NEW FEATURES IN THE SDDS TOOLKIT*

R. Soliday, M. Borland, L. Emery, H. Shang,
Argonne National Laboratory, Argonne, IL 60439 USA

Abstract

Self-Describing Data Sets (SDDS) and the corresponding SDDS tools have long been used at the Advanced Photon Source (APS) and other laboratories for data storage and analysis. Various programs in the toolkit have been added or improved recently. Support for the Mac OS X operating system has been added. Automated testing scripts are being used to reduce bugs prior to software upgrades. The Java version of the SDDS Toolkit has been used to integrate standard SDDS functions into MATLAB. The fitting of generic functions to SDDS data has been improved. Conversion of array data to column data has been added to allow analysis of array data with existing programs. The display of sddsplot movies by saving plots to files for rapid playback has been improved.

NEW PORTS

Macintosh OS X

The SDDS Toolkit [1], which was written to store and manipulate accelerator data at the Advanced Photon Source (APS), continues to evolve to meet new requirements at different facilities. The toolkit was previously available on Solaris, Linux, Windows, and VxWorks [2] operating systems. With the introduction of OS X (a variant of Unix) for Macintosh computers, we were able to make SDDS available on Macintosh computers. Since all of the toolkit programs, with the exception of the SDDS plotter, do not require graphics libraries, the port was relatively straightforward. The SDDS plotter uses X11 libraries and thus requires that XDarwin and Lesstif be installed on the system. The SDDS Toolkit source code and OS X binaries are available on the OAG web site, along with the installation instructions.

MATLAB

The Java version of the SDDS Toolkit has been used to add SDDS compatibility to MATLAB. This provides MATLAB users with a stable and reliable way to access SDDS datasets, which, being self-describing and uniform in structure, provide a more organized way of storing data than MATLAB's own data formats. In addition, the SDDS toolkit has data manipulation features similar to a database, which supplements the capabilities of MATLAB. This port also allows SDDS users to take advantage of plotting and analysis features of MATLAB that may be lacking in the SDDS Toolkit.

We created new MATLAB commands that can be used to produce complex plots from data stored in SDDS files. The **sdds3d** command reads an SDDS data file and converts it into a MATLAB 3D mesh including the X, Y, and Z data points and the corresponding labels. This mesh can then be passed to **sddspcolor** to create a flat colored contour plot. These plots can be stacked on top of each other if there are multiple pages of data to be plotted. The 3D mesh can also be passed to **sddssurf** to create a 3D shaded surface plot (see Figure 1). Other new MATLAB commands are **sddscontour**, **sddsmesh**, and **sddsplot3**. These can be used to create contour plots, 3D shaded surface plots with mesh lines, and plots of points in 3D space, respectively.

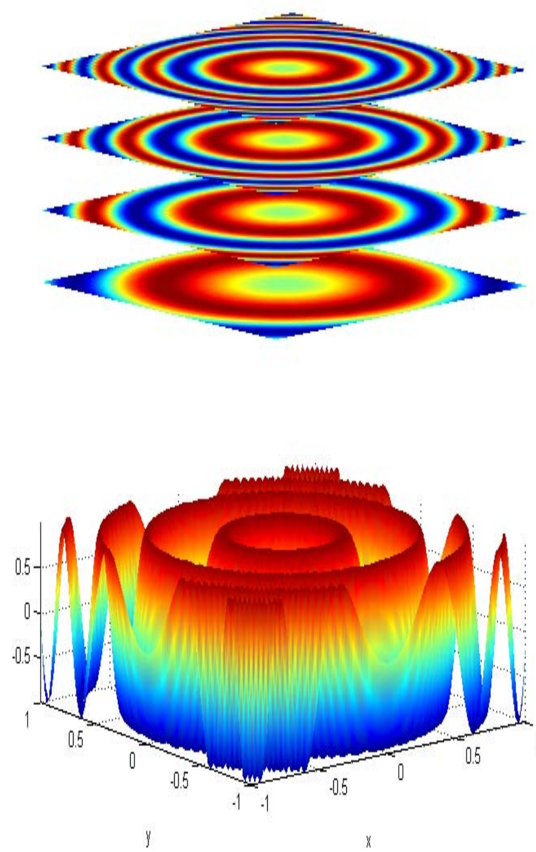

Figure 1: SDDS data plotted in MATLAB.

IMPROVEMENTS

Data Fitting

Fitting of generic functions to data can now be accomplished with **sddsgenericfit**. The user simply provides an arbitrary functional form, the range, and step

* Work is supported by U.S. Departmentof Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

sizes for the variables in the fitting function. The versatility of this program greatly reduces any need to design new fitting software for specific tasks. This will allow the user to create and modify high-level data analysis tools faster since it will not be necessary to create low-level data fitting routines.

Signal Analysis

The new program **sddsnaff** performs Numerical Analysis of Fundamental Frequencies (NAFF), which is Laskar's algorithm for accurate determination of the frequency components of a signal. NAFF is much more accurate than fast Fourier transforms (FFTs). The algorithm starts by removing the average value of the signal and applying a Hanning window. Next, the signal is FFT'd, and the frequency at which the maximum FFT amplitude occurs is found. This provides a starting frequency for numerical optimization of the "overlap" between the signal and $e^{i\omega t}$, thus determining ω to a resolution less than the frequency spacing of the FFT. Once ω is determined, the overlap is subtracted from the original signal and the process is repeated, if desired, to find another frequency component. Output from **sddsnaff** includes the frequency, amplitude, and phase of the frequency components.

Array Data Conversion

The SDDS file protocol allows storing data in both arbitrarily dimensioned arrays and one-dimensional columns of a common length. However, most of the SDDS data analysis tools work on column data. To support array manipulation without altering each program, **sddsarray2column** was created. This program converts SDDS array data to SDDS column data. In addition, it allows taking slices out of multidimensional arrays. The array data to be converted must meet two requirements: the number of elements taken from multiple arrays must be equal to each other and also equal to the number of rows in the data file if columns already exist. While not every SDDS data file with arrays will fit these requirements, **sddsarray2column** has been very useful at the APS and elsewhere.

Image Analysis

In order to develop automated analysis of image data from the LEUTL FEL, we developed the program **sddsspotanalysis**. This program analyzes a series of images to find beam spots. It performs noise reduction and background subtraction, and then determines the centroid, intensity, rms parameters, and degree of saturation. It allowed rapid, automated analysis of thousands of beam images from the FEL.

Frequently it is more efficient to store only the horizontal and vertical profiles of images rather than complete images, so we developed new tools for this case. The program **sddsbaseline** provides various methods of determining and removing baselines from such profiles. **Sddscliptails** provides various methods of clipping the tails from image profiles, where the tail is a dubious feature extending to the right or left of a peak. These programs can be used in sequence to process profiles, which can then be analyzed using **sddsprocess** to determine, for example, rms spot properties.

Along the same lines, a new image converter has been created called **tiff2sdds**. This program converts a TIFF image to an SDDS data file by summing the red, green, and blue color intensities for each pixel and storing the values as SDDS columns.

SDDS Plotter

Many improvements have been made to the SDDS plotting software. A new native Windows graphic interface obviates the need to have an X-windows emulator on Windows platforms. Another new feature of the SDDS plotter is the ability to write the binary plotting instructions to a file. This allows the user to store **sddsplot**-generated movies in a compact file for later replay, even if the original data is unavailable. A related improvement is an interval control in the SDDS plotter that allows changing the playback rate.

Other improvements to **sddsplot** include: the ability to display parameters of any data type on labels; support for wildcards in the independent variable data names and templates for the dependent variable names; support for PNG (portable network graphics) files; addition of a common-offset feature helpful for plotting data from many files at once; and addition of a "spectral" mode for more intuitive color-coding of large amounts of data.

Library and Miscellaneous Improvements

Many improvements have been made to the SDDS libraries. The SDDS library now supports platform-independent binary files, which allows mixing of hardware platforms (e.g., PC and SPARC). It supports synching of files over a network to improve reliability of fast data loggers. One can now work with an unlimited number of active SDDS files even if the operating system has a limit on the number of open files.

A function was added that converts algebraic expressions into reverse polish notation expressions; this has allowed the use of algebraic expressions in programs like **sddsprocess** with only minor coding changes. Data types of elements in SDDS files can now be changed using a new program called **sddscast**. We wrote and released **sddspoly**, a program for polynomial transformation of large quantities of input data. This program is used at APS for translation of BPM set points and offsets between various polynomial sets. The new program **sddsmultihist** allows creating histograms of arbitrary numbers of columns. The new program **sddseventhist** allows histogramming tagged data (e.g., alarms identified by channel name). Savitzky-Golay filters are now available for smoothing in **sddssmooth** and for taking numerical derivatives in **sddsderiv**. A variant of **sddsinterp** called **sddsinterpset** was created to perform multiple polynomial interpolations. This tool simplified calculating set points for individual magnets that have separate magnetic measurement files. Another new tool called **sddsmakedataset** is used to create SDDS files from data entered in the command line arguments. This can save programming effort when a short file is created within a script.

Java

We continued improvement of a Java-based three-dimensional plotting program for SDDS files. It can now plot all the file types that **sddscontour** can plot, as well as performing 3D color-coded scatter plots. Some of the usefulness of this program has been diminished with the release of the SDDS-to-MATLAB interface but it still is the best choice for three-dimensional plotting if MATLAB is unavailable. Another recent improvement to the native Java version of SDDS is the addition of SDDS array support. With this addition the Java version of SDDS is totally compatible with the latest version of SDDS libraries written in C.

STABILITY

Testing Scripts

To help increase stability and reliability, various testing scripts and simulators are routinely used to test and debug SDDS software prior to a release. Two types of storage ring simulators have been used. The first is a portable channel access server that simulates a storage ring orbit. This version does not require much memory but it has been hard to customize to fit new uses. The second is set up as a workstation IOC. This version accepts standard EPICS database files to create process variables and has shown itself to be highly customizable. Both of these simulators include quasi-Gaussian noise on BPMs and correctors. The workstation IOC version of the simulator has been used recently to test and debug the new SDDS waveform programs **sddswget** and **sddswput** [3]. A third simulator is running on vxWorks in an IOC. This simulator was used extensively during the porting and testing of SDDS/ EPICS software [4].

AVAILABILITY

OAG Web Site

The majority of SDDS applications are available though the Operations Analysis Group (OAG) web site (www.aps.anl.gov/asd/oag/software.shtml). Here the source code for the various programs can be downloaded. This code is offered under the terms of the EPICS Open License (www.aps.anl.gov/epics/license/open.php). We also offer precompiled versions for Linux, OS X, and Windows. These packages are offered under the same license with the exception of any packages that are compiled against EPICS 3.14 and contain channel access routines. This subset of packages is offered under the terms of the EPICS BASE License (www.aps.anl.gov/epics/license/base.php).

REFERENCES

[1] M. Borland and L. Emery, "The Self-Describing Data Sets File Protocol and Program Toolkit," Proc. ICALEPCS 1995, Chicago, Illinois, pp. 653-662 (1996).

[2] R. Soliday, "Proliferation of SDDS Support for Various Platforms and Languages," Proc. ICALEPCS 2001, San Jose, California, pp. 545-547 (2002).

[3] H. Shang et al., "New Features in the SDDS-Compliant EPICS Toolkit," these proceedings.

[4] F. Lenkszus et al., "Integration of Orbit Control with Real-Time Feedback," these proceedings.