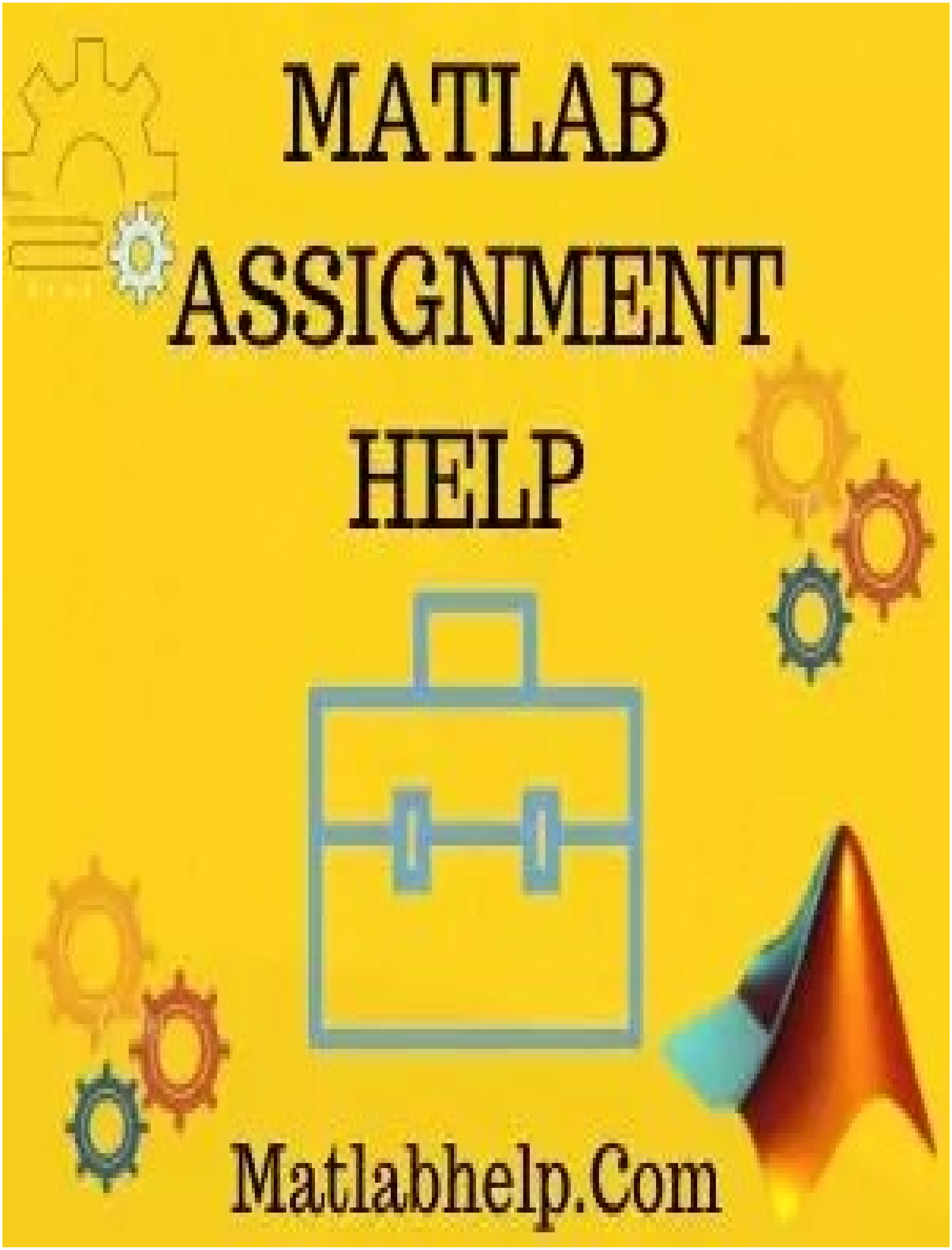
MATLAB
ASSIGNMENT
HELP
Matlabhelp.Com

Lecture2:

A first steps:

To get matlab to work out 1 + 1, type the following at the prompt:

```
1+1
```

matlab responds with

```
ans = 2
```

The answer to the typed command is given the name ans. In fact ans is now a variable that you can use again. For example you can type

```
ans*ans
```

to check that 2 × 2=4:

```
ans*ans
ans = 4
```

Matlab has updated the value of ans to be 4.

The spacing of operators in formulas does not matter. The following formulas both give the same answer:

```
1+3 * 2-1 / 2*4 1+3*2-1/2*4
```

The order of operations is made clearer to readers of your matlab code if you type carefully:

```
1 + 3*2 - (1/2)*4
```

Entering Matrices

The best way for you to get started with MATLAB is to learn how to handle matrices, to type a matrix into matlab you must:

- begin with a square bracket [
- separate elements in a row with commas or spaces
- use a semicolon ; to separate rows
- end the matrix with another square bracket].

For example type:

```
a=[1 2 3;4 5 6;7 8 9]
```

Matlab responds with

```
a=
     1   2  3
     4   5  6
     7   8  9
```

Variables and assignment:

Variables are named locations in memory where numbers, strings and other elements of data may be stored while the program is working. Variable names are combinations of letters and digits, but must start with a latter. MATLAB does not require you to declare the names of variables in advance of their use. This is actually a common cause of error, since it allows you to refer accidentally to variables that don’t exist. To assign a variable a value, use the assignment statement. This takes the form

```
variable=expression;
```

for example

```
a=6;
```

or

```
name=’Mark’;
```

To display the contents of a variable, use

```
disp(variable);
```

Useful Matrix Generators :

Matlab provides four easy ways to generate certain simple matrices. These are

Zeros	a matrix filled with zeros
ones	a matrix filled with ones
randi	a matrix with integer values distributed random elements

eye identity matrix

To tell matlab how big these matrices should be you give the functions the number of rows and columns. For example:

```
 >>u=randi(10,[2 2])
u=
     6     5
     1     4

 >>eye(3(
ans=
1       0     0
0       1     0
0       0     1
```

Subscripting :

Individual elements in a matrix are denoted by a row index and a column index. To pick out the third element of the vector u type:

```
>> u(3)
 ans =
0.1270
```

You can use the vector [1 2 3] as an index to u. To pick the first three elements of u type

```
>> u([1 2 3])
```

```
ans =
 0.8147   0.9058   0.1270
```

Remembering what the colon operator does, you can abbreviate this to

```
>> u(1:3)
 ans =
0.8147
0.9058
0.1270
```

You can also use a variable as a subscript:

```
>> i = 1:3;
 >> u(i)
ans =
0.8147
0.9058
0.1270
```

Two dimensional matrices are indexed the same way, only you have to provide two indices:

```
>>a=[1 2 3;4 5 6;7 8 9]
```

```
a=
1     2    3
4     5    6
7     8    9

>> a(3,2)
 ans =
8
>> a(2:3,3)
ans  =
6       9

>> a(2,:)
ans =
4     5      6

 >> a(:,3)
ans =
 3
 6
 9
```

The last two examples use the colon symbol as an index, which matlab interprets as the entire row or column.

If a matrix is addressed using a single index, matlab counts the index down successive columns:

```
> [a a(a)]
ans =
1   2  3  1  4  7
4   5  6  2  5  8
7   8  9  3  6  9
```

The colon symbol can be used as a single index to a matrix. Continuing the previous example, if you type

```
a(:)
```

matlab interprets this as the columns of the a-matrix successively strung out in a single long column:

```
>> a(:)
ans =
1
4
7
2
5
8
3
6
9
```