you specify whether or not you want the pages to be printed portrait or landscape oriented.

Page Setup

When you print a Transform window, it will be paginated if necessary. The Page Setup dialog, available from the **Page Setup...** command on the File menu, lets you select page margins and optional header and footer text.

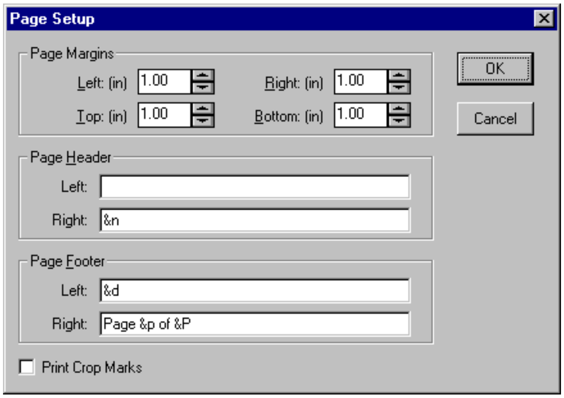


Figure 24-3: Page Setup dialog

Page Margins

The four page-margin fields give you complete control over the size and placement of text and graphics on the printed page. These values are specified in inches, and are relative to the printable area of the page as reported by the printer driver. Therefore, if you have exact page placement requirements, you will need to decrease your margins to compensate for the unprintable region.

Page Headers and Footers

Transform also allows you to specify header and footer text to appear at the top and bottom of each page of your output. Vertically, headers and footers are placed just outside the margin; they are suppressed if the corresponding vertical margin is less than the height of the selected font. Horizontally, they are placed just inside the margin and are justified to the outside.

The header and footer text may be set to any text or phrase of your choosing, or may be empty. Transform allows you to specify the following special tokens that will be replaced as each page is printed:

Token...	Replacement ...
&d	date MMM DD YYYY
&D	date DD MMM YYY
&n or &N	dataset name
&p	page number
&P	total number of pages
&t or &T	page number
&w or &W	total number of pages
&&	the total number of pages

Use of any other character following an & is reserved for future use.

Print Preview Command

Selecting the **Print Preview** command from the File menu shows the data, image or attribute window at a reduced size so you can adjust the layout before you print. Tools on the preview window let you print, view subsequent or previous pages, view two pages side by side, and zoom in or out of the window.

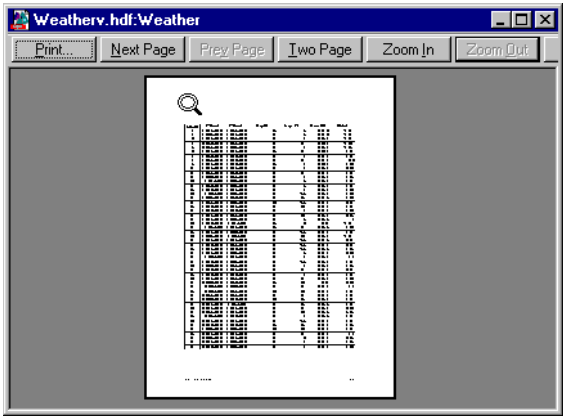


Figure 24-4: Print Preview Window

Print Command

Selecting the **Print** command from the File menu will open the Print dialog shown below.

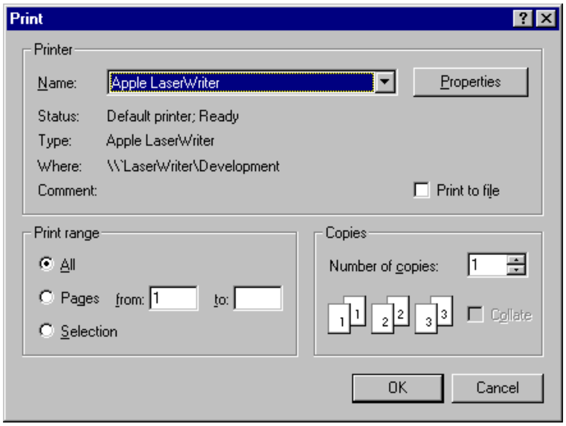


Figure 24-5: Print Dialog

Printer

Same as described for the Print Setup dialog above, with the addition of a Print to File option.

Print Range

The buttons under Print Range let you choose to print all pages, to limit output to a range of pages. Other options in the Print dialog will vary by printer.

The first time that you invoke the Print dialog during a Transform session, the default settings for the default system printer will appear.

Copies

Allows you to specify the number of multiple copies of the document you would like to print.

Selecting **OK** in the Print dialog prints the contents of the frontmost window. The details of printing the various window types (data, text window, etc.) are given in the next section. While the window is being printed, the following dialog will be visible and will show progress on a page-by-page basis.

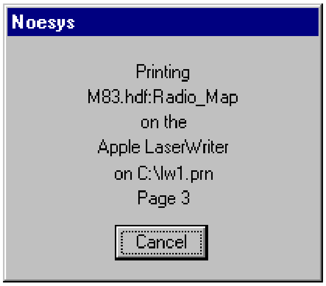


Figure 24-6: Print Progress Dialog

Interrupting or Aborting a Print

To abort the printing process, press the **Cancel** button on this dialog. It may take a moment for the abort signal to terminate processing, since it is checked at the beginning and ending of each page.

(If the output was queued to Windows Print Manager (the default), it may be aborted or cancelled; see the Print Manager documentation that came with your Windows system.)

Printing in Transform for Macintosh

This chapter provides an overview of printing from Transform to LaserWriters and EPS files. Although printing to non-Laserwriters is usually very similar, you may need to refer to your printer manual to take full advantage of all of the options available. It is best to print color images on a color printer, but black and white printers such as the Laserwriter can produce reasonable grayscale printouts of your images.

Page Setup

To setup your printer, select **Page Setup...** command from the File menu. If you are using a Laserwriter, you will see the dialog shown in the figure below. If you are not using a Laserwriter, your dialog may be different.

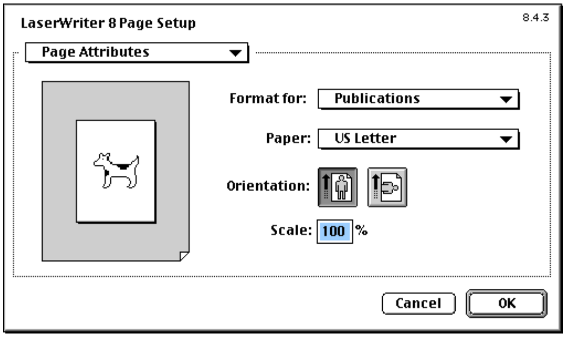


Figure 24-7: Page Setup...Dialog

Use this dialog to change the orientation of the paper or set an expansion factor. The percent reduction field may be used to reduce (or enlarge) the entire plot by a percentage factor where 100% is the same size plot as you see on your screen.

The Print... Command

Selecting **Print...** from the File menu prints the contents of the front-most window. Live images are printed actual size in the same position relative to the paper as they are placed relative to the canvas.

If you are using Apple's Laserwriter drivers, you will see the dialog in the figure below. The version number printed in the upper right corner of the dialog must be 6.0 or greater for grayscale printing. If you are not using a Laserwriter, your dialog may be different.

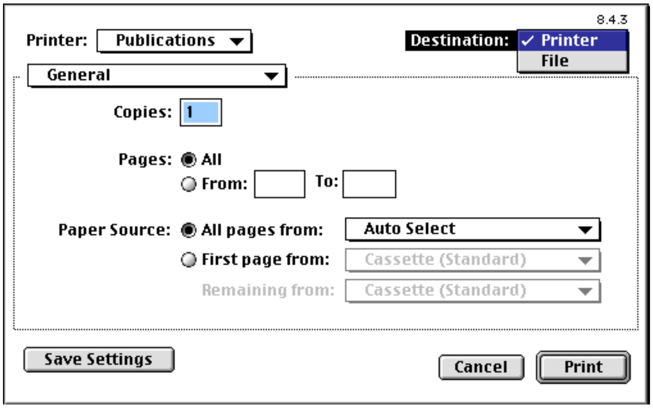


Figure 24-8: Print Dialog

To print to an EPS file, select **File** from the Destination pop-up menu. This will open an additional dialog that lets you specify the format.

Interrupting a Print

Command-period halts the printing process at the end of the page in process. One page of graphics takes a long time to print, so the interrupt request may not take effect immediately.

Selected Regions

The **Print...** command prints the entire window contents and ignores the selection region. To print only the contents of the selection region, first use **Extract Selection** from the Numbers menu to create a new data window. Generate a new image window if necessary, then print from the new window.

Choosing the Printing Font

The font for printing matches the font and size used in the window. For data and notebook windows, the font and size are specified in the Preferences dialog. For contour and axis labels, the font and size are specified from menus in the Labels and Axes tools.

Positioning the Image on the Page

The image canvas corresponds to the paper the image is printed on. New images always start out in the upper left corner of the canvas, so on paper they print in the upper left corner of the page. Use the Size tool and Overlay tool to position the image and overlay on the canvas, or use the Printing.txt macro from the Advanced Macros section of Chapter 22.

If your monitor has 72 dots per inch (typical for Macintosh monitors), your printed image will be exactly the same size as the screen image (unless you have changed the percentage enlargement in the Page Setup dialog). Transform always assumes that every pixel is 1/72 of an inch in size.

Setting Line Width

All Transform plots support the line_width setting to refine line thickness on print-outs. Before you create your plot, or in the Preferences macro, set the reserved variable line_width to the desired line thickness, in fractions of a point. We have seen good results setting the thickness to 0.4 points.

Printing Windows

Printing from dataset, image, and notebook windows is described below.

Printing Dataset Windows

The number of rows and columns of text that will fit on a page is determined by the parameters set using the **Print Setup...** command. Transform prints the contents of the window in spreadsheet-like fashion, left-to-right and top-to-bottom. The scale values are printed to the left of each row and above each column.

When multiple pages are required in Windows, the scale values are repeated on each page. To aid in viewing, a separator line is drawn between the scale values and the dataset values. Separator lines are also drawn between every five rows and columns. If the print range was set to Selection in the Windows Print dialog, Transform will only print the rows and columns in the selection region.

In Power Macintosh, pages are printed with the name of the dataset, page number, and a borderline along the top and left of the data array. Independent scale values are printed on each page so that multiple pages may be aligned. In addition, Transform always prints all the numbers in a data window, not just the visible window or the selection region. This can easily add up to many pages, so double-check the size of your dataset before printing.

Note

It is strongly recommended that you select a fixed width font, such as ‘Courier’ or ‘Courier New’, so that decimal points line up properly.

Printing Notebook Windows

The entire contents of the notebook are printed by default. If, however, you enabled Selection under Print Range in the Printer Setup dialog, only the highlighted region of the Notebook will be printed.

In Power Macintosh, the Notebook prints with line breaks set up to match the screen appearance. The contents of the Notebook are paginated and the pages are numbered. The font and size match the font and size set up for the window on the screen.

Printing Image Windows

The size and placement of most image window types—raster, interpolated, line, contour, vector, and histogram—is determined by the image size and margin settings on the Resize tool. For an overlay it is determined by the settings on the Overlay tool.

On the screen, the size and margin refer to screen pixels. On paper, these are interpreted as **points** (1/72 inch). For example, an image of size 216 x 216 will appear as exactly 3 inches square on paper. The image margins specify an offset from the page margins for the upper-left corner of the image area. The vertical axis and any other annotations are drawn in the margin area.

Printing Surface Plot Windows

Surface plots (except for wireframe) are drawn as a map of filled polygons. Drawing always takes place back-to-front so that the hidden lines in the rear are obscured. For best results, it is recommended that you select Hi-res Black & White or Hi-res Color Surface before printing your surface plots.

Surface plots are placed on the page so that the upper-left corner of the window aligns with the top-left margin of the page.

In Power Macintosh, in the lower left, the dataset name is printed along with the names of the independent variables, like the raster plots.

Since polygon filling takes a long time at high resolution, surface plots are usually easier to manage as bitmap representations rather than as series of overlapping polygons. In the Power Macintosh Print dialog, Transform provides a Print Bitmap checkbox for your selection. Turn the checkbox on or off before clicking **OK**. Printouts of the bitmap can be 4 to 10 times faster than the high resolution filled polygons, but are not as detailed.

Printing Vector Plots

Vectors on a vector plot are drawn as line segments, sometimes using a polygon or filled polygon for the arrow head. Vectors are calculated and drawn at the highest printer resolution available.

Printing with Overlays

After the base image is printed, any overlay present for that image is printed. The overlay is positioned on the page with the same offset and same relative size as was present in the image window.

Because PICT drawings in Power Macintosh are fixed at 72 DPI (dots per inch) the base resolution for printing overlays is always 72 DPI. However, you can get higher resolution for your overlay by bringing it in at double size (or more) and then reducing it in size with the Overlay tool.

Printing Line Graphs

Line graphs are drawn as line segments.

Print to File

Depending on your available print drivers, you can print data, image and attribute windows to a printer (*.prn) file by selecting the **Print to File** checkbox (Windows) or to a postscript file with the **File** command from the Destination pop-up menu (Macintosh) in the Print dialog. Clicking **OK** with this option selected will open the Print to File dialog.

Part IV: Appendices

Appendix A:

Transform Menus

This appendix reviews each of the menus and commands that appear in Transform's menu bar. The menu bar changes depending upon whether a file is open, and which window type is active. If the Transform program window is open, but no files are open, Transform commands appear in four menus: File, Edit, Macros, and Help. When a file is open, the following menus appear: File, Edit, Image, Numbers, Macros, Color Tables, Tools (Windows only), Windows, and Help.

Note that certain Transform windows have fewer menus. For example, the Color Tables menu is absent when a line plot is active.

File Menu

Figure A-1: File menu

The **New...** command lets you to create a new array with all of the numbers set to 0.0 or 1.0.

The **Open...** command accesses a file from disk. Use it to load datasets which were previously saved from Transform or to import data from a variety of file formats.

The **Save** command saves any updates to an open file.

The **Save As...** command saves a copy of the current dataset with the file name of your choice. Several export types are available.

The **Close** command closes the current window and removes it from the screen. When you close a dataset window, the data is removed from memory, and all dependent windows are also removed.

The **Close All** command removes all windows from the screen.

The **Print** command prints the current dataset, notebook or image in a manner consistent with parameters set using **Page Setup...** and **Printer Setup...**.

The **Page Setup...** command brings up a dialog that allows you to set page margins, and add header and footer text.

The **Printer Setup...** command (Windows only) brings up a dialog that allows you to select and configure a printer from the list of printers installed on your Windows system.

The **Exit/Quit** command closes all of the windows and exits the program.

Edit Menu

Figure A-2: Edit Menu

The **Cut** command removes the currently selected text (if appropriate) and copies it to the clipboard.

The **Copy** command copies a Windows metafile representation of the current plot, or copies currently selected data as text, to the clipboard.

The **Paste** command adds the contents of the current clipboard to the current window wherever appropriate.

The **Clear** command removes the currently selected text, image, or overlay from the currently selected window, whichever applies.

The **Copy As...** command (Windows only) brings up a dialog that lets you select which components in a window to copy. When appropriate, this dialog also allows you to choose what format to copy to.

The **Paste As...** command (Windows only) brings up a dialog which lets you choose what to paste from the clipboard. When appropriate, this dialog also lets you choose a format for the pasted component.

The **Paste Overlay** command (Power Macintosh only) adds the contents of the current Clipboard to an image or plot window as an overlay.

The **Copy 2x** and **Copy 4x** commands (Power Macintosh only) enlarge images (at two to four times their actual size) while copying them to the Clipboard. This feature is useful for resolution improvement.

The **Copy Bitmap** command (Power Macintosh only) copies a bitmap representation of the current plot to the Clipboard.

The **Paste New** command (Power Macintosh only) use the current contents of the Clipboard to create a new window (if possible) and a new dataset from the image or data present.

The **Synchronize** command links two or more datasets of the same size, so that when data points are selected in one dataset, the corresponding data points will be selected in the other dataset(s).

The **Font...** command brings up the dialog that allows you to select font and size for text in dataset and notebook windows, and for axes labels in image and plot windows.

The **Preferences** command allows you to specify display font, style, and size; specify new window size, as well as data and scale numerical format; and view all current settings, line-by-line in macro form. For more on Preferences, see Chapter 20. You can save changes in settings to disk.

The **Status Bar** command (Windows only) lets you display or hide the status bar. When enabled, the Status Bar is displayed in the bottom left margin of the window, and a checkmark appears in the menu next to the command.

The **Tool Bar** command (Windows only) lets you display or hide the tool bar. When enabled, the tool bar is displayed across the top margin of the Transform window, and a checkmark appears in the menu next to the command.

The **Tool Palette** command (Windows only) brings up a pull-right that provides access to four options for the display of the tool palette. Selecting **Off** hides the tool palette. Selecting **Horizontal** from the submenu displays the tool palette across the top of the Transform window. Selecting **Vertical** from the submenu displays the tool palette along the left margin of the Transform window. Selecting **Floating** displays the tool palette in a floating window that you can move within the Transform window as desired.

Image Menu

The Image menu allows you to generate the different types of Transform images and plots. See Chapters 12-19 for more information about each image and plot type.

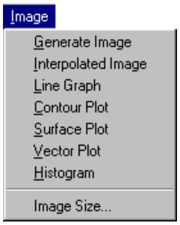


Figure A-3: Image Menu

The **Image Size.../Rectangle Size...** command brings up a dialog to control sizing of images and plots.

The **Circle Size...** command (Power Macintosh only) opens a dialog to control sizing information for any polar images created.

Numbers Menu

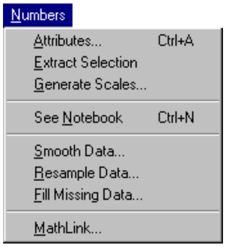


Figure A-4: Numbers Menu

The **Attributes...** command lets you select different characteristics for the current dataset, including the dataset name.

The **Extract Selection** command creates a new dataset that consists of the data values in the selection region of the current window.

The **Change Data Entry...** command (Power Macintosh only) lets you change the selected data value in your array.

The **Generate Scales...** command creates a new dataset from the current dataset with different row and column scales.

The **See Notebook** command opens a notebook window where you may enter notes about the dataset, or enter and execute macro expressions. In Windows, the **Calculate From Notes** command replaces **See Notebook** when the notebook window is active.

Calculate from Notes evaluates the current line or currently selected macro expression in the notebook.

The **Smooth Data...** command creates a new dataset by smoothing the values in the current dataset.

The **Resample Data...** command creates a new dataset of a different size from the current dataset by interpolating data values.

The **Fill Missing Data...** command provides several options for interpolating or replacing missing data in the current dataset.

The **MathLink...** command allows Transform to connect to Mathematica to accept commands and exchange data.

Macros Menu

See Chapter 22: Using Macros and Chapter 23: Macro Reference for more information on macro commands.

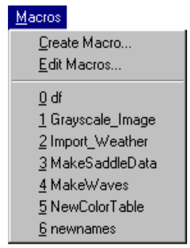


Figure A-5: Macros menu

The **Create Macro...** command prompts for a macro name and saves a macro based on the characteristics of the current window.

The **Edit Macros...** command presents a list of the currently saved macros and allows you to edit, rename, delete, import, export and create new macros.

Previously saved macros are listed below the below the horizontal divider line. Listed macros are executed when selected from the menu.

Color Tables Menu

The Color Tables menu lists color tables that can be applied to images and plots. See Chapter 13:Color and Color Tables for more information on color in Transform.

Figure A-6: Color Tables Menu

The **Custom Color Tables...** command (Windows only) brings up a dialog that lets you select a color table created outside of Transform, or that was included in a dataset you loaded. The name of the most recently-used custom color table appears between the dividers immediately below the **Custom Color Tables...** command.

The **Load Color Table...** command (Power Macintosh only) reads a color table from a disk file and applies it to the current image.

Tools Menu

The Tools menu provides an alternative to the tool palette and the tool bar. The top portion of this menu lets you select a tool for the current window, just as you would using the tool palette. The portion of this menu below the divider line changes with the current window type and tool. Selecting commands in this area is equivalent to pressing the corresponding toolbar button. For more about these tools and options, see Chapter 12.

Figure A-7: Tools Menu

Windows Menu

The Windows menu provides a way to manage open windows.

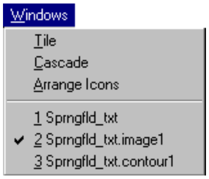


Figure A-8: Windows Menu

The **Show/Hide Window** command (Power Macintosh only) lets you hide and redisplay windows.

The **Tile** command arranges all open windows to fill the Transform window.

The **Cascade** command (Windows only) stacks open windows in a staggered fashion. Only the top window is fully visible, but the top and left edges of every window is visible beneath the previous window.

Arrange Icons (Windows only) aligns all iconified windows in a row in the lower left corner of the Transform window.

Below the divider, this menu also lists all windows that are currently open, including those that have been iconified in Windows.

Help Menu

In Power Macintosh, the Help menu provides access to Balloon Help.

In Windows, the Help menu provides access to Transform's on-line help.

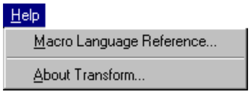


Figure A-9: Help menu

The **Macro Language Reference...** starts the on-line reference to the macro language.

The **About Transform...** command gives information on Transform, including the version number and copyright information. It also provides access to the technical support dialog.

Appendix B:

MathLink To Mathematica

MathLink can be used to connect Transform 3.4 and Mathematica 3.0, allowing Mathematica full scripting control over Transform. With third-party network software, you can even use MathLink to connect Transform to another computer that is running Mathematica.

Set Up A Mathlink Connection in Transform for Windows

Setting Up in Mathematica

In the 'Mathlink' directory is the file 'trnsfrm3.m'. Use the **Needs** or **<< (Get)** command to load this package:

```
In[1]:=   <<trnsfrm3.m
```

Now you have access to the list of Transform-specific MathLink calls. To set up a connection open a link with the following function:

```
In[2] := TransformConnect["3000"]
```

The parameter “3000” is a port name appropriate for your data transport protocol.

Note

See Mathematica and MathLink documentation for information on the various data-transport protocols available, and any port-naming restrictions. The TransformConnect[ ] function uses the default MathLink data transport protocol. To force a specific, protocol, use the alternative form, for example: **TransformConnect["3000", "TCP"]**.

When using the TCP protocol, the parameter is the socket number: a TCP/IP-specific number from 1024 to 32000. Choose any number you wish, but it must match on both sides of the link. If an error indicates the socket number is already in use, choose another number. Higher numbers are usually available.

This command does not return with any messages until Transform connects to this link, or it times out (several minutes). Start Transform and open the link from the other side (described below). When Transform connects it returns the symbol **$TransformReady**, so in Mathematica, you should see the following message along with a new prompt:

```
Out[2]= $TransformReady
In[3]:=
```

Now the link is open and you are ready to use Mathematica commands with Transform. Please refer to your MathLink manual to get a better idea of how connections are established, and of the connection options available.

Setting Up in Transform

To set up MathLink, select **MathLink...** from the Numbers menu to bring up the dialog shown in Figure B-1.

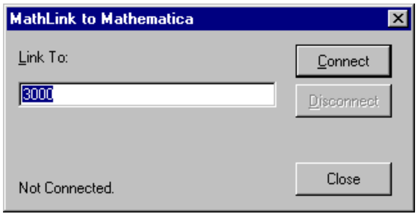


Figure B-1: MathLink Dialog

To connect to a Mathematica application running on your local workstation, enter the port name that you gave to Mathematica in the **TransformConnect[ ]** command. This will connect to Mathematica running on your computer system.

Note

To connect using the TCP protocol or to connect to Mathematica running on another computer system, enter the port name followed by "@hostname", for example **"3000@sunserver"**.

1. After entering your connection information, click **Connect**.

   If Mathematica is active and waiting, the connection may be immediate. The status message changes to indicate that the link is being attempted or that it has been opened. When the link is established, the status line changes to indicate success. If the link times out or fails, the status line provides an error message.

   The connection may fail because Mathematica is not ready or cannot be found at the specified port location. If so, check your connection address and correct it. Select an alternate Mathematica link (such as a copy of Mathematica running on a different workstation) and go back to Mathematica to check the status of your **TransformConnect[]** command.

2. Click **Close** to close the MathLink dialog and allow Transform and Mathematica to work with the open link.

   The link remains open until it is closed by a command from Mathematica or Transform.

Connecting Mathematica 2.2.x to Transform for Windows

If you are using an older version of Mathematica we recommend that you upgrade to the latest version. You can, however, link older versions of Mathematica to Transform 3.4. You can follow the same setup instructions provided in the previous pages, but you will need to make the following changes:

1. In the Mathematica notebook, use the **TransformConnect["3000", "TCP"]** command. Then in Transform, choose **MathLink...** from the Numbers menu, and in the “Link to” dialog type “3000@ip address”. This option requires your Windows machine to have an IP address.

2. Make sure that MLINK32.dll is installed in the c:\WNMATH22 directory.

3. Then modify the trnsfrm3.ini file to be:

   ```
   mthlnk32= c:\WNMATH22\MLINK32.dll
   ```

   If your Windows machine does not have an IP address then you will have to upgrade to the latest version of Mathematica before you can use it with Transform.

Using the Mathlink Connection in Transform for Windows

Below are the Mathematica commands that are used with Transform.

TransformCommand["call contour(currentdataset)"]

TransformCommand passes any string, or list of strings, to Transform for processing as a macro expression. When successful, Transform returns the symbol **$TransformReady** to Mathematica. Otherwise the symbol **$Failed** is returned. Transform commands commonly use quote marks as part of the command itself. To provide a quote mark within a Mathematica expression, use the backslash before the quote. For example, the following command opens a file by name:

```
In[8]:= TransformCommand["call open(\"myfilename\")"]
```

See below for an in-depth example using Mathematica to script Transform macros.

myMathematicaData = TransformGetArray["mydata"]
TransformPutArray[myMathematicaData, "mydata"]

TransformGetArray[] and **TransformPutArray[]** exchange data with Transform. **TransformGetArray[]** transfers an array from Transform to Mathematica; **TransformPutArray[]** does the opposite. When you read a dataset, **TransformGetArray[**] is the name of any Transform dataset window and a 2D Mathematica Table is returned. When you write a table with **TransformPutArray[]**, the first parameter is the name of the table in Mathematica and the second parameter is a string which will be used as the name of the dataset in Transform.

avalue = TransformGetValue["myvariable"]
TransformPutValue["thevalue", "myvariable"]

TransformGetValue[] and **TransformPutValue[]** exchange simple variables with Transform. **TransformGetValue[]** returns the value of the named variable. **TransformPutValue[]** assigns the named variable the value given. This is how you specify Transform keywords, used to set up and create plots. All values are exchanged as strings even if they represent numerical values.

TransformLinkName[]

TransformLinkName[] returns the Link identifier for the MathLink connection to Transform. It is useful when you need to use the Mathematica commands which operate directly on links, such as **Links[]** and **LinkError[]**.

Recovering From Errors in Transform for Windows

The Transform MathLink commands all return a value, the symbol **$Transform-Ready**, or upon error, the symbol **$Failed**. When Transform responds with **$Failed**, it places a descriptive error message in the macro variable **mathlink_lasterror**. The value of this variable may be displayed in a Transform notebook using the **print** command, or it may be fetched by Mathematica using the **TransformGetValue[ ]** command. For example:

```
In[6]:= TransformGetArray["test"]
Out[6]= $Failed
In[7]:= TransformGetValue["mathlink_lasterror"]
Out[7]:= Unknown dataset--'test'
In[8]:=
```

When you suspect that the link has become inoperable, you should close the connection and reconnect between Transform and Mathematica. Close the Transform side first by bringing up the MathLink dialog and clicking the **Disconnect** button. As soon as Transform closes the link, Mathematica detects the action and aborts any waiting MathLink command.

Note

If you are using the TCP protocol for MathLink, you may find that the socket number you chose cannot be reused for several minutes after the link is closed. Choose a new socket number until the previous one times out and becomes available again.

Closing the Mathlink Connection in Transform for Windows

You may close either side of the MathLink connection, the Transform side or the Mathematica side. You may also Exit from either program to disconnect.

To release the shared system resources being held, you should always close both sides of the connection.

Closing in Transform

Open the MathLink dialog again with the **MathLink...** menu command and click the **Disconnect** button.

Closing in Mathematica

Use the TransformDisconnect[] command.

```
In[12]:= TransformDisconnect[]
```

Mathematica Examples in Transform for Windows

The following Mathematica programming examples can be found in the MathLink Examples notebook shipped with Transform.

Example Program 1

In this example we evaluate an analytical function for use in Transform. It creates a table of numbers and sends them to Transform, then creates an image from the data. It assumes the link is already open.

```
atable = Table[ Sin[x] Cos[y], {x,-3,3,0.4},{y,-3,3,0.4}] ;
TransformPutArray[atable, "sincos"]
TransformCommand["currentplot = image(\"sincos\")"]
```

Example Program 2

The following example generates a sequence of datasets from an analytical expression in Mathematica and calls Transform to generate the graphic output, one image for each dataset. The images are saved to an HDF file. The resulting file is then converted to an AVI animation file, which may be viewed using the Windows Media Player. It also assumes the link is already open.

[Based on the example from Mathematica - A System for Doing Mathematics by Computer, Second Edition, page 175.]

Note

The example below uses a surface macro. Therefore, before running this example, you must create a macro name 'surface_macro', which generates a Hi-Res Color surface plot. See Chapters 22 and 23 for more on macros.

```
Do[tables[t]=Table[BesselJ[0,Sqrt[x^2 + y^2] + t],{x,-10,10},
            {y,-10,10}], {t,0,8,1}];
Do[{TransformPutArray[tables[i],"time"<>ToString[i]],
    TransformCommand["currentdataset=\"time"<>ToString[i]<>"\""],
    TransformCommand["call surface_macro"],
    TransformCommand["call saveas(currentplot,\"besslj.hdf\",13)"],
    TransformCommand["user_interactive=false"],
    TransformCommand["call close(\""<>"time"<>ToString[i]<>"\")"]},
   {i,0,8,1}]
TransformCommand["call makeavi(\"besslj.hdf\",\"besslj.avi\")"]
```

Set Up a MathLink Connection in Transform for Power Macintosh

This section describes the steps required to establish a MathLink connection between Transform and Mathematica.

Setting up in Mathematica

In the 'Mathematica' folder on your Transform program disk is the file 'TransformLink.m'. Be sure and place a copy of this file in your mathematica folder. In Mathematica use the **Needs** or **<< (Get)** command to load this package:

```
In[1]:=   <<TransformLink.m
```

Now you have access to the list of Transform-specific MathLink calls. To set up a connection open a link with the **TransformConnect[]** function:

```
In[2]:=  TransformConnect[]
```

This command does not return with any messages until Transform connects to this link, or it times out (several minutes), or you cancel the command. Now switch to Transform and open the link from the other side (described below). When Transform does connect it returns the symbol **$TransformReady**, so you should see the following message along with a new prompt:

```
Out[2]= $TransformReady
In[3]:=
```

Now the link is open and you are ready to use Mathematica commands with Transform.

Please refer to your MathLink documentation to get a better idea of how connections are established, and of the connection options available.

Setting Up in Transform

To set up MathLink, select **MathLink...** from the Numbers menu to bring up the MathLink to Mathematica dialog.

To connect to a Mathematica application running on your Macintosh, use the default entry, 'TransformLink'. See the TCP/IP section below for information on connecting to Mathematica on another computer.

1. After entering your connection information, click **Open Link**.

If Mathematica is active and waiting, the connection may be immediate. The status message changes to indicate that the link is being attempted or that it has been opened. When the link is established, the status line changes to indicate success. If the link times out or fails, the status line changes to provide an error message.

The connection may fail because Mathematica is not ready or cannot be found. If so, the standard program browser appears. Select an alternate Mathematica link (such as a copy of Mathematica running on a different Macintosh) or choose **Cancel** and go back to Mathematica to check the status of your **TransformConnect** command.

2. Click **OK** to close the MathLink dialog and allow Transform and Mathematica to work with the open link. The link remains open until it is closed by a command from Mathematica or Transform.

Setting up a TCP/IP MathLink Connection

MathLink supports TCP/IP connections when MacTCP is installed on your system. First you need to use a communications program to login to another computer via TCP/IP and run a version of Mathematica on that computer. You also have to run the TransformLink package in Mathematica in order to use a TCP/IP link to Transform.

In your non-Macintosh version of Mathematica, use the TCP/IP form of Transform-Connect:

```
In[2] := TransformConnect["3000"]
```

The parameter is the socket number: a TCP/IP-specific number from 1024 to 32000. Make up any number you like, but it must match on both sides of the link. If an error indicates the socket number is already in use, choose another number. Higher numbers are usually available.

On the Transform side of the TCP/IP link, enter a TCP/IP connection ID in the MathLink dialog. These are of the form **nnnn@hostname** where **nnnn** is the socket number and **hostname** is the network name of the computer that is running the remote copy of Mathematica. An example of this form would be:

```
3000@sunserver
```

Click **Open Link** to open the connection.

Using the MathLink Connection

Below we list all of the supported Mathematica commands for use with Transform.

```
TransformCommand["call contour(currentdataset)"]
```

TransformCommand passes any string, or list of strings, to Transform for processing as a macro expression. When successful, Transform returns the symbol **$TransformReady** to Mathematica. Otherwise an error message is returned.

Transform commands commonly use quote marks as part of the command itself. To provide a quote mark within a Mathematica expression, use the backslash before the quote. For example, the following command opens a file by name:

```
In[8]:= TransformCommand["call open(\"myfilename\")"]
```

See below for a more in-depth example of using Mathematica to script Transform macros.

myMathematicaData = TransformGetArray["mydata"]
TransformPutArray[myMathematicaData, "mydata"]

> **TransformGetArray** and **TransformPutArray** exchange data with Transform. Get transfers an array from Transform to Mathematica and put does the opposite. When you read a dataset, the **TransformGetArray** parameter is the name of any Transform dataset window and a 2D Mathematica Table is returned as the result. When you write a table with **TransformPutArray**, the first parameter is the name of the table in Mathematica and the second parameter is a string which will be used as the name of the dataset in Transform.

avalue = TransformGetValue["myvariable"]
TransformPutValue[thevalue, "myvariable"]

> **TransformGetValue** and **TransformPutValue** exchange simple variables with Transform. **TransformGetValue** returns the value of the named variable. **TransformPutValue** assigns the named variable the value given. This is how you specify Transform keywords, used to set up and create plots. All values are exchanged as strings even if they represent numerical values.

TransformLinkName[]

> **TransformLinkName** returns the Link identifier for the MathLink connection to Transform. It is useful when you need to use the Mathematica commands which operate directly on links, such as **Links[]** and **LinkError[]**.

Recovering from Errors in Transform for Power Macintosh

When you suspect that the link has become inoperable, you should close the connection and re-connect between Transform and Mathematica. Close the Transform side first by bringing up the MathLink dialog and clicking the **Close Link** button. As soon as Transform cuts off the link, Mathematica detects the action and aborts any waiting MathLink command.

If you are using the TCP protocol for MathLink, you may find that the socket number you chose cannot be reused for several minutes after the link is closed. Choose a new socket number until the previous one times out and becomes available again.

Closing the MathLink Connection in Transform for Power Macintosh

You may close either side of the MathLink connection, the Transform side or the Mathematica side. When you do, the other side detects the change in status and disconnects also. You may also **Quit** from either program to disconnect.

Closing in Mathematica

Use the **TransformDisconnect[]** command.

```
In[12]:= TransformDisconnect[]
```

Closing in Transform

Open the MathLink dialog again with the MathLink menu command and click the **Close Link** button.

Mathematical Examples

The following Mathematica programming examples can be found in the 'TransformLink Examples' notebook shipped with Transform.

Example Program 1

In this example we evaluate an analytical function for use in Transform. It creates a table of numbers and sends them to Transform, then creates an image from the data. It assumes the link is already open.

```
atable = Table[ Sin[x] Cos[y], {x,-3,3,0.4},{y,-3,3,0.4}] ;
TransformPutArray[atable, "sincos"]
TransformCommand["currentplot = image(\"sincos\")"]
```

Example Program 2

The following example creates an output sequence in Transform which may then be played back as an animation. It generates a sequence of datasets from an analytical expression in Mathematica and calls Transform to generate the graphic output, one image for each dataset.

[Based on the example from Mathematica - A System for Doing Mathematics by Computer, Second Edition, page 175.]

```
Do[tables[t]=Table[BesselJ[0,Sqrt[x^2 + y^2] + t],{x,-10,10},
      {y,-10,10}], {t,0,8,1}];

Do[{TransformPutArray[tables[i],"time"<>ToString[i]],
      TransformCommand["call surface_macro"],
      TransformCommand["call saveas(currentplot,\"TransMovie\", 9)"],
      TransformCommand["call close(\""<>"time"<>ToString[i]<>"\")"]},
             {i,0,8,1}]
```

Appendix C:

AppleEvents for Power Macintosh

AppleEvents is a Macintosh System 7 (or greater) feature which allows applications to exchange data and commands. To send commands to Transform, you need to use an application capable of sending AppleEvents. This appendix was tested using Macintosh OS 8 and Script Editor 1.1.2 which can be found in the 'Apple Extras:AppleScript' folder. Other resources for AppleScripting can be found on the Apple web site at http://www.apple.com or more specifically http://applescript.apple.com/default.html.

Using ScriptEditor to Control Transform

To send commands from the ScriptEditor to Transform, you can use several different AppleScript commands. The commands you would use most often to run Transform are:

tell

specifies which application the commands are being sent to. If the application is not already running, this command prompts you to launch the application.

end tell

ends the communication between the ScriptEditor and Transform.

activate

brings the target application to the front.

Do Script

Sends a macro command to Transform.

set macro to

used with Do Script macro to send a series of macro commands to Transform. Each macro command should be separated by & return & ¬. To make the Return symbol, press Option-Return.

set menuID to

same as activating a menu item.

set menuItem to

same as choosing a command from a menu.

set menuCode to menuID * 65536 + menuItem

the formula to calculate the menu code to send to Transform.

Do Menu menuCode

sends the menu code to Transform.

In addition, all text preceded by a -- indicates a comment.

Note

Although you can use AppleScript's **Do Menu** command to control Transform, you may find it easier to send macro commands to Transform using the **set macro to** command.

ScriptEditor Example

Here we present an example of using the ScriptEditor to send commands to Transform. In this example, ScriptEditor opens Transform, makes Transform the active application in the Finder, and then sends a series of macro commands to Transform. The commands tell Transform where to look to open a file, then create an image from that file. The **Do Menu** command is used to send the active image window to be printed. An additional series of macros are then used to save then close the file.

```
tell application "Transform PPC" --open communication with the app
      activate -- makes the target application active in the finder
      Do Script "call beep"
      set macro to ¬
        "path = 'HardDrive:Transform:Samples:Tstorm'" & return & ¬
        "call setfolder(path)" & return & ¬
        "call open('Xvel.hdf')" & return & ¬
        "user_interactive = true" & return & ¬
        "image_axes = true" & return & ¬
        "new =  image(currentdataset)" & return

      --the macro sets the path, opens a file and makes an image

      Do Script macro -- executes the lines in the "set macro to"

      set menuID to 129 --"same as choosing File on the menu"
      set menuItem to 10 -- "same as choosing Print off the File menu"
      set menuCode to menuID * 65536 + menuItem --menuCode formula

      Do Script "user_interactive = false" --turn off user prompts
      Do Menu menuCode -- Prints the active window in Transform

      Do Script "call setsavefolder('HardDrive:Transform:Samples')"
      Do Script "call saveas(currentdataset, 'Test.hdf', 1)"
      -- saves Xvel.hdf file into new Test.hdf file in Samples folder

      quit -- quits Transform

end tell -- ends communication with Transform
```

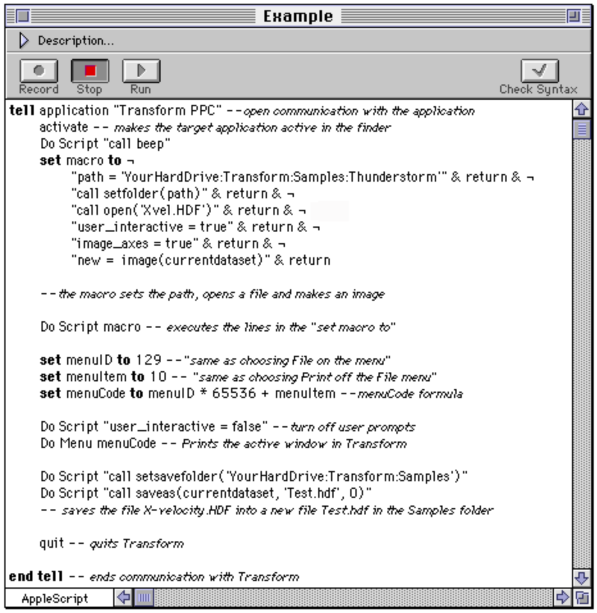


Figure C-1: ScriptEditor

Click the **Run** button to execute the script.

Note

You have to change path names to match your system hard drive.

Command Formatting

You can send multi-line scripts to Transform with a single **set macro to** command. However, since only one error code is returned from one command, it may be difficult to track down errors in multi-line scripts. You can create them by concatenating lines with return characters as in the following example:

```
set macro to "user_interactive=false" & return & ¬
"currentplot=contour(currentdataset)" & return
```

The ampersand is a concatenation operator and it is not passed to Transform. The special keyword return creates a return character which is added to the script that is passed to Transform. This return character is crucial because it separates commands.

Embedding quotes in commands works the same way. Because the double quote is used as part of the AppleScript language, we use single quotes where you would normally use double quotes in the Transform macro language. For example, in Transform, the filename Xvel.hdf would be in double quotes (e.g. "Xvel.hdf"). In the ScriptEditor, the proper syntax is as follows:

```
set macro to "call open('Xvel.hdf')"
```

Suppressing User Interaction

To suppress user interaction when sending macro commands, set the **user_interactive** flag to **false**. To set it from an external script, send the following:

```
set macro to "user_interactive=false"
```

MenuID Numbering in Transform

The following represents the menuID numbering in Transform.

Menu Name	menuID Number
File	129
Edit	130
Image	131
Numbers	132
Macros	134
Windows	135
Color Tables	136

The menuID is used to calculate the menu code:

```
set menuCode to menuID * 65536 + menuItem
```

MenuItem Calculation

A menuItem represents a command on a menu. Each menuItem is determined by its placement on the menu. The first item on the menu is number one; the rest of the commands are consecutively ordered as they appear in the menu. Any dividing lines on a menu are included in the number count. For example, in the Numbers menu, **Attributes** is menuItem number one, the first divider is menuItem number five, **Calculate From Notes** is menuItem number seven and MathLink is number thirteen.

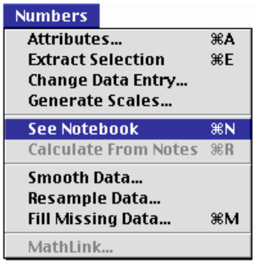


Figure C-2: Numbers Menu

Appendix D:
HDF Reference

HDF is an extensible, binary, public domain file format specification for storing data and images. All Fortner Software products use HDF as their primary data storage format. HDF files can store floating point data, scaling information, color images, text, and other items. HDF originated at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, where it was developed as a solution to the problem of sharing data among all of their different computers.

NCSA maintains and distributes a public domain software library for reading and writing HDF format files. The library is available on a variety of computers including Macintosh, Windows, Sun, VAX, Silicon Graphics, and Cray UNICOS. The software is written in C with both Fortran and C interfaces supported for making calls to the HDF libraries.

HDF Source Code, Manuals, and Web Sites

If you require total control of the contents of an HDF file through a C programming environment or you are interested in understanding the full details of the HDF format, you may utilize the HDF libraries.

Source code and documentation for the HDF storage routines is available from NCSA at the University of Illinois and Fortner Software. The libraries can be obtained from NCSA's HDF home page at http://hdf.ncsa.uiuc.edu or by anonymous ftp to ftp.ncsa.uiuc.edu.

HDF-EOS

HDF-EOS is a convention for HDF established by NASA for earth science data. NASA's Earth Observing System (EOS) is part of the Earth Science Enterprise project, an $8 billion, 15-year project to monitor long-term global environmental change. This project alone will produce terabytes of information per day, resulting in an overwhelming surge of scientific data stored in HDF and HDF-EOS.

NASA will administer the data flow from the satellites to Earth receiving stations called Distributed Active Archive Centers (DAACs) and make the data available to the public via the Internet within 24-48 hours. While the data is available to everyone outside of NASA, it is of particular interest to the scientists in the Environmental Protection Agency, National Weather Service, NOAA, Department of Fishing and Wildlife, Department of Agriculture, etc. Commercial users will include companies in the oceangraphic, mining, petroleum, agriculture and other industries.

HDF-EOS files are different from standard HDF files because they include three earth science data objects (point, swath and grid) which are not defined within the standard HDF library:

- A point object is a data group that contains data with geolocation information.
- A swath object is a data group that contains time-ordered data (e.g., swaths scanned by satellites).
- A grid object is a data group that contains projection data stored in a rectangular array.

HDF-EOS Source Code, Manuals, and Web Sites

For more information about HDF-EOS, go to the HDF-EOS Information Resources web site located at http://hdfeos.gsfc.nasa.gov/hdfeos/hdf.html. This site provides you access to the HDF-EOS documentation and libraries.

Web Sites

HDF and HDF-EOS

Additional information about HDF, HDF-EOS and their application can be found on the World Wide Web at the following web sites:

- HDFinfo.com - a clearinghouse of HDF information: http://www.hdfinfo.com/
- NCSA's web page: http://hdf.ncsa.uiuc.edu
- NASA Goddard Space Flight Center (GSFC) page on HDF: http://daac.gsfc.nasa.gov/REFERENCE_DOCS/HDF/gdaac_hdf.html
- Earth Science Enterprise web page: http://www.hq.nasa.gov/office/mtpe/
- EOS Data Resources web page - a list of all data centers distributing terabytes of science data in HDF: http://eospso.gsfc.nasa.gov/eos_homepage/

Data Download

The following web sites let you access data to be downloaded.

- TOMS (zipped): http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/FTP_SITE/readmes/toms_daily.html
- GOME: http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/FTP_SITE/readmes/gome_daily.html
- CZCS: http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/OCDST/czcs_readme.html
- SeaWiFS: http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/OCDST/OB_main.html
- SSMI: http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/hydrology/readme_html/ssmi_monthly_readme.html
- TRMM: http://lake.nascom.nasa.gov/DATA/TRMM/
- NSCAT: ftp://podaac.jpl.nasa.gov/pub/ocean_wind/nscat/
- DTED:
  http://164.214.2.59/geospatial/products/DTED/dted.html
  http://164.214.2.54/mel/data.html

- SDTS: http://edcwww.cr.usgs.gov/doc/edchome/ndcdb/ndcdb.html

Note that the SeaWiFS and TRMM sites require you to order the data for a fee. With the exception of the DTED and SDTS web sites, all data is stored in HDF. Noesys supports import of DTED and SDTS to HDF as described in the "Importing Files" chapter of the *Noesys User's Guide and Reference Manual*.

Appendix E:

Kernel Functions

Transform can calculate kernel convolution operations using built-in 3x3 kernels or using kernels you construct yourself. These kernel functions can be useful for taking derivatives, sharpening edges of the data, smoothing the data, etc.

Here is a listing of the kernel functions.

ddx(q)	dq/dx - difference from left to right
ddy(q)	dq/dy - difference from top to bottom
d2dx(q)	d^2q/dx^2 - second derivative from left to right
d2dy(q)	d^2q/dy^2 - second derivative from top to bottom
lap(q)	5 point laplacian
lap5(q)	same as lap(q)
lap9(q)	9 point laplacian
kernel(q,k)	generic kernel

Kernel Convolutions

A 3x3 computational kernel has the layout shown in Table E-1, where the value to be computed is in the center. The individual elements of the kernel make up coefficients in the equation used to find a new center value. The coefficients are multiplied with the data values in the array and added together. This kernel operation is repeated once for each member of the data array.

i-1 j-1	i j-1	i+1 j-1
i-1 l	i j	i+1 j
i-1 j+1	i j+1	i+1 j+1

Table E-1: Layout of 3x3 Kernel

In Figure E-1, the shaded square marks the current array element to be computed. The surrounding 3x3 grid of nine elements is used to compute the value in the middle for the result array.

The surrounding 3x3 grid of numbers is used for each point in the resulting array, along with the 3x3 kernel of coefficients, to compute the results with the following formula. If array 'K' contains the nine kernel coefficients, and array 'P' contains the computational subgrid diagrammed in Figure E-1, the resulting value for the center, '$f_{i,j}$', is given by:

$$f_{i,j} = \sum_{a=i-1}^{i+1} \sum_{b=j-1}^{j+1} P_{a,b} \, K_{a-i+2,b-j+2}$$

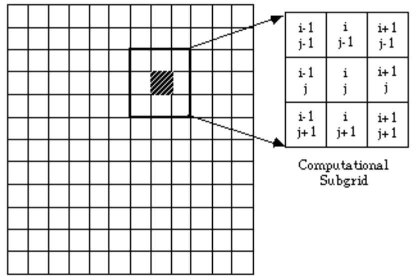


Figure E-2: 3 x 3 Kernel in Array Grid

Calculations are done once for each point within the computed region, shown in Figure E-2. Because the kernel computations require neighbors on each side in all directions, the computed region extends to one element from the edge. At the end of the kernel operation, the edge values are filled in by copying them from their nearest neighbors.

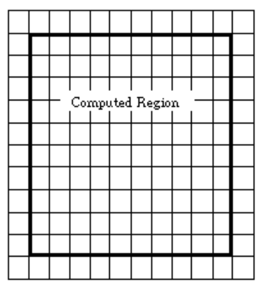


Figure E-3: Computed Region

When using the generic kernel function, you provide the names of two arrays, one to be used for data input, and the other containing the kernel coefficients. The computation proceeds across the rows of the dataset and down, evaluating the kernel at each point to produce an answer for the resulting array. The next section describes how to set up the kernel array.

Generic Kernel Examples

For example, to perform a differencing equation to approximate the 'x' derivative of an array that has equally spaced columns, one unit apart:

1. After opening your source array, select See Notebook from the Numbers menu. A notebook window appears.
2. Enter the following array into the Notebook window, with each number separated by a tab and a hard return at the end of each row.

0	0	0
-1	0	1
0	0	0

3. Select the array and choose **Copy** from the Edit menu. The array is copied onto the Clipboard.
4. In Windows, select a non-notebook window and choose **Paste As…** from the Edit menu; select **New Dataset** in the Paste As dialog and click **OK**. In Power Macintosh, select **Paste New** from the Edit menu. This creates a new data window.
5. Name the window 'd_kernel' using the Attributes dialog from the Numbers menu. You now have a usable kernel.
6. Enter and select the following formula in the notebook window.

   ```
   diffx = kernel(q, d_kernel)/2
   ```

   where ‘q’ is the name of your source array.
7. Choose **Calculate From Notes** from the Numbers menu. The command applies your 3x3 convolution to the data and produces a new dataset the same size as 'q'.

Look at the image to verify that each data point consists of the value to its right minus the value to its left divided by 2.

Try using the kernels shown in Tables E-2 through E-4 for experimentation. The kernel shown in Table E-2 smooths the data, averaging the center value with its neighbors, reducing any spikes and sharp transitions. The left kernel in Table E-3 brings out horizontal lines in the data; the right kernel brings out vertical lines. Finally, the kernel in Table E-4 sharpens edges in the data.

1	0	1
0	1	0
1	0	1

Table E-4: Kernel that Smooths Data

1	1	1
0	0	0
-1	-1	-1

1	0	-1
1	0	-1
1	0	-1

Table E-5: Kernels that Detect Horizontal and Vertical Lines of Data

1	-1	1
-1	1	-1
1	-1	1

Table E-6: Kernel that Sharpens Edges in Data

Kernels For Built-in Functions

This section presents the coefficient matrices for the built-in kernel functions (Tables E-5 through E-10). Certain constants are used throughout:

- The average width of the rows and columns— x and y:

  $$x = (x_{max} - x_{min}) / (n_{cols} - 1)$$
  $$y = (y_{max} - y_{min}) / (n_{rows} - 1)$$

  These constants are valid only when the columns are evenly spaced.

- The row and/or column differences for a point—h_i and h_j:

  $$h_i = x_{i+1} - x_i$$
  $$h_j = y_{j+1} - y_j$$

  where 'x_i' is the column scale value at column 'i' and 'y_j' is the row scale value at row 'j' value at row 'j'

  $$A = -h_i / [\ h_{i-1}\ (h_i + h_{i-1})]$$
  $$B = -\ (A + C)$$
  $$C = h_{i-1} / [\ h_i\ (h_i + h_{i-1})]$$
  $$f_i,j = A*P_{i-1,j} + B*P_{i,j} + C*P_{i+1,j}$$

A	B	C

Table E-7: ddx Kernel

$$A = -h_j / [\ h_j-1\ (h_j + h_j-1)]$$
$$B = -\ (A + C)$$
$$C = h_j-1 / [\ h_j\ (h_j + h_j-1)]$$
$$f_{i,j} = A*P_{i,j-1} + B*P_{i,j} + C*P_{i,j+1}$$

	A	
	B	
	C	

Table E-8: ddy Kernel

$$A = 2 / [\ h_{i-1}\ (h_i + h_{i-1})]$$
$$B = -\ (A + C)$$
$$C = 2/\ [\ h_i\ (h_i + h_{i-1})]$$
$$f_{i,j} = A*P_{i-1,j} + B*P_{i,j} + C*P_{i+1,j}$$

A	B	C

Table E-9: d2dx Kernel

$$A = 2 / [\ h_{j-1}\ (h_j + h_{j-1})]$$
$$B = -\ (A + C)$$
$$C = 2 / [\ h_j\ (h_j + h_{j-1})]$$
$$f_{i,j} = A*P_{i,j-1} + B*P_{i,j} + C*P_{i,j+1}$$

	A	
	B	
	C	

Table E-10: d2dy Kernel

Because of the use of x and y in the Laplacian approximations, they are only valid for evenly spaced rows and columns.

$$A = 1/(\ y^2)$$
$$B = 1/(\ x^2)$$
$$C = -2\ (A + B)$$
$$f_{i,j} = A*(P_{i,j-1} + P_{i,j+1}) + B*(P_{i-1,j} + P_{i+1,j}) + C*P_{i,j}$$

	A	
B	C	B
	A	

Table E-11: lap and lap5 Kernel

$$h = (x + y)/2$$
$$A = 1/[4 (h)^2]$$
$$B = 2A$$
$$C = -12A$$
$$f_{i,j} = A*(P_{i-1,j-1} + P_{i-1,j+1} + P_{i+1,j-1} + P_{i+1,j+1})$$
$$+ B*(P_{i,j-1} + P_{i,j+1} + P_{i-1,j} + P_{i+1,j}) + C*P_{i,j}$$

A	B	A
B	C	B
A	B	A

Table E-12: Lap9 Kernel

Appendix F:

External Functions (Macintosh only)

If you have access to a C compiler, you can write your own external notebook functions. Transform comes with a sample external function tested with the Metrowerks C compiler (IDE version 2.0). You can use these examples to create your own external functionality.

Using External Functions

External functions are stored in library files. Anytime Transform cannot find a function name in its internal list, a dialog box (Figure F-1) appears to help you choose an external library file. If you misspelled the function name, click the Return to Notebook button; otherwise click Select Library File and proceed to select and open the library from disk. Once you open a library, you can use any of the functions within it; all the functions remain available until you exit Transform.

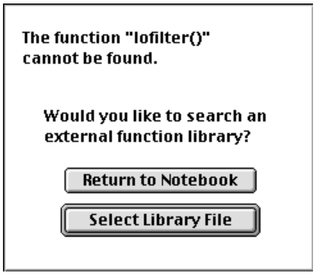


Figure F-1: Choosing an External Function Library

In addition, the popup menu in the Notebook window labeled 'Externals' provides easy access to all the names of functions that are available from open library files. You must open the library file as described above first before the names show up in the popup menu. Then you may choose functions from the menu.

Sample Files

rotate.c

Takes the name of one data set and one constant as arguments. It will cause the image to be shifted by "constant" number of columns to the right if positive constant or left if negative.

stdev.c

Computes the mean and standard deviation of a dataset and creates a new dataset comprised of the number of standard deviation each entry in the original dataset is away from the mean.

rotate. and stdev.

Metrowerks project files that can be used to build and modify External functions.

rotate and stdev

Prebuilt versions of External functions that can be called from Transform.

Compiler Settings for Creating Resources

External functions have specific settings that need to be set within the compiler environment. This section explains generically what these settings are and specifically how to set them within the Metrowerks environment. Metrowerks projects are provided with Transform with the settings already defined. These working examples can be found in the 'External Functions' folder which is part of the default Transform installation. Similar instructions could be used for other Macintosh compiler environments. Refer to the specific compiler documentation for information on how to create code resources with non-Metrowerks compilers.

There are several settings required to set up an external function. Within the Metrowerks environment, many of these will be set up in the Target Settings Panel by choosing the PPC target pane.

The first setting that needs to be made is the external function must be created as a code resource. This is done in Metrowerks by going to the Target Settings Panel, choosing the PPC Target pane, and selecting the Code Resource menu item on the Project Type popup menu.

It is recommended that the code resource be given the same name as the function that is to be called. Within the same pane, set the file name and resource name edit fields to have the same name as your desired function.

There are four more resource specific fields that need to be set within this pane. The Creator of the code resource must be set to FRtr, the Type must be set to DSff, the ResType must be set to DSfn, and each external function must be given its own unique resource ID number or ResID. It is recommended to start numbering at 1001 and number consecutively thereafter 1002, 1003, ...

Note that you may place more than one CODE resource into a resource file. However, in doing so each module must have a unique ID and name.

Creating an External Function in C

External functions are limited in the number and types of arguments they can accept. Each external function is called with either one or two arguments. If there are two arguments, the first is called the 'left' argument, the second called the 'right' argument. The first argument is always a Transform dataset, the second argument either a Transform dataset or a constant. The result of the function can either be a Transform dataset or a constant. Each of these parameters is defined by a structure of type scope_array which is listed below. This is the format for the data that is passed to and from the external function. An example of an empty function is given below. In your function, take information from the left and/or right parameters and place the resulting information into the structure provided for the answer. It is important that the C name of each module is “main” and must return a long integer, as shown in the example files “stdev.c” and “rotate.c”

```
long main(left,right,answer)
scope_array *left,*right,*answer;
{
/* contents of function */
}
```

The named function of the code resource must be the first executable code in the C file

Structure Definition

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Data structure for external functions.
The field 'kind' determines whether an argument is of constant or
array type.
kind is set to DS_CONSTANT or 0 if the structure contains a constant
value. The value of the constant will be stored in the cval field.
'kind' is set to DS_ARRAY or 1 if the structure contains an array. An
array is described by the set of ncols, nrows, rows, cols, and vals
fields.

External functions are called as:
your_fn(left,right,answer)
scope_array
*left, /* left parameter */
*right, /* right parameter */
*answer;  /* place to put the answer */

Answer will always contain pre-allocated space for an array of
resulting values, including the rows and cols arrays. You can
change any value in the rows,cols and vals arrays.
```

```
Do not change any values in the left or right storage.
If your routine returns only a constant, set kind == DS_CONSTANT
and put the answer in cval.
DON'T allocate anything you don't free yourself.
FREE everything you allocate.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*/
#define DS_ERROR -1
#define DS_CONSTANT 0
#define DS_ARRAY 1

typedef struct
{float cval, /* constant value when we are carrying a constant */

*rows, /* row labels, scale values: count = ncols */
*cols, /* col labels, scale values: count = nrows */
*vals; /* data values in the array, if there is an
array size = ncols*nrows */
int ncols,nrows; /* dimensions of the array */

char kind; /* ERROR, CONSTANT, ARRAY */
}
scope_array;
```

Appendix G:

Troubleshooting

This chapter is intended to be of assistance should you have problems running Transform on your system.

Video (Windows only)

Transform demands a high standard of performance for the video driver and some drivers do not meet the required standards.

You may want to consult the Technical Support System Information dialog to determine the current state of your video adapter; this dialog can be activated by pressing the **Support...** button on the About Transform dialog, which is available from the Help menu.

The **Save...** button on this dialog will save the technical information as a text file, suitable for printing, faxing, or electronic mail.

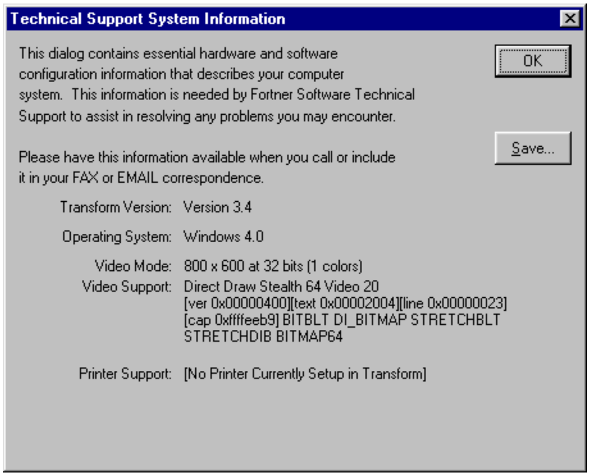


Figure G-1: Technical Support Information Dialog

There are two recommended options for getting an improved video driver. One option is to contact the maker of your video board and get the most up-to-date video driver for your board. Another option is to get a copy to Microsoft's Super VGA driver, a high quality driver that works with a variety of video boards. This driver is available by anonymous ftp from ftp.microsoft.com and is located in /Softlib/MSLFILES/SVGA.EXE. The SVGA.EXE file is a self-extracting archive.

Color Palette

Images generated in Transform are optimally displayed when your system is configured in at least 8-bit mode (256 colors). If not, you will still be able to view datasets, attributes and labels, but images requiring color may either be degraded or may not be generated.

To ensure your display is set correctly in Windows, go to the Start menu and select **Control Panel** from the Settings submenu. Double-click the **Display** icon from the Control Panel folder and select the **Settings** tab. Select the **256 Color** or greater setting from the Color palette pull-down menu.

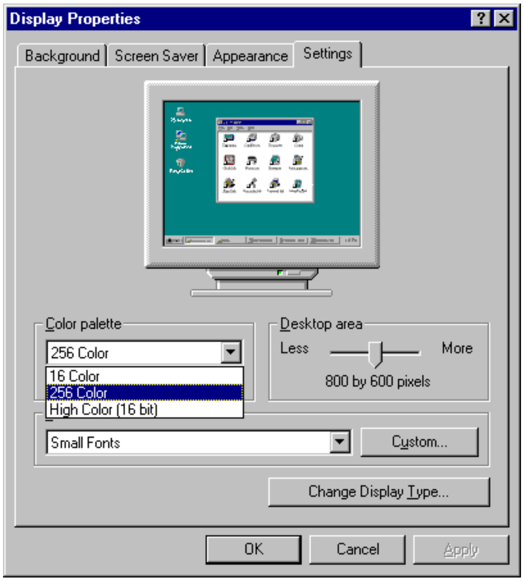


Figure G-1: Windows Display Properties Dialog

You will receive a message asking you to restart your machine so that the new settings will take effect.

Slow Performance

Transform is a computational-intense application that relies heavily on floating-point arithmetic and large amounts of memory. Should you find Transform to be exceptionally slow, please check the following:

- Does your system have a math coprocessor?

  Though Transform will run on systems without a math coprocessor, performance will be extremely slow. Therefore, a coprocessor is strongly recommended.

  486DX, 486DX2, and Pentium™ systems have floating-point hardware built into the CPU. Other CPUs, such as the 386SX, 386DX, and 486SX, require a separate coprocessor chip.

  If you are unsure whether your system has a coprocessor, you may find out by using the Microsoft Diagnostics program, MSD.EXE, located in your WINDOWS directory. You should exit Windows before attempting to run this program.

- Does your system have sufficient physical memory?

  Depending upon the size of your datasets, Transform can consume large amounts of memory. We recommend that your system have at least 8MB of physical memory. Transform can run with less memory, but with reduced performance. Expanding your paging space will allow you to load larger datasets or maintain more windows, but will not compensate for performance loss due to insufficient physical memory.

  If your datasets are very large or you wish to work with many of them at the same time, you may want to consider adding additional memory to your system.

HDF Libraries

Saving to Full Disks

When saving a file, make sure there is enough disk space available. Trying to save an HDF file when there is not enough disk space may cause HDF file corruption.

Compressed Datasets

Transform 3.4 uses the latest 4.1r1 HDF libraries; however, Transform is not compatible with compressed, or chunked and compressed HDF datasets.

Frequently Asked Questions

1. How can I create an image without a color bar?

   You can create images with axes but without color bars by using a macro. First, create an image with the attributes you want, then select **Create Macro** from the Macros menu. Next, select **Edit Macro** from the macros menu, and edit the macro you just created.

   As you scroll down the list of variable definitions, you will notice several sections starting with the line "*axes record", and ending with the line "call addaxis(currentplot,#)" Delete the third set of these axes definitions that end in the line

   ```
   "call addaxis(currentplot,4)"
   ```

   From this point on, you can create an image without a color bar by selecting the macro you previously created from the Macros menu.

2. What does **var()** do? When and how do I use **var()**?

   The subroutine **var()** will return the actual dataset (the numbers) from a string representing the dataset name or a window name. The subroutine **var()** is most commonly used with the reserved variable **currentdataset**. Using **var()** will allow you to use currentdataset within a function that expects a numerical argument. Note that in the manual where it lists a parameter q, it requires the actual data, and where it lists a parameter name (e.g., datasetname or windowname), it requires a string.

   Consider the following example:

   ```
   sinx = sin(var(currentdataset))
   ```

   Since the function sin expects an numerical argument you use var() to convert the string stored in currentdataset to the actual data. A new dataset named sinx is created such that each element in sinx is the sine of the corresponding element in currentdataset. Without the use of var() the sin function would return an answer of zero, since the value of the string represented by **currentdataset** is zero. Functions may also return an error “This function requires an array.” Most likely you need to use **var()** here too.

3. In my macro I have a line that reads

    ```
    "new = smooth(currentdataset, 2) "
    ```

    When I execute it, I get an error message, "This expression requires an array". What is causing this error?

    The variable **currentdataset** is a string variable that is set to the window name of an array and is not actually equal to the array itself. The function **smooth()** requires the array to be passed as an argument. Use the “var” command to access the array that the variable **currentdataset** refers to. The correct syntax is:

    ```
    "new = smooth(var(currentdataset), 2) "
    ```

4. How can I import an Excel spreadsheet into Transform?

    It is very simple to import spreadsheets from Excel into Transform. In Excel, under the File menu, select **Save As...** Choose "Text (Tab delimited)" from the **Save File as Type** menu. Note that this will only save the active sheet in your Excel notebook. Now, you can open the file in Transform as ASCII Text Columns or ASCII Text Matrix. You will have to specify the type of delimiter used to save the file in Excel (in this case Tab delimited).

    Other popular spreadsheets, such as Lotus 1-2-3 and Quattro Pro have similar methods for saving spreadsheets to ASCII text.

5. How can I open a series of slices from a 3D file, image each slice and save it to make an animation?

    Power Macintosh: please refer to the Anim3DData.txt macro in the Advanced Macros section of Chapter 22.

    Windows: this is available from the Research Systems web site (www.rsinc.com) under “Tech Tips.”

6. How can I let Transform automatically calculate the axis minimum and maximum, but allow me to select my own label spacing?

    There is a way to have Transform calculate the axis min and max and still let you specify the label interval. It can be done using the macros. In order to execute the macro properly, it is recommended that you create an image and save it as a macro. Then, you edit the image macro by inserting the following lines in the appropriate place.

```
***********************************************
    *Turn axis_auto and axis_autospacing off
    axis_auto = false
    axis_autospacing = false
***********************************************
    *Transfer X axis information to dataset
    xaxis_scale = x(Untitled)

    *Extract X axis min & max values
    xaxis_min = min(xaxis_scale)
    xaxis_max = max(xaxis_scale)

    *Set X axis_min and max
    axis_min = xaxis_min
    axis_max = xaxis_max

    *Round X axis min & max values for labeling
    xaxis_labelmin = ceiling(xaxis_min)
    xaxis_labelmax = floor(xaxis_max)

    *Set X axis_labelmin and max
    axis_labelmin = xaxis_labelmin
    axis_labelmax = xaxis_labelmax
***********************************************
    *Transfer Y axis information to dataset
    yaxis_scale = y(Untitled)

    *Extract Y axis min & max values
    yaxis_min = min(yaxis_scale)
    yaxis_max = max(yaxis_scale)

    *Set Y axis_min and max
    axis_min = yaxis_min
    axis_max = yaxis_max
     *Round Y axis min & max values for labeling
    yaxis_labelmin = ceiling(yaxis_min)
    yaxis_labelmax = floor(yaxis_max)

    *Set Y axis_labelmin and max
    axis_labelmin = yaxis_labelmin
    axis_labelmax = yaxis_labelmax
***********************************************
    *Close temporary datasets without saving
    user_interactive = false
    call close("xaxis_scale")
    call close("yaxis_scale")
    user_interactive = true
**************************************************
```

7. How can I automatically locate and draw (with a dot) the maximum (or maxima) in a contour plot?

   While it is not possible to automatically locate and draw a symbol on a contour plot for the maximum value in the dataset, there is a way to produce a black dot for the maximum value in a contour image. To make the image, use the following procedures:

   - Create your contour image and turn off the axes.
   - Select **Copy** from the Edit menu.
   - Apply the following macro to your dataset.

     ```
     maximum = max(Your Dataset Name)
     x = LTmask(Xvel, maximum)
     ```

   - From the new 'x' dataset, create an image.
   - With the new image active, select Paste from the Edit menu.

8. In Transform, how can I translate matrix data to column data?

   Using the macro Matrix-Column, described in the Advanced Macros section of Chapter 22 you can turn your matrix data into column data.

9. I often spend over an hour making a plot/image exactly as I want it. How do I make a second image that is 98% similar to the first?

   Once you make a plot/image that you like, choose **Create Macro** from the Macros menu. This will create a macro that records all of the attributes of your current plot. To apply this macro/image to another dataset bring that dataset to the front, making it the active window. Then go to the Macros menu and choose the macro you have just made. If you want to change some of the labels, or customize each one, you can edit the macro you made and use variables, or user prompts to customize each image/plot. **Note:** If you give a new macro the name of an existing macro, it will automatically overwrite the existing macro.

10. How can I make an animation with Transform?

    To create an animation, make a series of images and save them as HDF files. To animate a series of HDF files they must be numbered sequentially (e.g.,File1.hdf, File2.hdf, File3.hdf). For an example macro on how to create a series of HDF files, see the Advanced Macros section of Chapter 22.

11. I'm trying to open a 2 MB TIFF image in Transform. I've given Transform 10 MB of RAM. Why do I run out of memory before it opens?

    You need to give Transform more memory. When Transform opens a TIFF, it requires a great deal of RAM to process the image. Transform creates and displays a dataset that corresponds to the image and it displays the image. A 2 MB TIFF file requires about 15 MB of RAM to open and process. A good rule of thumb to use when opening TIFF images is to give Transform RAM equal to roughly 8 times the size of the TIFF you are opening. Transform stores numbers in 32-bit format, while some TIFF files can store numbers in as little as 4 bits.

12. **(Windows only)** I have an image file that contains a picture of a map. When I open the file and overlay it onto an image in Transform, the results look great. However, when I try to print it, my machine crashes. Why?

    It is not possible to print an image that has been overlaid onto another image. In order to print, the overlay must be displayed as a contour plot. To accomplish this, follow these steps:

    - Open the image file into Transform. You will get your image and a dataset that is all zeros and ones.
    - In the Notebook type: `Overlay = your_image_datasetname * 255`
    - This will make a new dataset called "Overlay with all zeros and 255".
    - Using the "Overlay" dataset, make a contour plot. In the Contour dialog, set "Number of Automatic Levels" to **1**, and choose "Set Auto Levels", then **OK**.
    - Create another new dataset by choosing **New** from the File menu. Make it 3x3 and all zeros.
    - Create a contour plot from this new dataset. The plot is blank.
    - Remove the axis on the Overlay.contour plot, then copy it.
    - Choose **Paste Overlay** from the Edit menu to paste onto your blank New.contour1 image.
    - Save the new file with the contour image and overlay to an HDF file.
    - Open the HDF file, copy the overlay of the map and paste onto your image. You will now be able to print your image and overlay.

13. When printing, I would like my image to be 4"x2" and centered on an 11"x8.5" page. How can I do that?

    This is best achieved using macros. Specifically, the "image_h", "image_v", "image_marginleft", and "image_margintop" reserved variables can be used to create a custom macro with this functionality. See the description of the Printing.txt macro in the Advanced Macro section of Chapter 22.

14. I have a set of files in a directory that I want to open, make an image from and then save into a new directory. Is there a way I can automate this process?

    The Transform macro language allows you to automate this process. Please refer to the FileLoop.txt macro in the Advanced Macros section of Chapter 22. If your files do not automatically import into Transform, you may also need to look at the ImportMacro.txt file.

15. I have a set of directories/folders that contain sets of files that I want to open, make an image from and then save in new directories/folders. Is there a way I can automate this process?

    The Transform macro language allows you to automate this process. Please refer to the DirLoop.txt macro in the Advanced Macros section of Chapter 22. If your files do not automatically import into Transform, you may also need to look at the ImportMacro.txt file.

16. I have a series of TIFF files that I want to convert to HDF files so that I can use them in T3D to make a 3D visualization. How can I do this?

    The Transform macro language allows you to automate this process. Please refer to the Tiff to HDF.txt macro in the Advanced Macros section of Chapter 22.

17. I have a series of files that are not automatically imported into Transform. Is there a way to avoid filling in all of the import information for each file?

    The Transform macro language can automate the import of non-HDF files for you. Please refer to the ImportMacro.txt macro in the Advanced Macros section of Chapter 22.

18. Previously I was able to import my Matlab files into Transform for Windows, now I get an error:

    "Matlab files containing VAX/VMS D-float or G-float records are not supported".

    The header in the Matlab file shows:

    MATLAB 5.0 MAT-file, Platform: PCWIN, Created on: Wed Apr 2 12:05:45 1997

    How can I import this file into Transform?

    The output format in Matlab Version 5 has undergone significant changes. This is most likely why you get an error trying to import into Transform. To work around this you can export files from Matlab 5 with a -v4 (version 4 flag) and the file will load into Transform. The save command looks like this:

    ```
    save \path\filename variable_name -v4
    ```

19. **(Power Macintosh only)** I have HDF files that are generated from Park Scientific instrumentation. When I open the files in Transform 3.4 the data is all wrong. When I open the files in the latest Noesys version, the dataset values appears correct (the scale values are all the same number and appear to be wrong). If I use Save or Save As from the Noesys File menu and then try to open the file in Transform, again the data is still wrong. How can I look at my Park Scientific HDF file in Transform for Power Macintosh?

    Park Scientific files are not written out with standard HDF library calls and therefore do not import entirely correctly into HDF based programs from Fortner Software. To look at your Park Scientific HDF files in Transform you can use the converter located in the ‘Extras’ folder of the Noesys CD. See the Read Me text file provided for more details.

20. **(Power Macintosh only)** In Transform 3.02 for the Macintosh, I used to be able to add entries to the Tables menu by using View Utility 3.01. Now, when I try to add a new color table to the Color Tables menu using View 3.03 it doesn't get added to the Transform 3.4 Color Tables menu. How do I do this?

    To add a View Utility color table directly onto the Transform Color Tables menu follow these steps:

    - Create the color table in the View Utility (found in the‘Transform:Extras' folder).
    - Go to the Colors menu and choose Save Color Table... and save the file 'Color.bin' as binary. The View Utility saves color tables as 8-bit unsigned integers of 3 rows and 256 columns.
    - Open the Color.bin file into Transform as a binary matrix, unsigned 8-bit integer, no skip bytes, 3 rows and 256 columns.
    - Choose See Notebook from the Numbers menu.
    - In the Notebook type these two lines and save them as a macro by choosing Create Macro... from the Macro menu.

    ```
    mycolortable = transpose(var(currentdataset))
    call savecolortable(var(currentdataset), "mycolortable")
    ```

    - Select your macro from the Macro menu.

    You have added the color table “mycolortable” to the Color Tables menu in Transform for this session. If you always want this color table to appear on the menu, create a Startup_Macro that automates this process. See Chapter 22 for more information on Startup_Macros.

Other Problems

Please refer to the Transform Release Notes for additional help or information that was not available when this manual was printed.

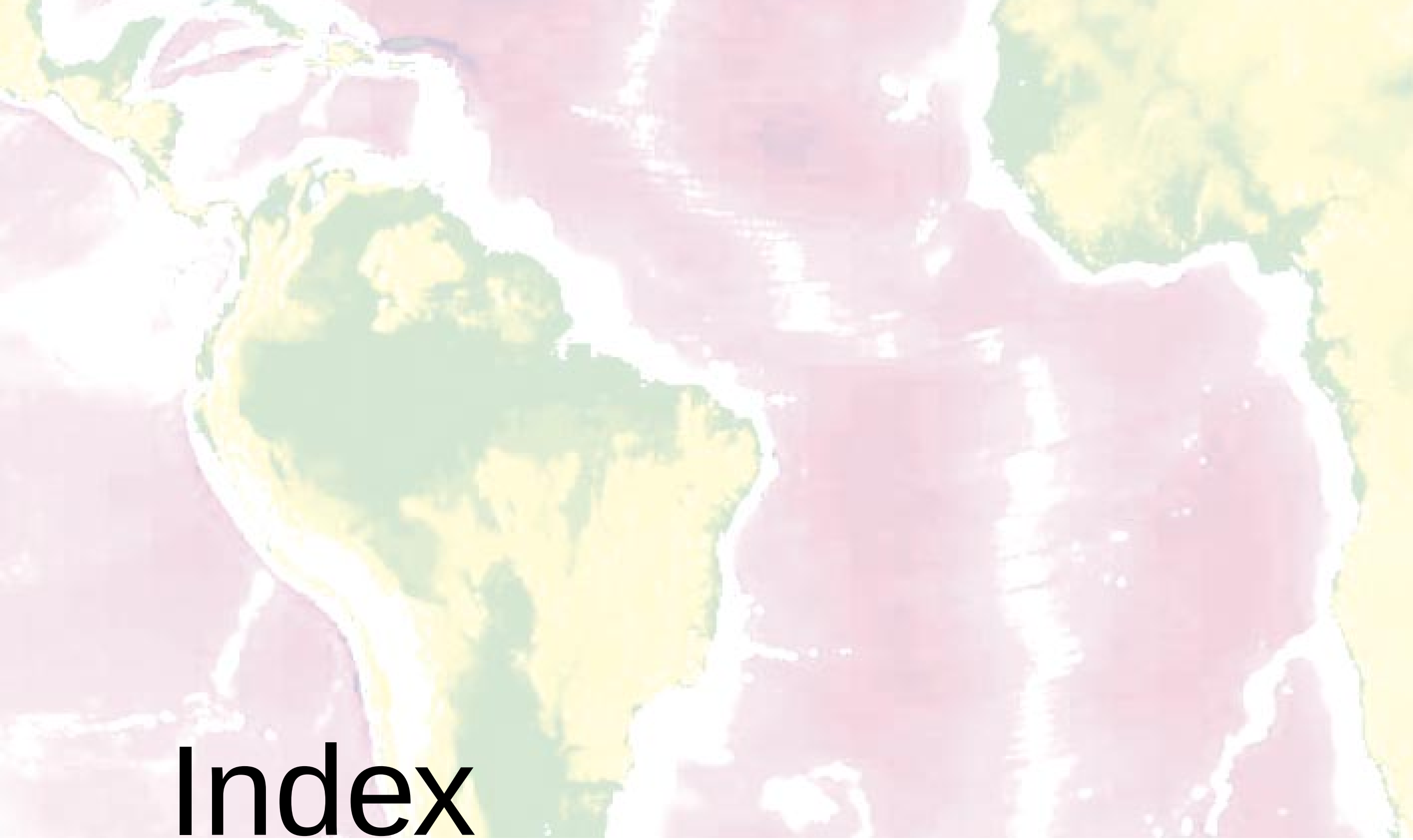
Appendix H:

Suggested Reading

The following list includes publications that you may find useful for background or reference purposes.

- *MathLink® Reference Guide* (1990-1993). Wolfram Research.
- Firebaugh, M.W. (1993). *Computer Graphics: Tools for Visualization*. Wm. C. Brown.
- Fortner, B.I. (1996). *The Data Handbook: A Guide to Understanding the Organization and Visualization of Technical Data*.
- Fortner, B.I. and Meyer, T.E. (1996). *Number by Colors: A Guide to Using Color to Understand Technical Data*.
- Franke, R. (1982). *Scattered Data Interpolation: Tests of Some Methods*. Mathematics of Computation, Vol. 38, Number 157, (January 1982) pp. 181-195. American Mathematical Society.
- Friedhoff, R.M, & Benzon, W. (1989). *Visualization: The Second Computer Revolution*. W.H. Freeman.

- Isaaks, E.H., & Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics*. Oxford University Press.
- Johnston, C. (1986). *Contour Plots of Large Data Sets*. Computer Language (May 1986), 63-67.
- Kay, D.C., & Levine, J.R., (1992). *Graphics File Formats*. McGraw-Hill.
- Russ, J.C. (1992). *The Image Processing Handbook*. CRC Press.
- Sabin, M. (1986). *Data Representation Graphics: A Survey of Contouring Methods*. Survey paper of a Eurographic (UK) tutorial in the North Holland Computer Graphics Forum 5 (September 1986), pp. 325-340.
- Tufte, E.R. (1983). *The Visual Display of Quantitative Information*. Graphics Press.
- Wolff, R.S., & Yaeger, L. (1993). *Visualization of Natural Phenomena*. Telos/Springer-Verlag.
- Wolfram, S. (1991). *Mathematica: A System for Doing Mathematics by Computer (2nd ed.)*. Addison-Wesley.



Index

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W