Learning and Practicing Interactive Data Language (IDL) to Manipulate Scientific Data

Jamika Baltrop, Wanda-Marie Carey, Brittnei Teasley, Chelsea Vick
CERSER
1704 Weeksville Road, Box #672
Elizabeth City, North Carolina 27909

Abstract-The Center for Remote Sensing of Ice Sheets (CRESIS) headed at the University of Kansas (KU) was funded by the National Science Foundation (NSF) to explore the Polar Regions (Greenland and Antarctica), and research the various changes occurring with ice sheets. CReSIS uses various types of radar to analyze ice sheet data. Researchers use radar to probe the ice sheets to get huge amounts of data- Synthetic Aperture Radar (SAR) data. SAR data contains more information on the ice sheets for discovery. During the International Polar Year (IPY 2007-2009), luminous data sets will be obtained- Terabytes of data will be in just one field campaign. This means large storage devices and fast computers (multi-core) will be needed to process the data sets and retrieve results in a timely manner, which will outstrip the current capacity of the grids of storage and computers.

In response, Indiana University (IU), Elizabeth City State University (ECSU), and the University of Kansas (KU) initiated a Polar Grid project for the purpose to set up a Cyber – Infrastructure for Remote Sensing of Ice Sheets. This Grid will consist of the state-of-the-art computers and storage hardware, and also application/processing tools, and scientific gateways for the Polar Science Community to conveniently access the resources. It is important, too, to educate and train the researches, educators, and students for polar science. The Center of Excellence in Remote Sensing Education and Research (CERSER) of ECSU has committed to engage the students and train them for polar science with hand-on practices and skills for future study, research and career dedications to the polar science field.

To provide support for Polar data collection, an advanced scientific programming and visualization environment will be used to develop interfaces for computation and visualization- computer-intensive tasks such as in big array operations. In this project, Interactive Data Language (IDL) was investigated as the package for efficient and convenient data visualization capacities in the forms of graphics, images and photographs. 2D and 3D images require intensive computation and efficient visualization, which are crucial for the Polar Grid project.

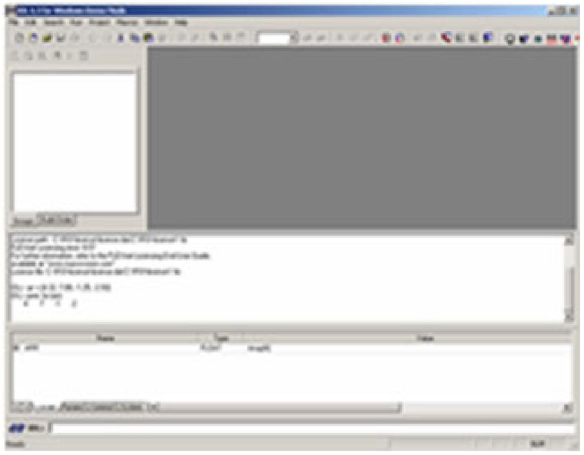

Figure 1. The Screen shot of an IDL program.

The project involved learning the IDL language and environment. IDL is an array-oriented data analysis and visualization application, which is widely used in research, commerce, and education. Its application areas include engineering, medical physics, astronomical, space, and earth science. It offers rapid interactive data analysis and visualization, a programming environment, and end user applications. IDL is available for Windows, UNIX, Linux, Macintosh and VMS platforms and Operating Systems. The high availability facilitates data analysis and visualization in multi-platform environment, and ensures high code portability among platforms and systems.

I. INTRODUCTION

Interactive Data Language provides a host of tools which aid in scientific calculations and analysis. The general purpose of IDL is it is a scientific computing package that provides mathematical functions, animation and scientific visualization tools. Once being introduced to the basics of IDL, it will be maps and charts that can assists in classrooms as well as outside of the classroom

II. IDL MODES

There are two modes that we used and focused in on in our research: interactive mode and compiled mode. Together these two modes allowed for quick and efficient results when writing and executing programs and various other mathematical functions.

A.Interactive Mode

Interactive mode allowed us to rapidly visualize and analyze data concisely using single-line commands. Commands typed in the command input window were immediately executed by the press of the 'Enter Key'. This mode also allowed for us to see quick results of our work in the form of an image, plot, or other graphical representation without the usual edit/compile/execute model used by languages, such as C++ or FORTRAN.

B.Compiled Mode

Compiled mode is the where sequences of IDL commands are compiled, edited, and executed. We used the compiled mode to create programs, where the input commands that are larger than ten lines that could not be created using interactive mode.

III. ARITHIMETIC FUNCTIONS

Mathematically, IDL assists in a number of ways including the simplest forms of math and arithmetic functions. Assigning a variable to a desired number or mathematic function was only the beginning. These variables will later be used to perform operations that present answers to an equation. The matrix was introduced as a format in which equations are calculated. Braces will be used to define vectors, or 1-dimensional arrays, which we then used within a matrix. The matrix can assist in multiplying a large array of numbers. When the output is printed, the first index corresponds to the column, and the second index to the row. Furthermore, vectors can be multiplied together or by a matrix.

A.Creating a Matrix

Matrices can be explicitly constructed from vectors. The vectors are first defined using numbers enclosed in square braces. For example, we assigned vectors v1, v2, and v3 to be

```
v1 = [1, 2, 0]
v2 = [1, 0, 0]
v3 = [4, 5, 6].
```

The three vectors are each inputted as single-line commands in the Command Input Window and displayed in the Output Log Window. We then assigned the three vectors to any variable (letter) such that

```
A = [[v1], [v2], [v3]]
```

is also display in the output log window. The entire matrix can now be viewed using the command: 'print, A' which yields the result

```
1    2    0
1    0    0
4    5    6.
```

B. Multiplying a Vector by a Matrix

Perhaps the simplest task of computing arithmetic functions in IDL is multiplying a vector by a matrix. As in the previous section we defined a matrix A to be [[1, 2, 0], [1, 0, 0], [4, 5, 6]]. The next step is assigning a vector numbers for which the matrix was multiplied by, whereas

```
v = [1, 2, 3].
```

Subsequently, matrix A is again displayed by use of the command 'print, A', as well as the vector being 'print, v' outputting the result

```
1    2    3.
```

Multiplying the vector by the matrix, the command 'print, v ## A'(the pound sign meaning multiplication) now displays

```
15    17    18
```

as being the outcome of the problem.

C. Compiling IDL Commands into Programs

The advantages of IDL programming is the simplicity of compiling single-commands into a program. We used the compiled mode to verify correct results when inputting the IDL commands, or for larger sets of commands. We began by retyping the entire IDL commands in the Editor Window using PRO (function) and the file name (MATRIX) as the header and END (closing function), resulting

```
PRO MATRIX
v = [1, 2, 3]
print, A
print, v
print, v ## A

END.
```

The program was then saved on the local hard drive of the computers we were using. To run and execute the program we typed the path name

```
C:\RSI\IDL63\matrix.pro
```

in the command input line, followed by the command '.compile matrix.pro' which again displayed the output of the matrix?

IV. IMAGING

Possibly the most significant function of IDL is imaging. This could be used within any mapping or reconstruction of a certain area. In particular, the team zoomed in on a section of New York and improved the visual clarity, or resolution of the map. We also made the pixels on the map of New York a value of 140 into a full range of brightness. We made the contrast of the enhancements on the picture as well. We set the minimum brightness to 140, the maximum brightness to a scale of 200 and the image will display a brighter picture. Smoothing and Sharpening the picture was also an area we touched. Unsharp masking made the picture a bit blurry because it contained low frequency to the original image. Sharpening Images with differentiation functions made a more sharpen image.

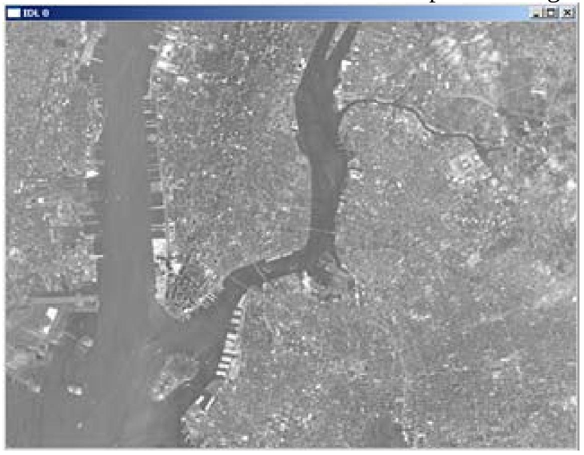

Figure II. Image of Manhattan, New York imported into IDL in TIFF format.

V. FUTURE WORK

Due to the limited time in researching and carrying out our project, our goal is to continue to work on creating a Cyber-Infrastructure for the Remote Sensing of Ice Sheets and establishing a grid for the usage of advanced computing,processing and storage. The task ahead is to produce two interfaces:

1) One interface will be used to demonstrate how to perform computations using an array.

2) The other interface will be used to visualize the data in the form of a graphic image or photograph.

The interface will use the library of functions Application Programming Interface (API) that will be used in conjunction with computer programs such as IDL to build applications. An example would be is when you open your e-mail box instead of typing a series of commands, you can simply use the menus and icons to send and retrieve you mail.

REFERENCES

[1] University of Minnesota-Supercomputing Institute tutorial http://www.msi.umn.edu/software/idl/tutorial/idl-images.html

[2] Boston University-Scientific and Visualization (SCV) tutorial http://scv.bu.edu/documentation/tutorials/IDL/idl_webtut.html