**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **09/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **Matlab from mathworld** | **USN:** | **4AL15EC 024** |
| **Topic:** | **Programming, project** | **Semester & Section:** | **8th - A** |
| **GitHub Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |

**REPORT –**

MATLAB (*matrix laboratory*) is a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) [numerical computing](https://en.wikipedia.org/wiki/Numerical_analysis) environment and [proprietary programming language](https://en.wikipedia.org/wiki/Proprietary_programming_language) developed by [MathWorks](https://en.wikipedia.org/wiki/MathWorks" \o "MathWorks). MATLAB allows [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) manipulations, plotting of [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) and data, implementation of [algorithms](https://en.wikipedia.org/wiki/Algorithm), creation of [user interfaces](https://en.wikipedia.org/wiki/User_interface), and interfacing with programs written in other languages.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the [MuPAD](https://en.wikipedia.org/wiki/MuPAD" \o "MuPAD) [symbolic engine](https://en.wikipedia.org/wiki/Computer_algebra_system) allowing access to [symbolic computing](https://en.wikipedia.org/wiki/Symbolic_computing) abilities. An additional package, [Simulink](https://en.wikipedia.org/wiki/Simulink), adds graphical multi-domain simulation and [model-based design](https://en.wikipedia.org/wiki/Model-based_design) for [dynamic](https://en.wikipedia.org/wiki/Dynamical_system) and [embedded systems](https://en.wikipedia.org/wiki/Embedded_system).

As of 2020, MATLAB has more than 4 million users worldwide.[[20]](https://en.wikipedia.org/wiki/MATLAB#cite_note-mathworksCompanyOverview-20) MATLAB users come from various backgrounds of [engineering](https://en.wikipedia.org/wiki/Engineering), [science](https://en.wikipedia.org/wiki/Science), and [economics](https://en.wikipedia.org/wiki/Economics).

Use [table](https://in.mathworks.com/help/matlab/ref/table.html) to store tabular data that you can use later in fitting and analysis at the command line. Use [readtable](https://in.mathworks.com/help/matlab/ref/readtable.html) to import data without NONMEM® interpretation of column headers. The readtable function lets you use name-value pair arguments in which you can specify options such as the type of delimiter and whether the first row contains header names. If you want to use the data for fitting using sbiofit or sbiofitmixed, convert it to a groupedData format using [groupedData](https://in.mathworks.com/help/simbio/ref/groupeddata.html).

To prepare the data file for import, remove any comments that are present at the beginning of the file.

For example:

% Text files

data = readtable('tobramycin.txt');

% Text files with . in place of missing values

data = readtable('tobramycin.txt', 'TreatAsEmpty', '.');

## Syntax

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical [shell](https://en.wikipedia.org/wiki/Command_line_interface) or executing text files containing MATLAB code.[[24]](https://en.wikipedia.org/wiki/MATLAB#cite_note-24)

### Variables

Variables are defined using the assignment operator, =. MATLAB is a [weakly typed](https://en.wikipedia.org/wiki/Strong_and_weak_typing) programming language because types are implicitly converted.[[25]](https://en.wikipedia.org/wiki/MATLAB#cite_note-25) It is an inferred typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects,[[26]](https://en.wikipedia.org/wiki/MATLAB" \l "cite_note-26) and that their type can change. Values can come from [constants](https://en.wikipedia.org/wiki/Constant_(computer_science)), from computation involving values of other variables, or from the output of a function. For example:

**>>** x = 17

x =

17

**>>** x = 'hat'

x =

hat

**>>** x = [3\*4, pi/2]

x =

12.0000 1.5708

**>>** y = 3\*sin(x)

y =

-1.6097 3.0000

### Vectors and matrices[[edit](https://en.wikipedia.org/w/index.php?title=MATLAB&action=edit&section=4" \o "Edit section: Vectors and matrices)]

A simple array is defined using the colon syntax: *initial*:*increment*:*terminator*. For instance:

>> array = 1:2:9

array =

1 3 5 7 9

defines a variable named array (or assigns a new value to an existing variable with the name array) which is an array consisting of the values 1, 3, 5, 7, and 9. That is, the array starts at 1 (the *initial* value), increments with each step from the previous value by 2 (the *increment* value), and stops once it reaches (or to avoid exceeding) 9 (the *terminator* value).

>> array = 1:3:9

array =

1 4 7

the *increment* value can actually be left out of this syntax (along with one of the colons), to use a default value of 1.

>> ari = 1:5

ari =

1 2 3 4 5

assigns to the variable named ari an array with the values 1, 2, 3, 4, and 5, since the default value of 1 is used as the increment.

[Indexing](https://en.wikipedia.org/wiki/One-based_indexing) is one-based,[[27]](https://en.wikipedia.org/wiki/MATLAB#cite_note-27) which is the usual convention for [matrices](https://en.wikipedia.org/wiki/Matrix_(mathematics)) in mathematics, unlike zero-based indexing commonly used in other programming languages such as C, C++, and Java.

Matrices can be defined by separating the elements of a row with blank space or comma and using a semicolon to terminate each row. The list of elements should be surrounded by square brackets []. Parentheses () are used to access elements and subarrays (they are also used to denote a function argument list).

>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]

A =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

>> A(2,3)

ans =

11

Sets of indices can be specified by expressions such as 2:4, which evaluates to [2, 3, 4]. For example, a submatrix taken from rows 2 through 4 and columns 3 through 4 can be written as:

>> A(2:4,3:4)

ans =

11 8

7 12

14 1

A square [identity matrix](https://en.wikipedia.org/wiki/Identity_matrix) of size *n* can be generated using the function eye, and matrices of any size with zeros or ones can be generated with the functions zeros and ones, respectively.

>> eye(3,3)

ans =

1 0 0

0 1 0

0 0 1

>> zeros(2,3)

ans =

0 0 0

0 0 0

>> ones(2,3)

ans =

1 1 1

1 1 1

[Transposing](https://en.wikipedia.org/wiki/Transpose) a vector or a matrix is done either by the function transpose or by adding dot-prime after the matrix (without the dot, prime will perform [conjugate transpose](https://en.wikipedia.org/wiki/Conjugate_transpose) for complex arrays):

>> A = [1 ; 2], B = A.', C = transpose(A)

A =

1

2

B =

1 2

C =

1 2

>> D = [0 3 ; 1 5], D.'

D =

0 3

1 5

ans =

0 1

3 5

Most functions accept arrays as input and operate element-wise on each element. For example, mod(2\*J,n) will multiply every element in *J* by 2, and then reduce each element modulo *n*. MATLAB does include standard for and while loops, but (as in other similar applications such as [R](https://en.wikipedia.org/wiki/R_(programming_language))), using the [vectorized](https://en.wikipedia.org/wiki/Array_programming" \o "Array programming) notation is encouraged and is often faster to execute. The following code, excerpted from the function *magic.m*, creates a [magic square](https://en.wikipedia.org/wiki/Magic_square) *M* for odd values of *n* (MATLAB function meshgrid is used here to generate square matrices *I* and *J* containing *1:n*).

[J,I] = meshgrid(1:n);

A = mod(I + J - (n + 3) / 2, n);

B = mod(I + 2 \* J - 2, n);

M = n \* A + B + 1;

### Structures

MATLAB supports structure data types.[[28]](https://en.wikipedia.org/wiki/MATLAB#cite_note-28) Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names[[29]](https://en.wikipedia.org/wiki/MATLAB" \l "cite_note-29) (field look-ups by name, field manipulations, etc.).

### Functions

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Variables and functions are case sensitive.[[30]](https://en.wikipedia.org/wiki/MATLAB#cite_note-30)

### Function handles

MATLAB supports elements of [lambda calculus](https://en.wikipedia.org/wiki/Lambda_calculus) by introducing function handles,[[31]](https://en.wikipedia.org/wiki/MATLAB" \l "cite_note-31) or function references, which are implemented either in .m files or anonymous[[32]](https://en.wikipedia.org/wiki/MATLAB#cite_note-32)/nested functions.[[33]](https://en.wikipedia.org/wiki/MATLAB#cite_note-33)

### Classes and object-oriented programming

MATLAB supports [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics.[[34]](https://en.wikipedia.org/wiki/MATLAB#cite_note-34) However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending on whether the class has *handle* as a super-class (for reference classes) or not (for value classes).[[35]](https://en.wikipedia.org/wiki/MATLAB#cite_note-35)

Method call behavior is different between value and reference classes. For example, a call to a method

object.method();

can alter any member of *object* only if *object* is an instance of a reference class, otherwise value class methods must return a new instance if it needs to modify the object.

An example of a simple class is provided below.

**classdef** Hello

methods

function greet(obj)

disp('Hello!')

end

end

**end**

When put into a file named hello.m, this can be executed with the following commands:

**>>** x = Hello();

**>>** x.greet();

Hello!

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **9/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **IOT** | **USN:** | **4AL15EC024** |
| **Topic:** | **Chapter 6** | **Semester & Section:** | **8TH SEM &A Section** |
| **Github Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS**   ****IoT:Everything needs to be secure**** It is that phase in the IT scene, where IoT has become more than a buzzword. It is now a core technology that is influencing everything it’s touching. What started off as a simple infrastructure connecting a few devices with sensors is now a massive trillion dollar industry. Not just in the US but in India as well, the industry is simply exploding with opportunities and revenue.  IoT or the Internet of Things is opening up newer avenues in the IT industry scene in India in tandem with Big Data and Data Science. However, innovations are still on the cards and are happening at places around the country, where countless experts and researchers are working on real-life concerns and coming up with technical solutions. Today, if IoT stands relevant to the Indian lifestyle, it has four big opportunities in front of it for innovation. Have a look. Agriculture As you know, the agriculture sector in India is in a bad shape. Being predominantly a country that depends on monsoon for crop irrigation, we desperately need a solution that could better crop yields, bring down water requirements and let farmers take better-informed decisions on what to harvest. With data and sensors, this is completely possible and in states like Andhra Pradesh, smart farming is a reality, where farmers are trained on using technology for farming. With smartphones, apps, sensors and controllers, farmers can control field conditions from remote and implement precision irrigation for optimized harvest. Energy Consumption One of the other biggest concerns we have in front of us is energy consumption. With growing population, we have lost track of energy consumption today and thanks to our lethargic attitude, we hardly care about the environment. In such cases, it is technology that has to offer an ideal solution and thankfully, it does. With IoT, it is easier to limit the consumption of electricity and fuel. Smart homes and smart illumination are all designed to automatically turn off when there is no one in a room and give real-time updates on energy consumption. This eliminates us from taking energy requirements for granted and paves the way to make informed decisions on electricity and fuel usage. Healthcare Of all the basic needs, healthcare is very crucial and it is saddening to note that there are tons of counterfeit medicines in the market out there. What’s the worse part? People are unaware of such medicines and intake them without their knowledge. This is a serious concern that IoT has the potential to solve. As a relieving factor, there are some companies working on offering genuine medicines by using IoT infrastructures and QR codes to assess the legitimacy of medicine purchases. If these reach more people, fake medicines can be permanently be done away with. Transportation Connected cars are increasingly making their ways to the market. High-end models now come connected to the internet and offer you a smartphone app for various usage. However, more innovation on this can be more beneficial to owners in terms of fuel usage, emissions per trip, notifications on a car or spare part malfunctioning, driver behaviour, speed limiter, geographic locations, door and trunk lock/unlock, social media sharing and more. In fact, features to add carpooling facilities can also have a huge impact on the environment and we are sure this would soon be a reality. IoT security challenges A number of challenges prevent the securing of IoT devices and ensuring end-to-end security in an IoT environment. Because the idea of networking appliances and other objects is relatively new, security has not always been considered top priority during a product's design phase. Additionally, because IoT is a nascent market, many product designers and manufacturers are more interested in getting their products to market quickly, rather than taking the necessary steps to build security in from the start.  A major issue cited with IoT security is the use of hardcoded or [default passwords](https://whatis.techtarget.com/definition/default-password), which can lead to security breaches. Even if passwords are changed, they are often not strong enough to prevent infiltration.  Another common issue facing IoT devices is that they are often resource-constrained and do not contain the compute resources necessary to implement strong security. As such, many devices do not or cannot offer advanced security features. For example, sensors that monitor humidity or temperature cannot handle advanced encryption or other security measures. Plus, as many IoT devices are "set it and forget it" -- placed in the field or on a machine and left until end of life -- they hardly ever receive security updates or patches. From a manufacturer's viewpoint, building security in from the start can be costly, slow down development and cause the device not to function as it should.  Connecting legacy assets not inherently designed for IoT connectivity is another security challenge. Replacing legacy infrastructure with connected technology is cost-prohibitive, so many assets will be retrofitted with smart sensors. However, as legacy assets that likely have not been updated or ever had security against modern threats, the attack surface is expanded.  In terms of updates, many systems only include support for a set timeframe. For legacy and new assets, security can lapse if extra support is not added. And as many IoT devices stay in the network for many years, adding security can be challenging.  IoT security is also plagued by a lack of industry-accepted standards. While many IoT security frameworks exist, there is no single agreed-upon framework. Large companies and industry organizations may have their own specific standards, while certain segments, such as industrial IoT, have proprietary, incompatible standards from industry leaders. The variety of these standards makes it difficult to not only secure systems, but also ensure interoperability between them.  The convergence of IT and operational technology (OT) networks has created a number of challenges for security teams, especially those tasked with protecting systems and ensuring end-to-end security in areas outside their realm of expertise. A learning curve is involved, and IT teams with the proper skill sets should be put in charge of IoT security.  Organizations must learn to view security as a shared issue, from manufacturer to service provider to end user. Manufacturers and service providers should prioritize the security and privacy of their products, and also provide encryption and authorization by default, for example. But the onus does not end there; end users must be sure to take their own precautions, including changing passwords, installing patches when available and using security software. How to protect IoT systems and devices IoT security methods vary depending on your specific IoT application and your place in the IoT ecosystem. For example, IoT manufacturers -- from product makers to semiconductor companies -- should concentrate on building security in from the start, making hardware tamper-proof, building secure hardware, ensuring secure upgrades, providing firmware updates/patches and performing dynamic testing. A solution developer's focus should be on secure software development and secure integration. For those deploying IoT systems, hardware security and authentication are critical measures. Likewise, for operators, keeping systems up to date, mitigating malware, auditing, protecting infrastructure and safeguarding credentials are key. |