

Project Paper

For this semester project we were tasked with completing several programming tasks in a new language that we had not previously had experience with in our classroom instruction and the language that I choose to tackle these problems was matlab. Over the course of this paper I will cover the history of Matlab, it's primary features, and my experience coding with it.

Matlab's development started in the late 1970s with the goal access to linear algebra libraries LINPACK and EISPACK without the need of learning the language Fortran and quickly found a following in the applied mathematics community. In 1984 the language was rewritten in C and Mathworks was founded with the rewritten libraries now known as jackpack. Finally, in 2000 matlab was again rewritten to use a new set of libraries for matrix manipulation called LAPACK and this is how the language remains today.

Matlab has a variety of features with the most noteworthy being cross compatibility/communication with the operating system and other languages, GUI creation ,matrix/array features, and type conversion. Matlab's compatibility/communication features come from the fact that it can directly call functions and subroutines in C or Fortran, it can directly call libraries that are written in Perl, Java, ActiveX, and NET, matlab code can also be called from Java using the Matlab toolbox, matlab can send command line/ bash commands using system(), and the matlab compiler allows matlab programs to be run as standalone applications for a variety of platforms and even as excel add ins. Most of the array/matrix features in matlab comes from the flexibility given to the programmer as arrays/matrices do not have a fixed length or even fixed number of dimensions and at the same time most built in matlab functions and operators like mod() or * for example also support matrices/arrays and will

apply the operation to each element of the matrix/array so this combined with the larger number of built in functions for matrices/arrays than other languages allows matlab to be a very powerful language for working with arrays and matrices with many built in options to manipulate them. Matlab's type conversion strengths comes from the weakly typed nature of the language and when mismatched types are used in a == or an array/matrix for instance the language will attempt to convert between the two in some predefined valid way but will give a error if it cannot, but at the same time explicit type casting is also possible to convert from one type to another and this gives the programmer a fair number of options for typing in matlab. Finally, the intuitive GUI creation comes from a built in tool in the Matlab suite called GUIDE which permits guis to be built in a visually interactive way and then manipulated via actually programming behaviors for the elements with the language and this allows GUI creation to be very simple with matlab.

For most of the problems in the project the built in capabilities in matlab actually made things relatively easy or at least similar in difficulty to other languages for every problem except for problem 4 where it was more difficult than in some other languages. Problems one and three really demonstrated some of the strengths of the language and its built in functions as I could just simply open a file with fopen and read in all of the contents with textscan for file IO and this made sorting easy to do with a while loop and separate function for finding ascii total for problem 3 and even more simple for problem 1 since I could just use the built in unique() function to find the unique words in the scanned file contents. For problems 2 and 6 the language did not hinder me in any way since problem 2 was just a basic while loop where I had to prompt the user every time which could be done with the input() function and problem 6 was just some

basic recursion and since the language natively supports recursion and I already knew the base case and recursive case this was also no more difficult than doing it in any other language I have used. Problem 5 was a unusual one because it involved using a 2d array and finding the row and col averages of a matrix and while the language have robust native support for matrices that are created in a very similar fashion to other languages the way they are indexed using `mat(index, dimension)` confused me for a bit but once I understood the syntax it was not any more difficult than other languages but it was a little unintuitive at first coming from mostly C based languages. Finally, problem 4 really demonstrated some of the limitations of matlab since it required the ability to read user input from the linux terminal without halting execution until a key is entered and this is not possible natively in matlab since most methods to do this require a GUI and multithreading + event based programming has limited support at best I had to use a workaround using `system()` in matlab to call the bash `read` command to listen for a keystroke for one second and store the output of the command in a input variable.

Overall, I found my time working with this new language to be a very interesting experience; while I found the syntax to be relatively straight forwards, the object oriented like implementation of functions to be intuitive, and I even found myself appreciating the robustness of the built in functions and libraries for file io/arrays/matrices, the support for event based programming and or multithreading certainly dragged on the experience a bit, but despite that my opinion of the language is overall positive since it presented me with fairly little restrictions on my ability to perform the tasks I had to do.

Sources(MLA format)

1. “Array Indexing.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/learn_matlab/array-indexing.html.
2. “Cast.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/cast.html.
3. “Compiler.” *MATLAB & Simulink*, Mathworks, www.mathworks.com/help/compiler/.
4. “Control Flow.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/control-flow.html.
5. “Disp.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/disp.html.
6. “File IO.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/low-level-file-i-o.html.
7. Garrels, Machtelt. “8.2. Catching User Input.” *Nested Loops*, The Linux Documentation Project, Dec. 2008, tldp.org/LDP/Bash-Beginners-Guide/html/sect_08_02.html.
8. Garrison, David. “Writing Apps in MATLAB.” *MATLAB & Simulink*, Mathworks, 2012,
www.mathworks.com/company/newsletters/articles/writing-apps-in-matlab.html.
9. “Matrices and Arrays.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html.
10. Moler, Cleve. “The Origins of MATLAB.” *MATLAB & Simulink*, Mathworks, 2004,
www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html.
11. “Prompt.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/input.html.

12. “Sort.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/sort.html.
13. “Textscan.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/textscan.html.
14. Tutorials Point. “MATLAB Functions.” *Www.tutorialspoint.com*, Tutorials Point, 19
Mar. 2018, www.tutorialspoint.com/matlab/matlab_functions.htm.
15. “Unique.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/unique.html.
16. “Calling External Functions.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/calling-external-functions.html.
17. “APIs for External Languages.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/programming-interfaces-for-external-languages.html.
18. “Perlfile.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/help/matlab/ref/perl.html.
19. “MATLAB Compiler SDK.” *MATLAB & Simulink*, Mathworks,
www.mathworks.com/products/matlab-compiler-sdk.html.

Sources(Direct links)

http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_08_02.html

https://www.mathworks.com/help/matlab/learn_matlab/array-indexing.html

https://www.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html

https://www.tutorialspoint.com/matlab/matlab_functions.htm

<https://www.mathworks.com/help/matlab/ref/cast.html>

<https://www.mathworks.com/help/matlab/ref/unique.html>

<https://www.mathworks.com/help/matlab/ref/input.html>

<https://www.mathworks.com/help/matlab/low-level-file-i-o.html>

<https://www.mathworks.com/help/matlab/ref/disp.html>

<https://www.mathworks.com/help/matlab/ref/sort.html>

<https://www.mathworks.com/help/matlab/ref/textscan.html>

<https://www.mathworks.com/help/matlab/control-flow.html>

<https://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html>

<https://www.mathworks.com/help/compiler/>

<https://www.mathworks.com/company/newsletters/articles/writing-apps-in-matlab.html>

<https://www.mathworks.com/help/matlab/programming-interfaces-for-external-languages.html>

<https://www.mathworks.com/help/matlab/calling-external-functions.html>

<https://www.mathworks.com/help/matlab/ref/perl.html>

<https://www.mathworks.com/products/matlab-compiler-sdk.html>