The function `SIMPLE2D_M` is a MATLAB version of the `SIMPLE2D` function required to solve the second question in the assignment 2 of the ECSE 543.

Syntax:

```
Potential = SIMPLE2D_M('file.dat')
Potential = SIMPLE2D_M('file1.dat','file2.dat','file3.dat')
```

Description:

`Potential = SIMPLE2D_M('file.dat')` accepts a single input `'file.dat'` containing the mesh and boundary condition info and returns the value of the potential at each node (including those resided on the boundaries). The format of the input file is exactly the same as the original SIMPLE2D function available in the Finite-Element lecture note and shown in Fig. 1 below. However, it returns only a single array in the format shown in Fig. 2. It is important to note that the input file contains three sets of data with different number of columns. Do not add extra entries to make the number of columns the same. The data are distinguished based on the number of columns.

If it is difficult for you to prepare such an input file, you can alternatively use the following syntax:

```
Potential = SIMPLE2D_M('file1.dat','file2.dat','file3.dat')
```

in which `'file1.dat'`, 'file2.dat', and `'file3.dat'` are three separate input files. Each of them contains only one set of the data as shown in Fig. 3. It returns exactly the same answer as the previous syntax.

Send your questions and feedbacks to ali.akbarzadehsharbaf@mail.mcgill.ca

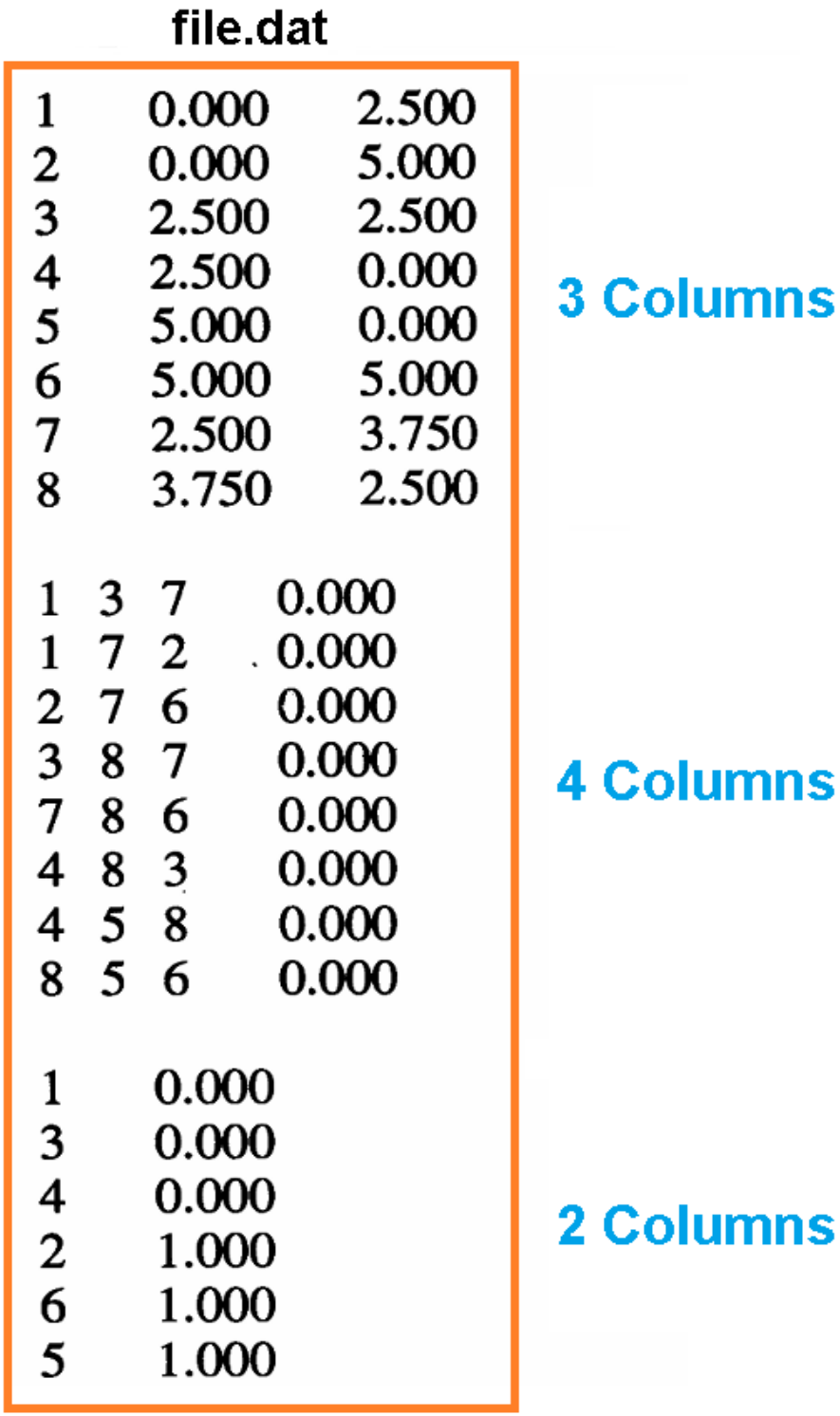

Fig. 1 The input file must contain three set of data with different number of columns as written next to each one.

SIMPLE2D_M Output

Node #	X	Y	Potential
1	0.00000	2.50000	0.00000
2	0.00000	5.00000	1.00000
3	2.50000	2.50000	0.00000
4	2.50000	0.00000	0.00000
5	5.00000	0.00000	1.00000
6	5.00000	5.00000	1.00000
7	2.50000	3.75000	0.58000
8	3.75000	2.50000	0.58000

Fig. 2 Format of the output variable.

file1.dat

1	0.000	2.500
2	0.000	5.000
3	2.500	2.500
4	2.500	0.000
5	5.000	0.000
6	5.000	5.000
7	2.500	3.750
8	3.750	2.500

file2.dat

1	3	7	0.000
1	7	2	0.000
2	7	6	0.000
3	8	7	0.000
7	8	6	0.000
4	8	3	0.000
4	5	8	0.000
8	5	6	0.000

file3.dat

1	0.000
3	0.000
4	0.000
2	1.000
6	1.000
5	1.000

Fig. 3 Input files format for the alternative syntax. Each set of data must be stored in a separate file.