Open source data analysis and visualization software for optical engineering

Greg A. Smith[a*], Benjamin J. Lewis[a], Michael Palmer[a], Dae Wook Kim[a], Adrian R. Loeff[a], James H. Burge[a]

[a]College of Optical Sciences, University of Arizona, 1630 E. University Blvd., Tucson, AZ 85721

ABSTRACT

SAGUARO is open-source software developed to simplify data assimilation, analysis, and visualization by providing a single framework for disparate data sources from raw hardware measurements to optical simulation output. Developed with a user-friendly graphical interface in the MATLAB™ environment, SAGUARO is intended to be easy for the end-user in search of useful optical information as well as the developer wanting to add new modules and functionalities. We present here the flexibility of the SAGUARO software and discuss how it can be applied to the wider optical engineering community.

Keywords: Data analysis software, data visualization software, MATLAB™-based data processing platform

1. Motivation

Optical engineering frequently involves generating, comparing, and making sense of data from several sources. One fragment of code might read in raw hardware data. Another might perform some data conversion or processing. A third fragment might be used to generate a standard display for a report. Frequently, multiple pieces of software or calculations are involved, each with its own native set of units, sign conventions, and other details that must be considered. Often the fragments of code were originally written by people who are no longer with the organization. Sometimes just combining two pieces of existing code to do a simple calculation or generate a figure for a presentation can be an exercise in frustration if the code is not well documented and assumptions are buried in the details of the code.

Facing the need to simplify and organize software within the Large Optics Fabrication and Testing (LOFT) group at the University of Arizona, SAGUARO[1-3] (Software Analysis Graphical-user-interface from University of Arizona for Research in Optics) was developed and is available to the public as non-profit freeware at:

http://www.loft.optics.arizona.edu/saguaro/

In this article we briefly introduce SAGUARO in section 2, describing the core framework. Sections 3 and 4 contain examples demonstrating optical surface quality analysis and working with finite element analysis data.

2. SAGUARO

The goals of SAGUARO are:

- Simplify common data analysis and visualization tasks within the optical engineering community
- Create a flexible framework that can easily be extended to meet changing needs
- Facilitate optical engineering community involvement in developing interoperability
- Develop an intuitive user interface that is useful to both technicians as well as optical scientists and engineers

SAGUARO is built on the concept of modularization and has nearly 30 standard modules such as Zernike fitting, spatial frequency filtering and plotting routines[3]. Developed with a graphical interface in the MATLAB™ environment[4], the SAGUARO kernel connects the modules into a cohesive program.

* Correspondence should be addressed to the SAGUARO team email address: SAGUARO@optics.arizona.edu

Novel Optical Systems Design and Optimization XV, edited by G. Groot Gregory, Arthur J. Davis,
Proc. of SPIE Vol. 8487, 84870F · © 2012 SPIE CCC code: 0277-786X/12/$18 · doi: 10.1117/12.929040

2.1. Organization

The SAGUARO release package is a simple directory structure that can be placed anywhere on a computer. In the base directory, the SAGUARO.m and SAGUARO.cmd scripts start the SAGUARO program from within MATLAB™ or the Windows™ environment†.

The remainder of SAGUARO is organized into the directories described in Table 1.

Directory	Description
Kernel	The set of routines which combines SAGUARO code into a functioning program are contained in this directory. Many files in this directory are MATLAB™ protected to avoid fragmentation of SAGUARO core functionality.
StandardModules	Modules in this directory come standard with the SAGUARO release and contain all scientific processing. They are open-source accessible for review and editing as needed.
UserModules	Location for user-generated modules. By placing your scripts here, upgrading will be easier since all custom routines can be copied to the new installation location.
Data	Although SAGUARO can load data from anywhere, this directory is the default location for storing/retrieving data sets. It contains a few sample data sets.
Templates	Example data sets and modules are located in this directory. Developers with MATLAB™ scripts they wish to incorporate into SAGUARO should consult this directory for examples and templates demonstrating how SAGUARO functions.
Documentation	Supplementing online video tutorials, this directory provides documents such as a "Quick Start Guide", details about data types, and a description of version changes.

Table 1: SAGUARO directories and description of folder contents

2.2. Standard Data Types

Data in SAGUARO is stored in data sets, each of which corresponds to one of the seven data types described in Table 2. These are built around common optical engineering data types.

Data type	File extension	Data format	Description
Map	.map	2D matrix [a × b]	Data that exists as a matrix and models a surface, such as an OPD map
Mask	.mask	2D matrix [a × b]	Matrix of 1s and NaNs describing regions of interest
Frequency Map	.freqmap	2D matrix [a × b]	Frequency components of Map data (e.g. Fourier transform of a map)
Zernike Coefficients	.zernike	2D matrix [a × 3]	1st column: Zernike coefficient, 2nd column: n, 3rd column: m (where Z_m^n)
Layer Map	.layermap	3D matrix [a × b × c]	A series of map data types
Coordinates	.coordinates	2D matrix [a × b]	Each column may represent any list of data (e.g. 1st column: x, 2nd column: y, 3rd column: z)
General	.general	arbitrary	Arbitrary user-defined data type

Table 2: SAGUARO data type information including file extension, data format, and brief description

Additional information about the data types can be found in the documentation that is provided with the SAGUARO release[1].

† Although SAGUARO can be started directly from within Windows™ using the command script, MATLAB™ is still required to be installed.

Most data types require units specified by the user. Conversion between many common optical engineering units is accomplished with the 'UnitConvert' module, which only offers units that are logically related. For example, heights in a data set specified in inches can be converted to many other length units such as angstroms, meters feet or miles but not milliwatts or hertz.

2.3. Documentation

SAGUARO comes with several sources of documentation including a "Quick Start Guide" which describes the installation process and provides a walk-through example. Basic help for each module is accessed through the "Module Help" button on the Graphical User Interface (GUI). Detailed examples of data types and module construction are available for module developers in the "Templates" directory.

Additionally, we have a section on our webpage called "SAGUARO 101" which contains video tutorials on how to use SAGUARO. We also have a forum called the "SAGUARO Garden" where people can ask questions, interact, and share modules:

http://www.loft.optics.arizona.edu/saguaro/saguaro-101/
http://www.loft.optics.arizona.edu/saguaro/saguaro-garden/

3. Example – Analyzing Optical Surface Quality

To illustrate the potential for SAGUARO to speed up optical engineering applications, a common example from the Large Optics Fabrication and Test group is provided. The scenario is the polishing of a large mirror through an iterative process. After polishing for a length of time, the optic will be measured and evaluated to determine how to fine-tune the motions of the polishing machine in order to improve the surface figure in the next iteration of polishing. Because this sequence is often repeated several times, many of the SAGUARO features discussed in this section can be stored as a macro. Macros enable sequences of modules to be run in a pre-determined sequence and use the last-entered module parameters as default values to help speed repetitive tasks. The entire example can be run in less than 1 minute.

3.1. Import Surface Data from Interferometer

During a polishing run, an interferometer is used to acquire a surface image of the optic being manufactured in order to measure progress. Some simple analysis can be done using the commercial software provided with the interferometer, but the capabilities of the software are often insufficient for a specific task and because the software is frequently proprietary, it cannot be easily modified. After a basic review of the data using the commercial software, the surface information is exported to a text file.

Upon starting SAGUARO, we click to load a new data set and select the text file. We get a warning that the data is not a standard SAGUARO data type and the surface map is added to SAGUARO as a "general" data type. This is a typical operation when loading raw data.

One module available for general data types is GenDataConvert. Reviewing the module help shows that this module will allow us to transform from the general data type to something more suitable. In this case a "map" data type best reflects the type of information we have.

Executing the module brings up several dialog boxes asking about the data type to convert to (map), the dimensions of the data (1m diameter with height in µm), and which data values should be considered as background values. These values are parameters of our measurement and how it should be processed. After entering these parameters, the header and preview areas (illustrated in Figure 1) show successful conversion to a map data type.

Header

	Name	Value
	Data Type	map
	Date Created	07/04/12 10:28 AM
	X size	1000
	Y size	1000
	X unit	mm
	Y unit	mm
	Z unit	µm

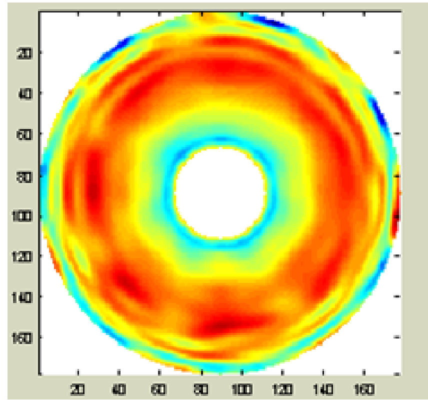


Figure 1: Raw data from an interferometer read into SAGUARO and converted to a "map" data type via the GenDataConvert module. Header information (left) and preview of loaded surface data (right).

3.2. Zernike Fitting

With the data in SAGUARO as a map, many modules are available for processing. To start, we want to view the low order Zernike polynomials to get information on the basic shape of the optic.

One way to compute the Zernike values of the map is with the "Map2Zernike" module. Reading the module help, it becomes clear the module works using the Zemax™ ordering of Zernike coefficients. We are interested in only the low order terms, and we choose to fit only up to primary spherical (Z11).

On executing the module, several dialog boxes appear which are specific to the module calculation. After calculating, the Zernike values are shown in the "data area" of the SAGUARO screen as illustrated in Figure 2.

	Z_m^n	n	m
1	2.2203e-05	0	0
2	-1.6681e-06	1	1
3	1.1083e-05	-1	1
4	5.0152e-06	0	2
5	3.9387e-06	-2	2
6	-5.9817e-06	2	2
7	1.5984e-05	-1	3
8	-2.4342e-06	1	3
9	7.1955e-06	-3	3
10	3.3089e-05	3	3
11	-0.0011	0	4

Figure 2: Data area output from the "Map2Zernike" module showing the 11 lowest-order Zernike terms.

Reading the table, it can be seen that although the Zernike terms were specified using Zemax™ ordering, SAGUARO has converted them to a more standard subscript/superscript notation. In this example, most terms are very small, but Z_4^0 (i.e. Z11) is -0.0011µm or in other words, there is -1.1nm of primary spherical.

Because SAGUARO modules are written by many individuals, the conventions used within a module may vary, but the input and output data types will always follow SAGUARO standards. Modules are nothing more than MATLAB™ functions that perform the desired scientific computation. The only requirement of a module is that the output data be in the correct format for the data type. As needed, the module help information accessible through the GUI provides module-specific information necessary to run the module.

3.3. Low-Order Zernike Filtering

Visualizing the low-order Zernike terms is accomplished by running the "Zernike2Map" module using the Zernike values calculated in the previous step. This module provides the option to remove any Zernike terms, and in this case we choose to remove piston, tip, and tilt terms (Z1-Z3) which are more likely to reflect our measurement setup than any actual surface feature. The surface is shown in the SAGUARO preview window and in Figure 3.

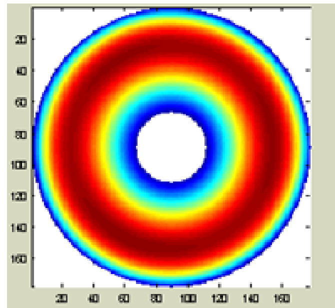


Figure 3: Surface generated from low order Zernike terms using the "Zernike2Map" module.

As expected based on the Zernike table output, the primary spherical term dominates the image.

3.4. Mid-Spatial-Frequency Components

Investigating mid-spatial-frequency components of the surface error map is important for guiding computer controlled polishing runs as the surface specification is often given in terms of structure function or power spectral density[5]. To generate a map of the mid-spatial-frequency components, the above process could be used with different Zernike orders. Instead, we choose to subtract the low-order Zernike image from the original image as illustrated in Figure 4. This is performed using the "MapCombine" module which handles simple arithmetic operations on two maps.

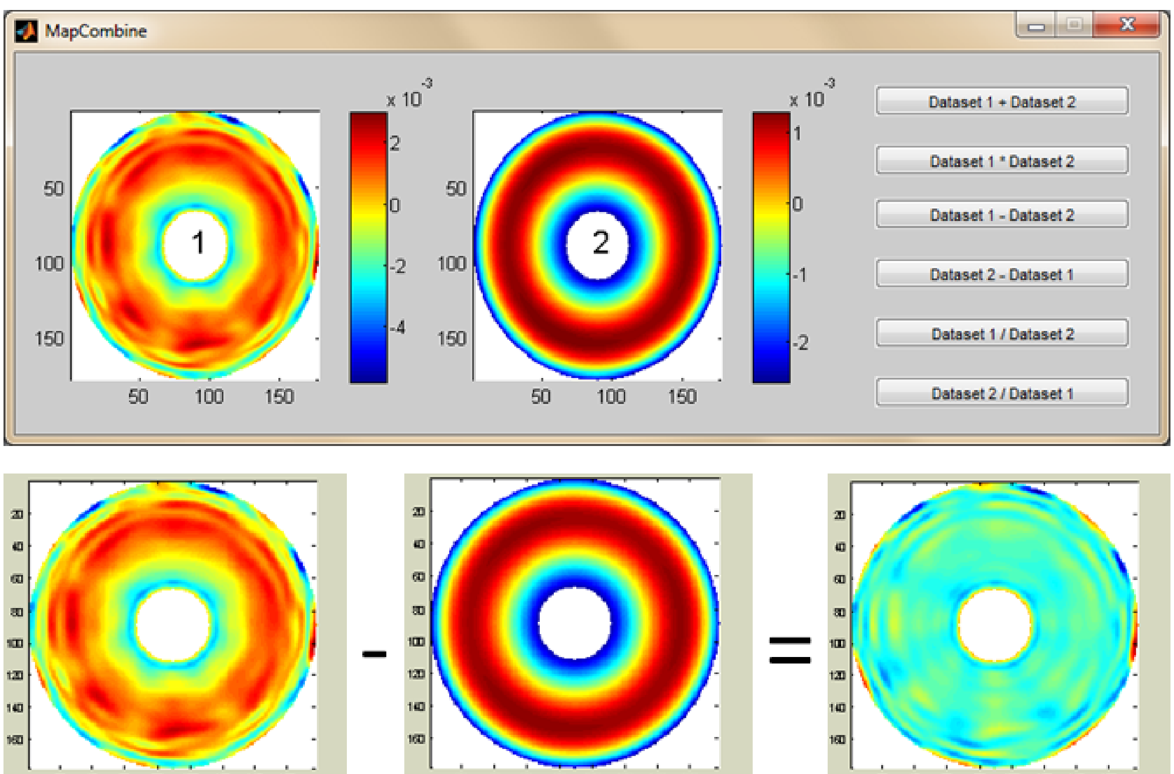


Figure 4: The "MapCombine" module allows the low order Zernike map to be subtracted from the original map data set to give the mid-spatial-frequency components. MapCombine module dialog box (top) and summary of calculation (bottom).

SAGUARO is designed to be context aware. This means that only modules which are appropriate for the data type will be made available. This also means that modules which require multiple inputs will only be available when appropriate multiple data sets are selected by Ctrl-clicking or Shift-clicking within the data set area. The data sets can come from different raw data or from various stages of processing a single data set. As long as the chosen data sets have the correct data types for the module, they can be used in SAGUARO.

3.5. Surface Statistics and Presentation

The final action we wish to perform on our data is to get some information on the residual mid-spatial-frequency components of our surface to evaluate surface figure and add to a status report. For this, the "MapStatistics" module is run on the error map data. The output is shown in Figure 5.

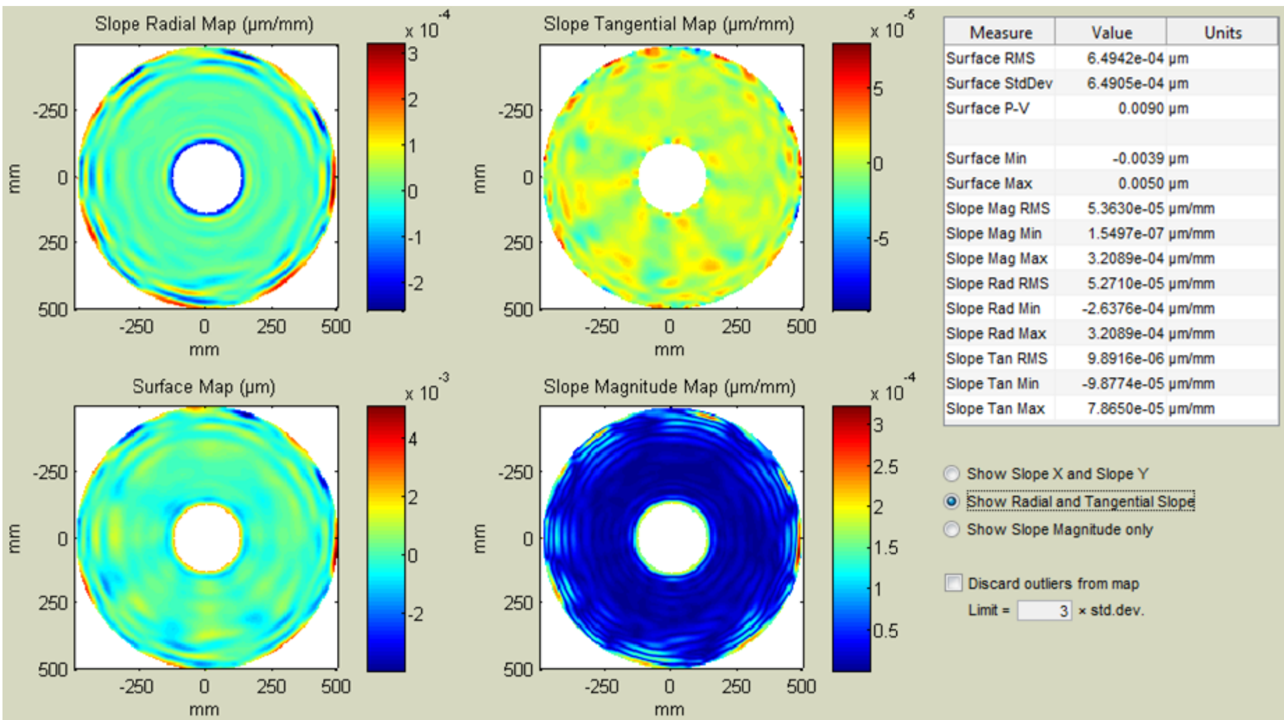


Figure 5: "MapStatistics" module output for the mid-spatial-frequency components of the surface error map. Surface and slope statistics are computed and displayed.

Several optical engineering statistics and images are displayed. In addition to surface peak-to-valley and RMS values, surface slope statistics are displayed in either Cartesian or polar coordinates. Maps showing slope variations across the surface will be helpful in refining the parameters for the next round of polishing runs.

For the example demonstrated here, the mid-spatial-frequency surface has less than 1nm rms variation which meets the target goal for this optic. Using the "MapPlot" module, we can go back and tweak earlier images before saving them for display in a report, or we could use the "Add Globals" button to export all the data sets to the MATLAB™ environment to run additional processes that have not yet been made into SAGUARO modules.

4. Example – Loading Finite Element Analysis Data in SAGUARO

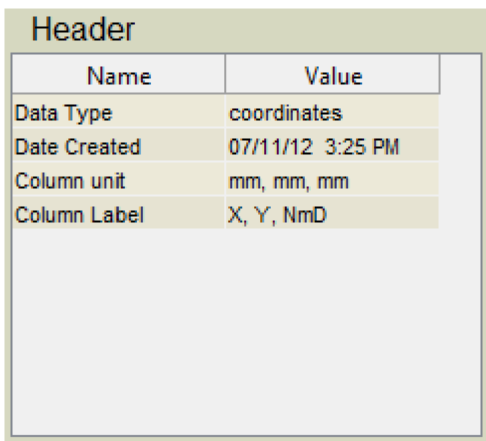
Many modules in SAGUARO are written for map data types, but not all raw data is generated in the nice rectangular grid that the map data type requires. A common case where this occurs is when working with Finite-Element-Analysis (FEA) results in opto-mechanics. Typically, this data is provided as a set of Cartesian x/y/z coordinates where the x/y values are placed at irregularly spaced nodes needed for the FEA model.

In this example, we take a set of irregularly spaced FEA coordinates and convert them to a map data set. This permits access to the many modules available for map data types. In SAGUARO, this can be done with only a few clicks or even run as a macro.

4.1. Loading Coordinates Data

As in section 3.1, the process begins by loading raw data into SAGUARO. For this example, we have computed the surface displacement due to a three-point mount using an FEA program. Because we frequently do this type of analysis, a custom SAGUARO module has been created in the 'UserModules' directory to parse the raw FEA data and bring it into SAGUARO as a standard data type. Because the FEA grid is irregular, the raw data is naturally described using the "coordinates" data type which contains a list of Cartesian x/y position values along with surface displacement values.

Upon loading, the header values shown in Figure 6 verify conversion to the coordinates data type. All units are in millimeters and labels of 'X', 'Y' and 'NmD' have been chosen to represent the two position axes and the third column of normal displacement values.


Header

Name	Value
Data Type	coordinates
Date Created	07/11/12 3:25 PM
Column unit	mm, mm, mm
Column Label	X, Y, NmD

Figure 6: SAGUARO header information after loading FEA coordinates data through a custom SAGUARO module.

4.2. Displaying Surface

To verify the data has been loaded correctly, we first view it. The simple preview box in the SAGUARO GUI only shows the raw data as three columns of values as seen on the left side of Figure 7. The display is a nearly meaningless set of colors. Using the "CoordinatesPlot" module allows us to view the data in a more typical format as shown on the right side of Figure **7**.

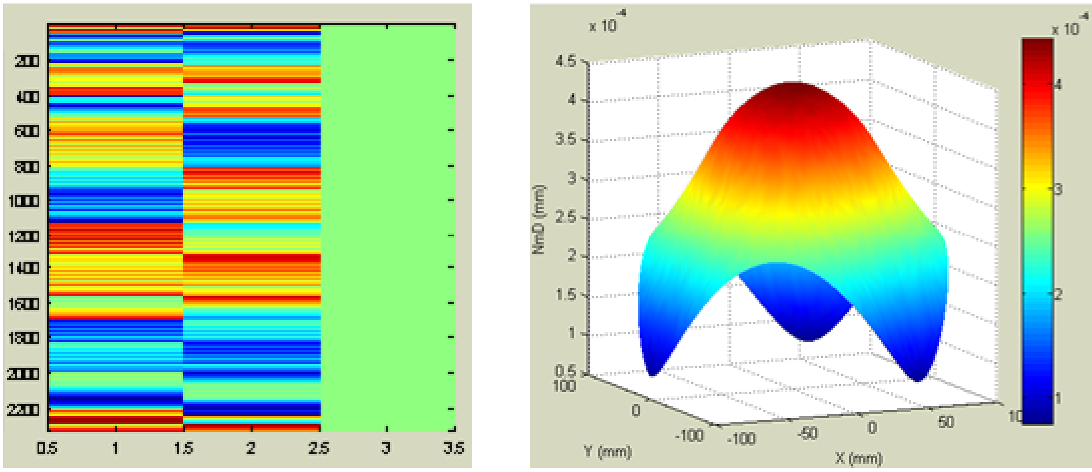


Figure 7: Display of a "coordinates" data set. Preview window in SAGUARO only shows raw data in columns (left). "CoordinatesPlot" module allows the data to be visualized as a surface (right).

The CoordinatesPlot module assumes Cartesian data of the form x/y/z and generates a surface of the data. In this example, the data is displayed as surface plot without the markers. The result is a 3D graphic of the surface which shows the surface displacement around the three mounting points.

4.3. Converting to Map

Converting the coordinates data to the map type is accomplished through the "Coordinates2Map" module. This module will interpolate and resample the data to produce a regularly-spaced set of data points on a grid. When executing the module, it notices the label on the Z axis has been changed to "NmD" (shorthand for "normal displacement") and warns of possible errors. Since the data is in the correct x/y/z format, the message can be ignored.

The next dialog box asks for size information, both in terms of the actual size in mm and the number of pixels the output map type should contain. Our data set varies from -100mm to +100mm in each direction. If we specify larger values, the Coordinates2Map module will attempt to extrapolate the data. Choosing smaller values will crop the data. In this example we choose a size of 200mm to capture the entire image.

The module then interpolates the data and constructs the map data type. Results from the header area and preview area after conversion are shown in Figure 8.

Header

Name	Value
Data Type	map
Date Created	07/12/12 7:37 AM
X size	200
Y size	200
X unit	mm
Y unit	mm
Z unit	mm

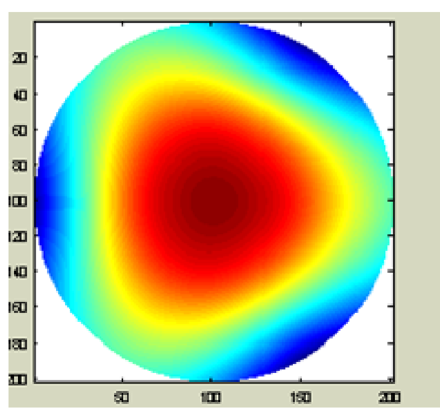


Figure 8: Output of Coordinates2Map module showing correct conversion. Header (left) shows map data type of expected sizes; preview image (right) agrees with the coordinates plot.

At this point, the original coordinates data has been converted into the map data type. The preview shows the expected behavior of a round optic, with a large value in the center and the three-fold symmetry from the three mounting points clearly visible. Now that the data is in the map data type, all the modules available for that type are at our disposal.

Moving between data types is fairly common, especially when importing raw data. Several modules in SAGUARO such as the "Coordinates2Map" and "Zernike2Map" modules are available for switching between data types. This example was based on the concept of irregularly spaced finite element analysis data, and with a few clicks it was converted to a regular grid of data which is more suited to complicated processing routines.

5. Resources

Additional resources can be found on the SAGUARO website[1] including links to the latest version of the code, a forum where users can share modules, and documentation about SAGUARO including short video tutorials about getting started with SAGUARO and developing modules for SAGUARO. Email sent to the contact author of this article will be answered by the SAGUARO development team.

6. Conclusion

SAGUARO, presented by the LOFT group at the University of Arizona, is designed to simplify optical engineering tasks by providing a user-friendly yet flexible platform that permits easy interoperability between code modules. SAGUARO is intended to help the optics community deal with the headaches that come from maintaining code and to make software more manageable.

In this article, we showed how SAGUARO can streamline data analysis and processing typical of the needs of the optics community. An overview of the core framework showed the simple file structure which helps organize data, and the base SAGUARO data types that reflect common optical engineering applications. Documentation in several forms is also available to make SAGUARO easy to use and to assist with development of modules. Examples for processing raw surface data and basic finite element analysis results were provided. Both examples can convert raw data to useful analysis results in less than a minute.

7. Acknowledgements

We would like to thank Won Hyun Park for providing the example finite element analysis data used in section 4 of this article. Additionally, we would like to acknowledge and express our appreciation to all the people who have contributed comments and suggestions during the beta period and those who have contributed SAGUARO modules.

REFERENCES

[1] For the latest copy of SAGUARO from the Large Optics Fabrication and Testing (LOFT) group, visit: http://www.loft.optics.arizona.edu/saguaro/.
[2] Smith G. A., Lewis B. J., Kim D. W., Palmer M., Loeff A. R., and Burge J. H., "SAGUARO: Data Analysis Software for Optical Engineering", in Optical Fabrication and Testing, Optical Society of America, 2012, paper OTu1D.7.
[3] Dae Wook Kim, Benjamin J. Lewis and James H. Burge, "Open-source data analysis and visualization software platform: SAGUARO", Proc. SPIE 8126, 81260B (2011).
[4] http://www.mathworks.com/products/matlab/.
[5] Dae Wook Kim, Hubert Martin, and James H. Burge, "Control of Mid-spatial-frequency Errors for Large Steep Aspheric Surfaces", in Optical Fabrication and Testing (OF&T) Technical Digest (Optical Society of America, Washington, DC), OM4D.1 (2012).