Chapter 2 Data in NCSA DataScope

Chapter Overview

NCSA DataScope works with a variety of two-dimensional dataset formats. This chapter defines the requirements for evenly spaced, unevenly spaced, and polar forms with which the program works. The chapter also offers a brief tutorial that explains how to prepare your data in a format compatible with NCSA DataScope.

Two-Dimensional Datasets

NCSA DataScope is designed to work with *two-dimensional arrays* of floating-point numbers. (Unless otherwise designated, the word *array* in this document will refer to two-dimensional arrays.) Once a 2D array is in memory, the data is displayed as printed numbers in a spreadsheet format that allows you to use the application's other commands on this data. NCSA DataScope's working representation of a dataset has the following properties:

- Four byte floating-point numbers are used for all values.
- Each array can be defined in terms of three variables: two independent variables which correspond to the row and column labels, and a dependent variable, which has a value at every point in the array.
- The names of the dependent variable and the independent row and column variables are stored and used when appropriate. The dependent variable name also functions as the name of the window in which the data appears, and is used in notebook calculations.
- The size of the grid, the number of rows (rows), and the number of columns (cols) are stored.
- An array exists for each of the independent variables: one has an entry for each column and defines the spacing between columns of the grid; the other has an entry for each row, and defines the spacing of rows of the grid.
- The values of the dependent variable are located inside the grid, one at each grid point. The number of values is therefore *rows*cols*.
- A format field, given as a FORTRAN specification (e.g., F8.4, E11.5), is maintained for the dependent variable and for the independent variables. When printed as text, the floating-point numbers appear in this format.

Evenly Spaced Grids

In the example given in Figure 2.1, *cols* equals 6 and *rows* equals 4. The grid, therefore, has 24 data points. The row labels array contains {1, 2, 3, 4} and is named Temp. The column labels array contains {1, 2, 3, 4, 5, 6} and is named Time. The dependent variable is named Pressure and must have 24 values, one for each point in the grid. The example highlights the value at (Temp = 2, Time = 3) where Pressure has the value 1.105.

Figure 2.1 Sample Evenly Spaced Dataset

	Time					
Temp	1	2	3	4	6	6
1	0.090	0.090	1.111	1.247	1.247	0.156
2	1.000	1.001	$\boxed{1.105}$	0.156	1.244	1.850
3	1.001	1.104	0.045	1.209	1.621	2.391
4	1.002	1.422	1.233	1.771	2.631	2.753

Pressure=1.105

Unevenly Spaced Grids

In the following example, Figure 2.2, rows and cols remain 4 and 6, but the values associated with the independent variables define a scale which does not have to be linear. In this example, the row labels are almost linear {10.1, 20.3, 30.5, 40.7}, whereas the column labels are almost logarithmic {1.01, 1.23, 25.3, 78.50, 555.40, 3423.11}. In fact, the row and column labels are defined in the HDF file with the scientific data. They may have any set of increasing or decreasing values.

The indicated value in Figure 2.2 is still at grid point (2, 3), but this is shown on the screen as the point where Temp = 20.3 and Time = 25.3. The value of Density at this point is 1.105.

Figure 2.2 Sample Dataset with Nonlinear Row and Column Values

	Time					
Temp	1.01	1.23	25.3	78.50	555.40	3423.
10.1	0.090	1.001	1.105	1.211	1.001	0.045
20.3	1.000	1.105	$\boxed{1.105}$	1.211	1.209	1.000
30.5	1.104	1.000	1.104	1.207	1.209	1.000
40.7	1.104	1.002	1.103	1.103	1.203	1.203

Density=1.105

Polar-Oriented Data

Polar-oriented data differs from rectangular-oriented data in one respect: the value for each row should be treated as a radius value, and the value for each column becomes a theta angle which should always have values in the range 0 to 2π. Polar data can, in fact, be treated as both cartesian *and* polar in so far as the display data is concerned.

File Formats

NCSA DataScope reads floating-point data from HDF or ASCII text files. The following sections describe these file formats and explain the advantages of using HDF files.

HDF Files

HDF files are data files which are created using NCSA's Hierarchical Data Format for file storage. HDF is a flexible, standard file format designed at NCSA for sharing graphical and floating-point data among different programs and machines. It lets you store datasets with extra information about your data. This makes the files easier to use and manage for programs such as NCSA DataScope, and saves you the trouble of tracking this information externally.

Specifically, HDF files can contain all of the data for NCSA DataScope's one dependent and two independent variables, along with the auxiliary data for the dataset. The auxiliary data might include the row and column labels, printing formats, variable names, user notes, and associated images; and is automatically retrieved when you load an HDF file.

For more information regarding HDF, refer to Chapter 6.

Text Files

If you do not have access to HDF libraries on the computer where you generate your data, you can store your data in a text file instead, and then import it into NCSA DataScope. Text files are comprised of ASCII characters. Text files to be read by NCSA DataScope should be created according to the following criteria:

- Real numbers can be in any standard text format for floating-point numbers, but must be separated by spaces, tabs, or carriage returns.
- Integers can have any field width as long as they are separated by spaces or tabs, or carriage returns.
- Maximum and minimum values define the range of number values in the region of interest. *Outliers*, numbers which are larger or smaller than the important portion of the data, are automatically ignored during image generation only, but remain in the dataset unchanged and are not ignored by notebook computations.

- Arrays should be ordered by rows, left to right, top to bottom, depending upon how you want images to be generated.

The format should look like this:

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

The following example shows an array in FORTRAN, declared with dimensions of 5 rows of 10 columns. Although the example shows a situation where the entire row fits on one line of the file, the data values may be split among any number of lines.

For an array that is declared in FORTRAN, for example:

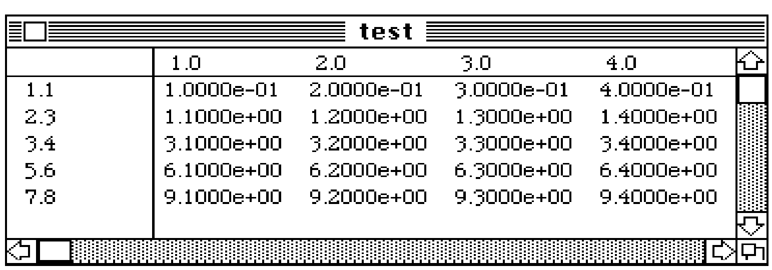
```
REAL*4  MYDATA(5,10)
```

a sample data file might be:

```
5  10
9.9e0  0.0e0
1.1  2.3  3.4  5.6  7.8
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
```

When this data file is saved in ASCII format and loaded into NCSA DataScope using the Load Text command, a text window such as that presented in Figure 2.3 is produced. The particular floating-point display notation is determined by the Display Format settings in the Attributes dialog box (see "The Text Window" in Chapter 3).

Figure 2.3 ASCII Text Window



Loading and Saving HDF Files

To load an HDF file:

1. Choose Open from the File menu or press ⌘-O.

2. Select and open the desired HDF file in the dialog box that appears.

NCSA DataScope checks to verify that you have chosen an HDF file and presents an error message if you have not. The HDF file is then processed and the floating-point dataset is read into memory along with any and all accompanying information. NCSA DataScope creates a text window for the display of the data. If images or notes are contained in the file, it also creates and displays the appropriate image or notebook windows. NCSA DataScope ignores information from the HDF file that is beyond its ability to interpret.

To save a dataset in an HDF file:

1. Select the text window that displays the desired dataset to make it frontmost, or an image or notebook window associated with that dataset. Each new dataset must be saved in a separate file, even if it was created with a calculation from an existing one.

2. Choose Save or Save As from the File menu or press ⌘-S.

The four byte floating-point data is written into the HDF file in binary format as a *scientific data set* (SDS), which includes its row and column independent variable information and formats. The details for storage of an SDS are documented in the *NCSA HDF Calling Interfaces and Utilities* manual.

If the data in the frontmost window has an associated notebook window or image windows which were generated from the data, the Save or Save As command saves into the same HDF file the most recent image of each of the three types and the notebook that is linked to the current dataset. When you load this file, each of these windows is recreated and displays the appropriate contents.

Loading Text Files

To load a text file:

1. Choose Load Text from the File menu or press ⌘-T.

2. Select and open the desired text file in the dialog box that appears.

NCSA DataScope reads the file and creates a text window in which to display the data. Now, you can generate images from this dataset, perform calculations on it, or save it in an HDF file (described in the preceding section).