**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **4-July-2020** | **Name:** | **Raziya Banu** |
| **Course:** | **Matlab** | **USN:** | **4AL16EC058** |
| **Topic:** | **Overview on Matlab** | **Semester & Section:** | **8th sem & ‘B’ section** |
| **Github Repository:** |  |  |  |

|  |
| --- |
| **Image of session** |
| **Report –**  In my first session today I have studied about – **Overview on Matlab** ABSTRACT **Introduction**  The tutorials are independent of the rest of the document. The primarily objective is to help you learn quickly the first steps. The emphasis here is “learning by doing”. Therefore, the best way to learn is by trying it yourself. Working through the examples will give you a feel for the way that MATLAB operates. In this introduction we will describe how MATLAB handles simple numerical expressions and mathematical formulas. The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.  MATLAB [1] is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems.  MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox.  • How to quit MATLAB 1.3.1 Starting MATLAB After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows.  The major tools within or accessible from the desktop are:  • The Command Window  • The Command History  • The Workspace  • The Current Directory  • The Help Browser  • The Start button 2  The graphical interface to the MATLAB workspace 3 When MATLAB is started for the first time. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs. Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run. You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt: >> for full version EDU> for educational version Note: To simplify the notation, we will use this prompt, >>, as a standard prompt sign, though our MATLAB version is for educational purpose. 1.3.2 Using MATLAB as a calculator As an example of a simple interactive calculation, just type the expression you want to evaluate.  Let’s start at the very beginning. For example, let’s suppose you want to calculate the expression, 1 + 2 × 3. You type it at the prompt command (>>) as follows, >> 1+2\*3 ans = 7 You will have noticed that if you do not specify an output variable, MATLAB uses a default variable ans, short for answer, to store the results of the current calculation. Note that the variable ans is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example, >> x = 1+2\*3 x = 7 will result in x being given the value 1 + 2 × 3 = 7. This variable name can always be used to refer to the results of the previous computations. Therefore, computing 4x will result in >> 4\*x ans = 28.0000 Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic operators. 4 Table 1.1: Basic arithmetic operators Symbol Operation Example + Addition 2 + 3 − Subtraction 2 − 3 ∗ Multiplication 2 ∗ 3 / Division 2/3 1.3.3 Quitting MATLAB To end your MATLAB session, type quit in the Command Window, or select File −→ Exit MATLAB in the desktop main menu. 1.4 Getting started After learning the minimum MATLAB session, we will now learn to use some additional operations. 1.4.1 Creating MATLAB variables MATLAB variables are created with an assignment statement. The syntax of variable assignment is variable name = a value (or an expression) For example, >> x = expression where expression is a combination of numerical values, mathematical operators, variables, and function calls.  On other words, expression can involve:  • manual entry  • built-in functions  • user-defined functions 5 1.4.2  Overwriting variable Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line.  Then the sequence of commands looks like this:  >> t = 5; >> t = t+1 t = 6 1.4.3  Error messages If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, \*, in the following expression >> x = 10; >> 5x ??? 5x | Error: Unexpected MATLAB expression. 1.4.4 Making corrections To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key ↑. When the command is displayed at the command prompt, it can be modified if needed and executed. 1.4.5 Controlling the hierarchy of operations or precedence Let’s consider the previous arithmetic operation, but now we will include parentheses. For example, 1 + 2 × 3 will become (1 + 2) × 3 >> (1+2)\*3 ans = 9 and, from previous example 6 >> 1+2\*3 ans = 7 By adding parentheses, these two expressions give different results: 9 and 7. The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. Exponentiations are done first, followed by multiplications and divisions, and finally by additions and subtractions.  However, the standard order of precedence of arithmetic operations can be changed by inserting parentheses. For example, the result of 1+2×3 is quite different than the similar expression with parentheses (1+2)×3. The results are 7 and 9 respectively. Parentheses can always be used to overrule priority, and their use is recommended in some complex expressions to avoid ambiguity. Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules. Hierarchy of arithmetic operations Precedence Mathematical operations First The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward. Second All exponentials are evaluated, working from left to right Third All multiplications and divisions are evaluated, working from left to right Fourth All additions and subtractions are evaluated, starting from left to right most computer programs. For operators of equal precedence, evaluation is from left to right. Now, consider another example: 1 2 + 32 + 4 5 × 6 7 In MATLAB, it becomes >> 1/(2+3^2)+4/5\*6/7 ans = 0.7766 or, if parentheses are missing, >> 1/2+3^2+4/5\*6/7 ans = 10.1857 7 So here what we get: two different results. Therefore, we want to emphasize the importance of precedence rule in order to avoid ambiguity.  1.4.6 Controlling the appearance of floating point number MATLAB by default displays only 4 decimals in the result of the calculations, for example −163.6667, as shown in above examples. However, MATLAB does numerical calculations in double precision, which is 15 digits. The command format controls how the results of computations are displayed. Here are some examples of the different formats together with the resulting outputs. >> format short >> x=-163.6667 If we want to see all 15 digits, we use the command format long >> format long >> x= -1.636666666666667e+002 To return to the standard format, enter format short, or simply format. There are several other formats. For more details, see the MATLAB documentation, or type help format.  Note - Up to now, we have let MATLAB repeat everything that we enter at the prompt (>>). Sometimes this is not quite useful, in particular when the output is pages en length. To prevent MATLAB from echoing what we type, simply enter a semicolon (;) at the end of the command. For example, >> x=-163.6667; and then ask about the value of x by typing, >> x x = -163.6667 1.4.7 Managing the workspace The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a clear command at the start of each new independent calculation. 8 >> clear The command clear or clear all removes all variables from the workspace. This frees up system memory. In order to display a list of the variables currently in the memory, type >> who while, whos will give more details which include size, space allocation, and class of the variables. |