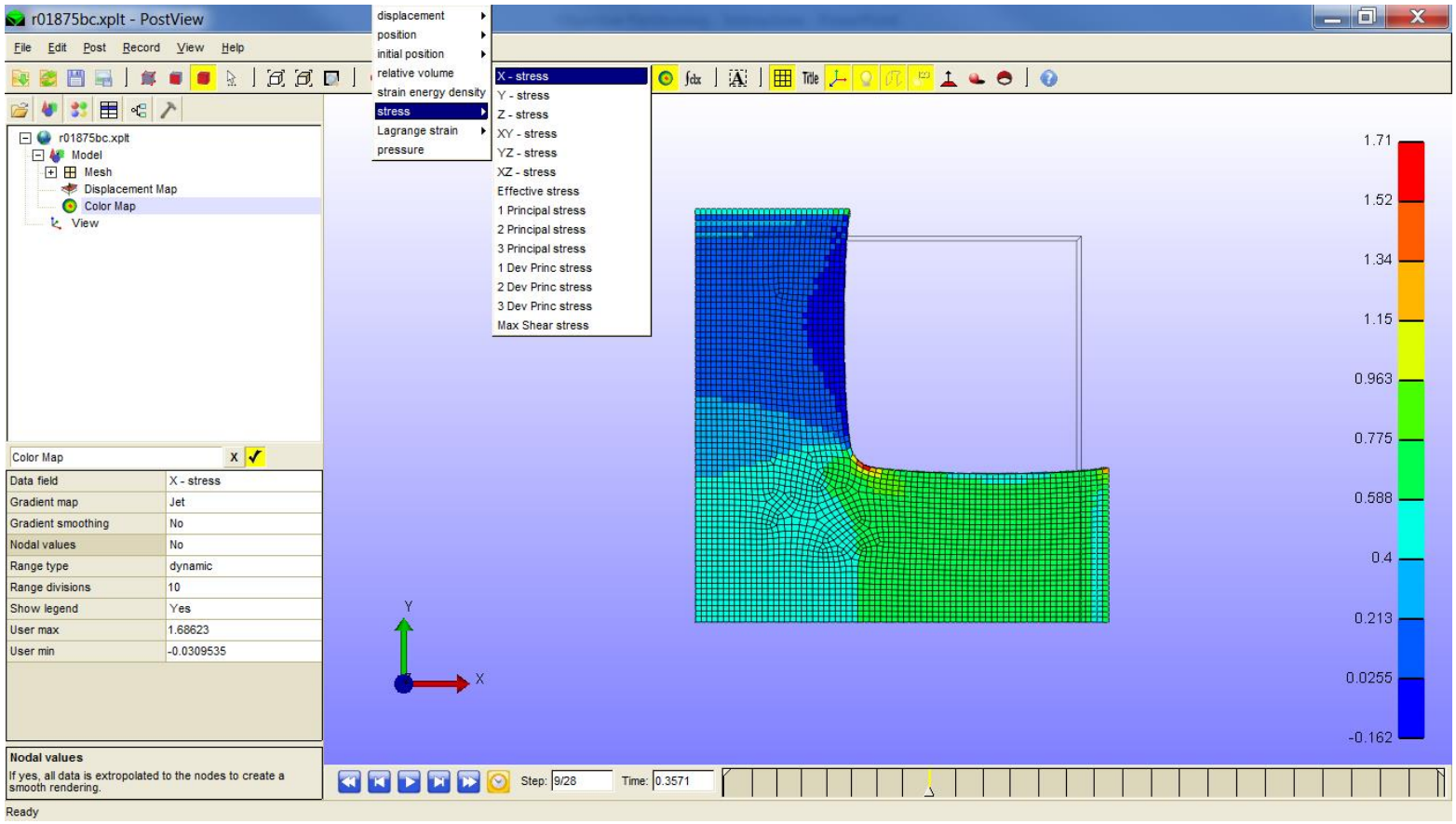
How to implement Objective Partitioning?

1. Export physical data from PostView (.xplt) file

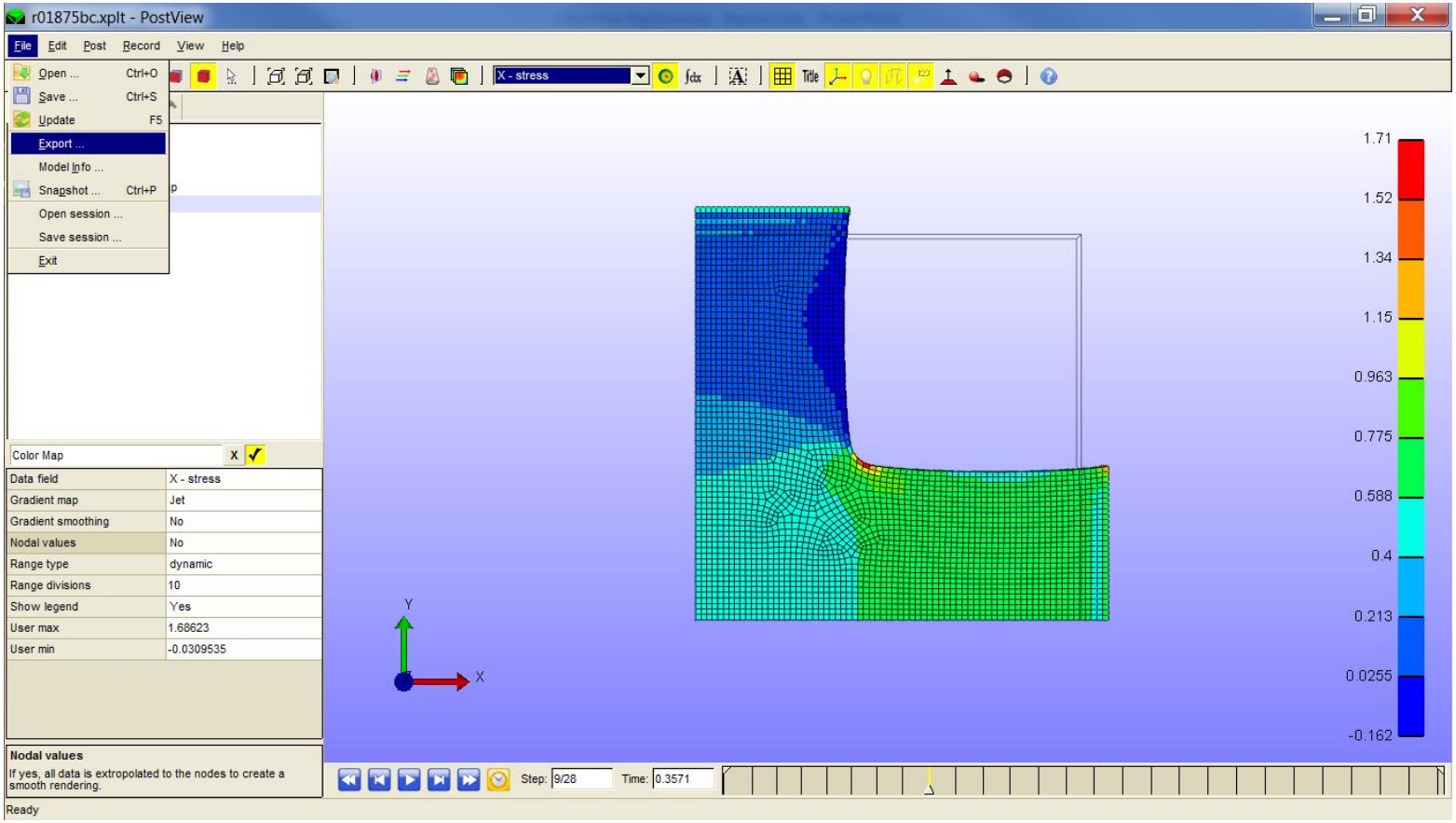
1.1. Select physical property of an interest (e.g. x – stress)



How to implement Objective Partitioning?

1. Export physical data from PostView (.xplt) file

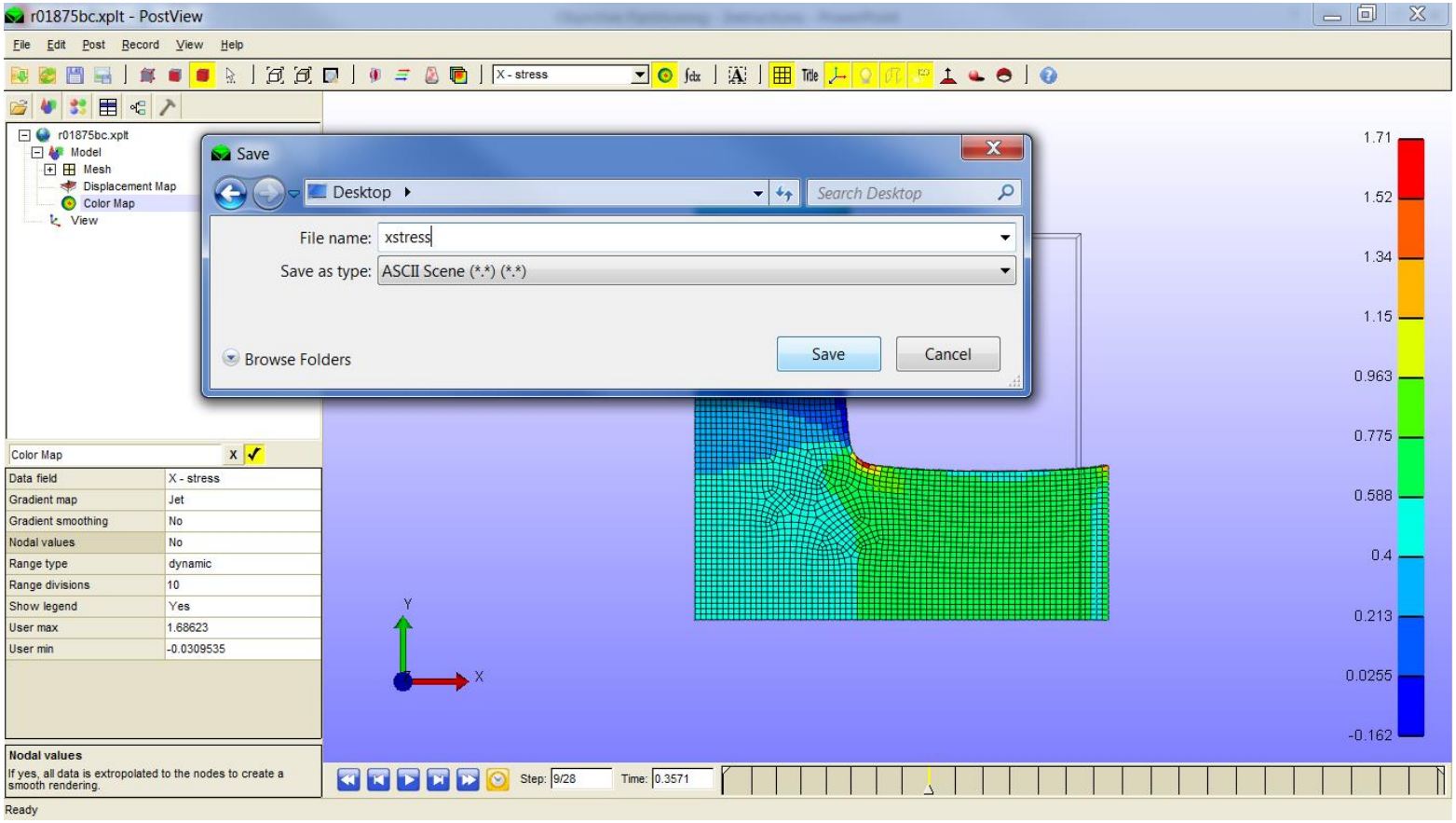
1.2. Export data file



How to implement Objective Partitioning?

1. Export physical data from PostView (.xplt) file

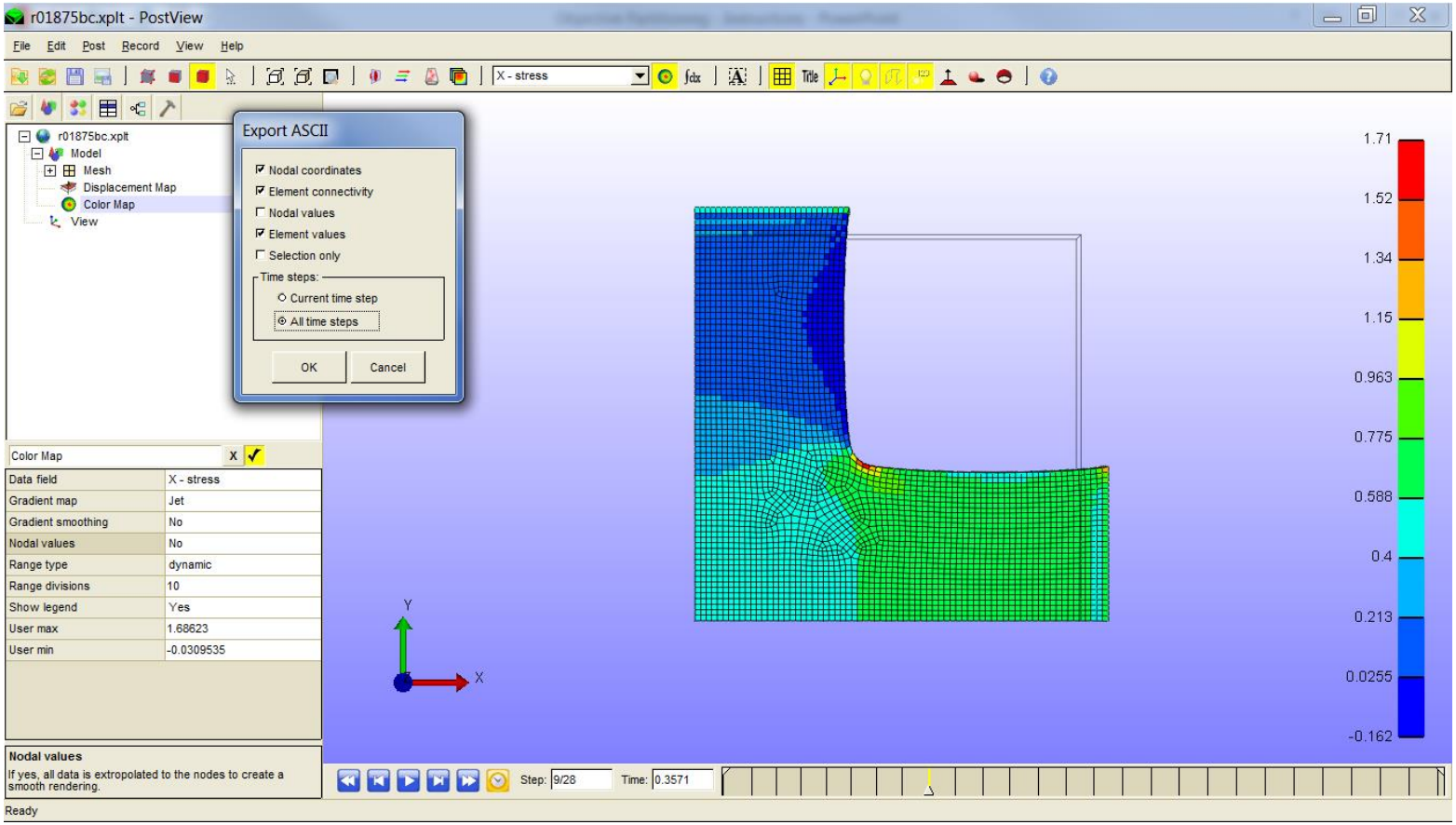
1.3. Save data file as an ASCII extension



How to implement Objective Partitioning?

1. Export physical data from PostView (.xplt) file

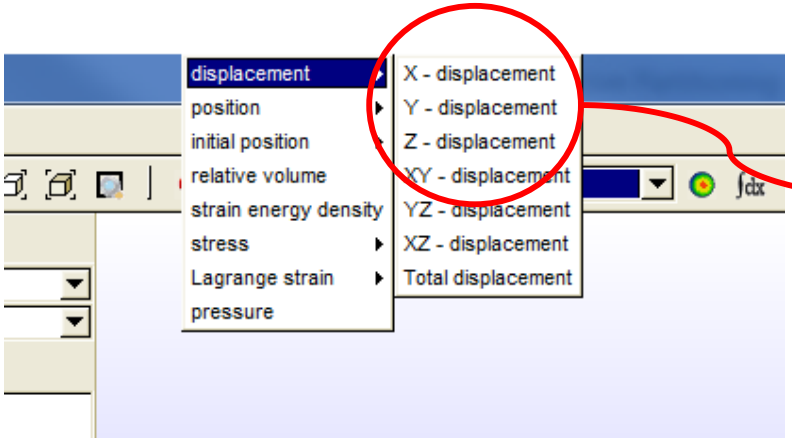
1.4. Save nodal coordinates, element connectivity, and values



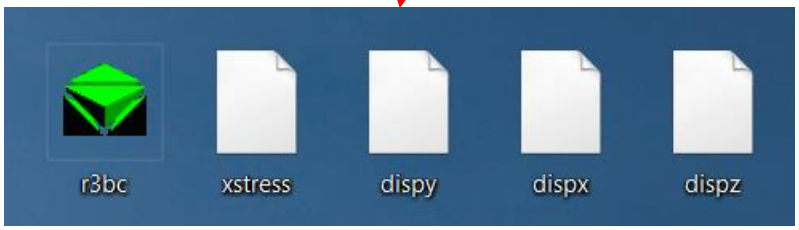
How to implement Objective Partitioning?

1. Export physical data from PostView (.xplt) file

1.5. Save/Export the displacement data



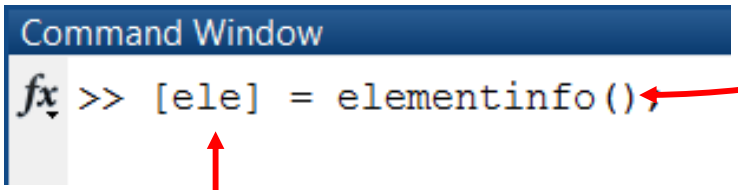
In addition to physical properties of interest, export:
X – displacement
Y – displacement
Z – displacement



How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

2.1. Executing code



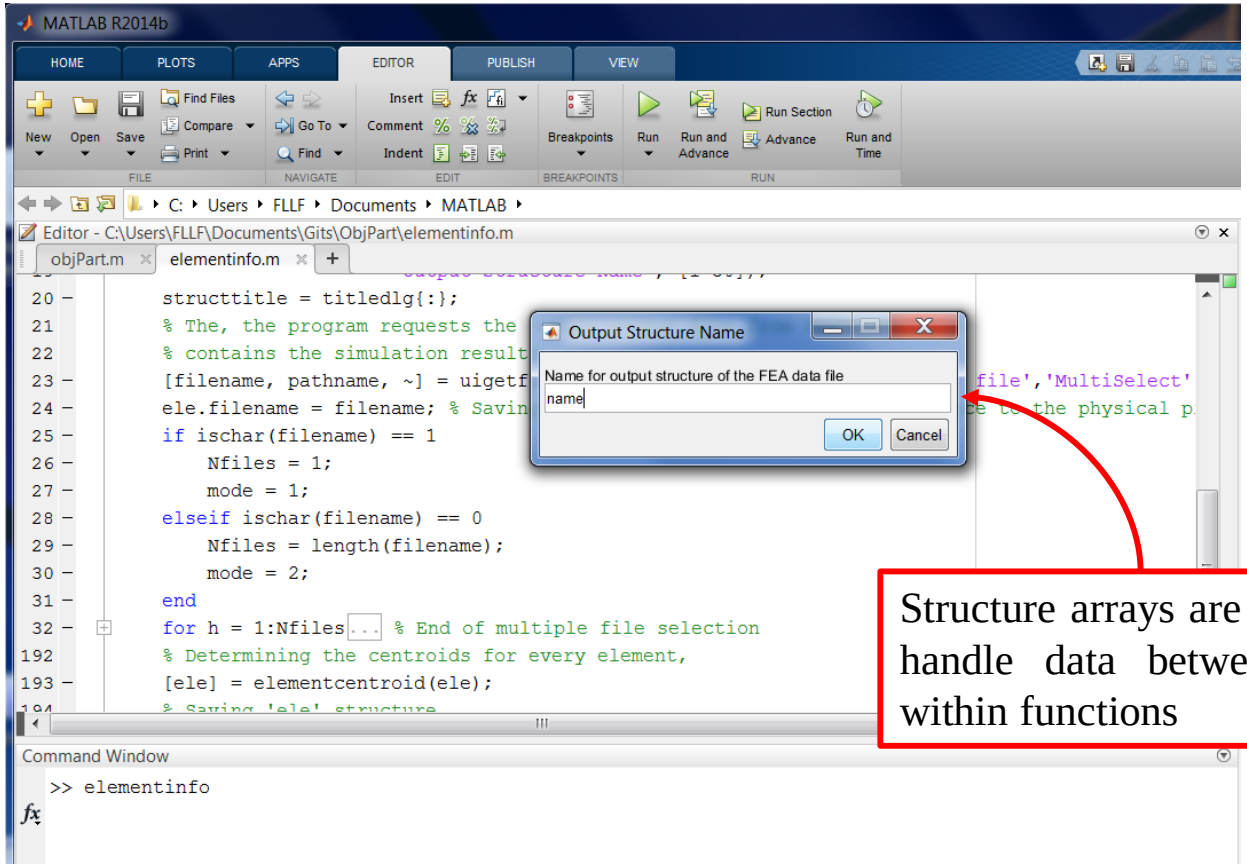
No inputs are needed as the functions requests data through browser windows

The user may choose a variable name to identify the output structure within the MATLAB® workspace

How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

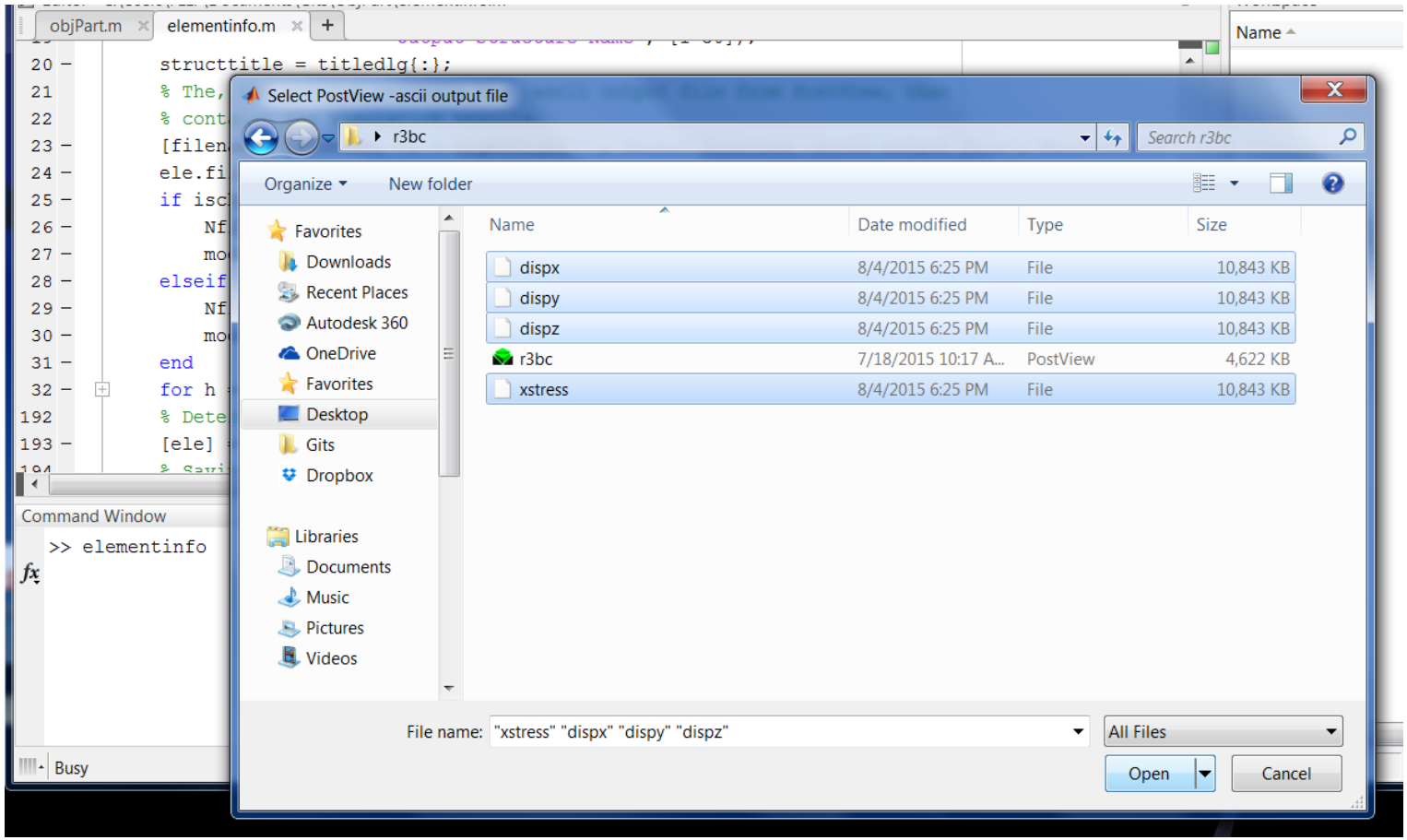
2.2. Input name for output structure



How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

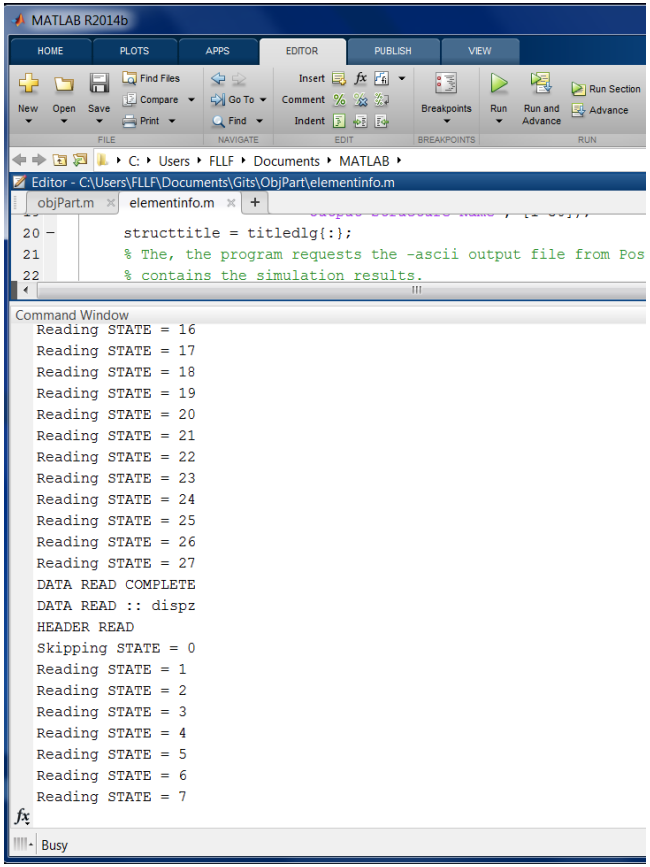
2.3. Select all the data files needed and those of interest



How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

2.4. Wait for data to be read



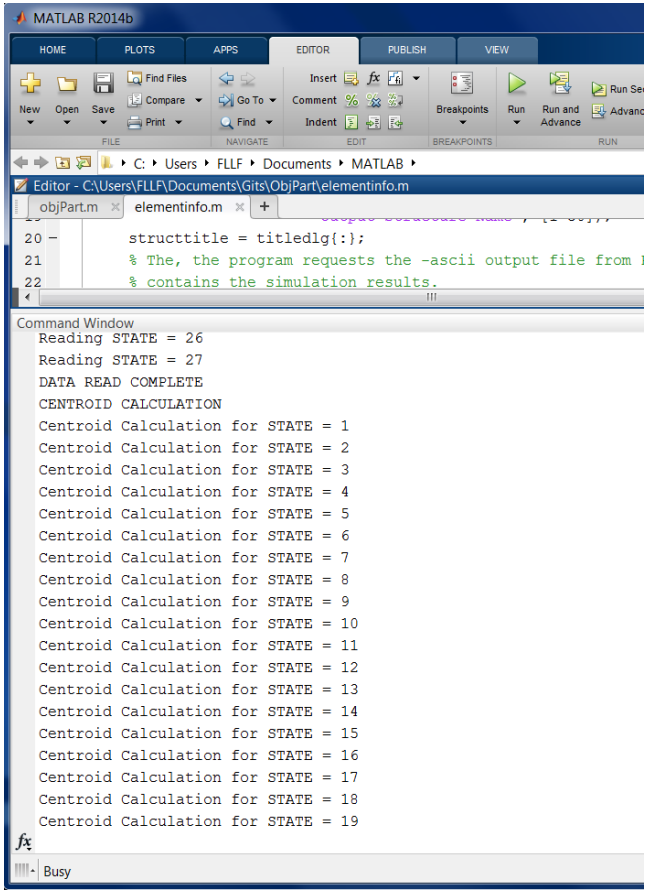
Data from each selected file is read in order

Messages in the command window indicate the current file and the simulation time/state

How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

2.5. Wait for centroid calculation

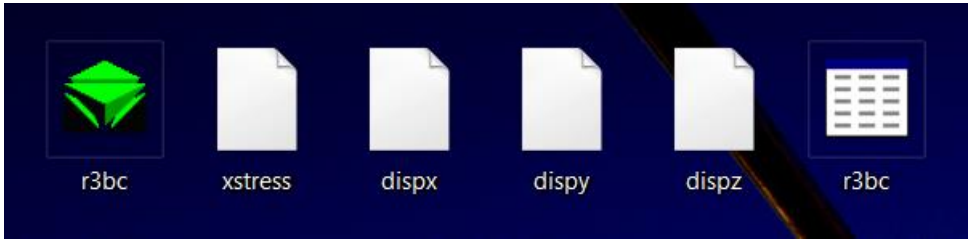


Centroid calculation is an additional task that prolongs the read process

How to implement Objective Partitioning?

2. Compile simulation data using “elementinfo.m”

2.6. Identify output



PostView file

Data files
(ASCII exports)

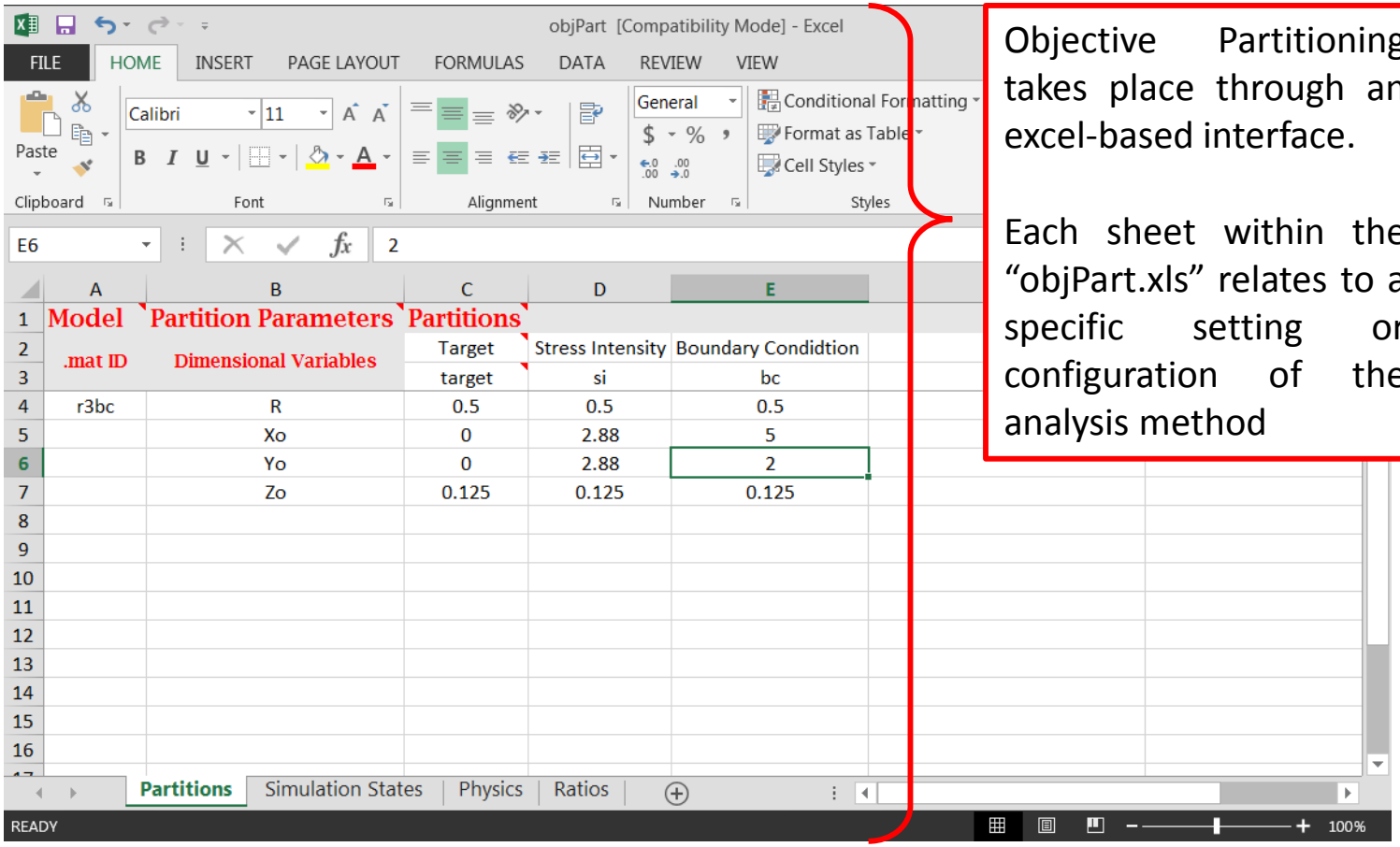
Structure Array from
“elementinfo.m”

File saved in the
directory of inputs

How to implement Objective Partitioning?

3. Performing Objective Partitioning through "objPart.m"

3.1. Preparing input



	A	B	C	D	E
1	Model	Partition Parameters	Partitions		
2	.mat ID	Dimensional Variables	Target	Stress Intensity	Boundary Condidtion
3			target	si	bc
4	r3bc	R	0.5	0.5	0.5
5		Xo	0	2.88	5
6		Yo	0	2.88	2
7		Zo	0.125	0.125	0.125

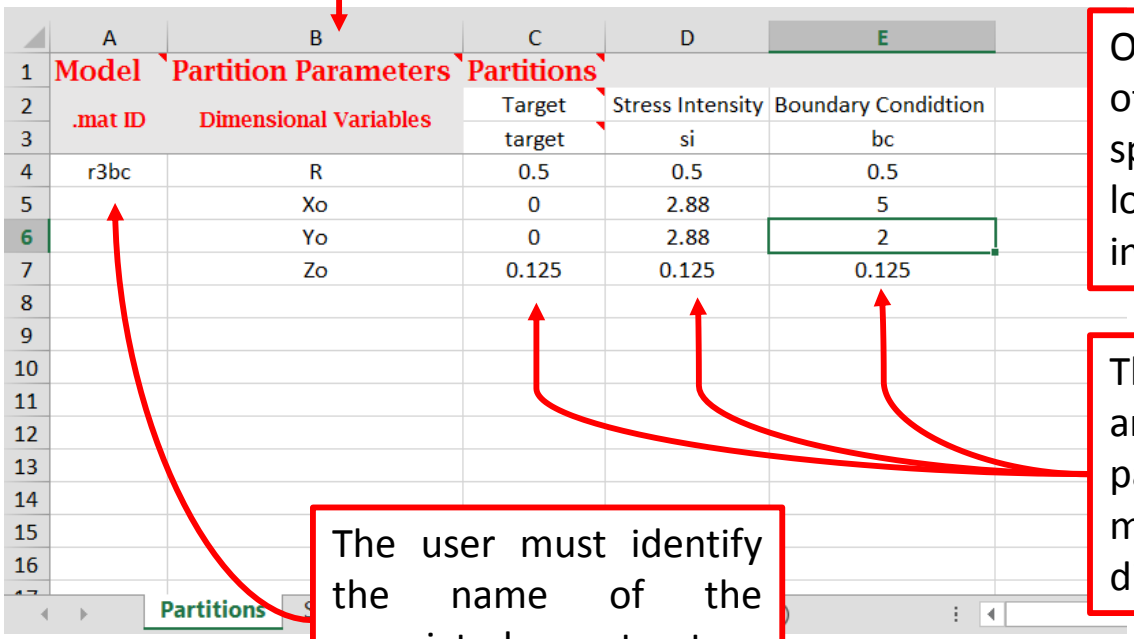
Objective Partitioning takes place through an excel-based interface.

Each sheet within the "objPart.xls" relates to a specific setting or configuration of the analysis method

How to implement Objective Partitioning?

3. Performing Objective Partitioning through "objPart.m"

3.1.1. Defining Partitions



	A	B	C	D	E
1	Model	Partition Parameters	Partitions		
2	.mat ID	Dimensional Variables	Target	Stress Intensity	Boundary Condidtion
3			target	si	bc
4	r3bc	R	0.5	0.5	0.5
5		Xo	0	2.88	5
6		Yo	0	2.88	2
7		Zo	0.125	0.125	0.125

Partitions

Objective Partitions, as of now, are defined by spheres of radius R and located at an origin of interest (Xo, Yo, Zo)

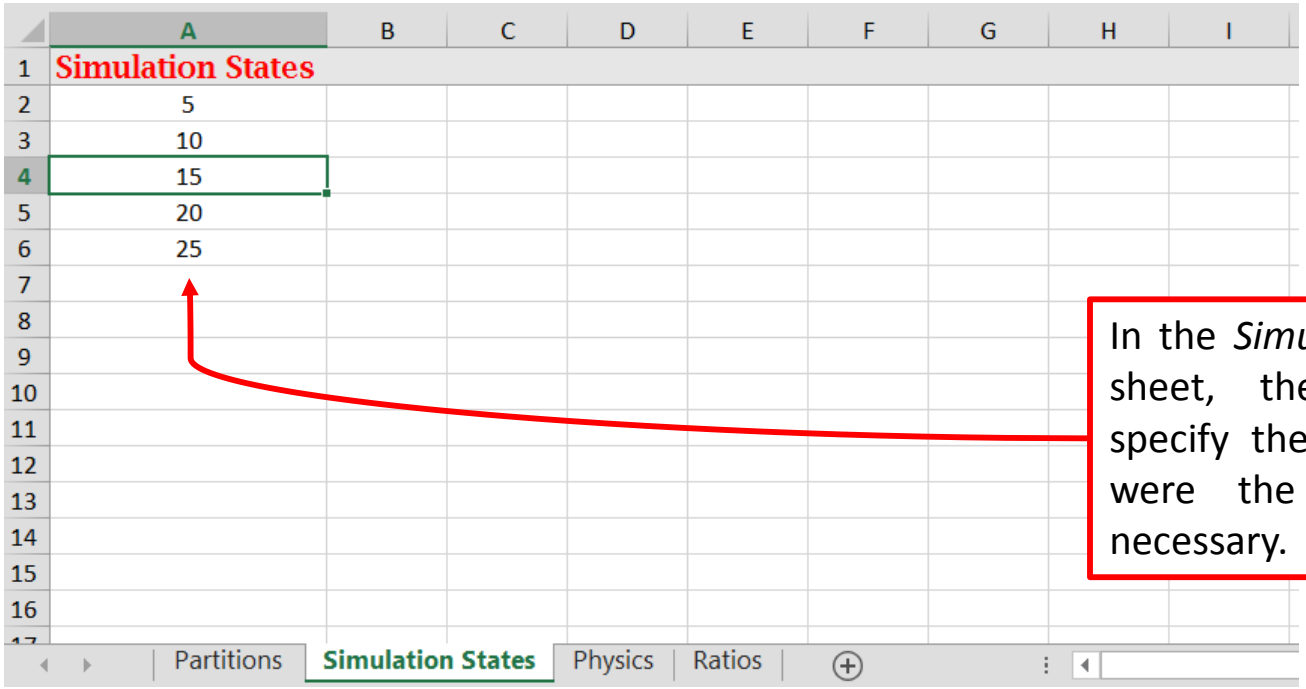
The user may specify any number of partitions. All of which must have a name and dimensions

The user must identify the name of the associated structure array

How to implement Objective Partitioning?

3. Performing Objective Partitioning through "objPart.m"

3.1.2. Specifying Simulation States

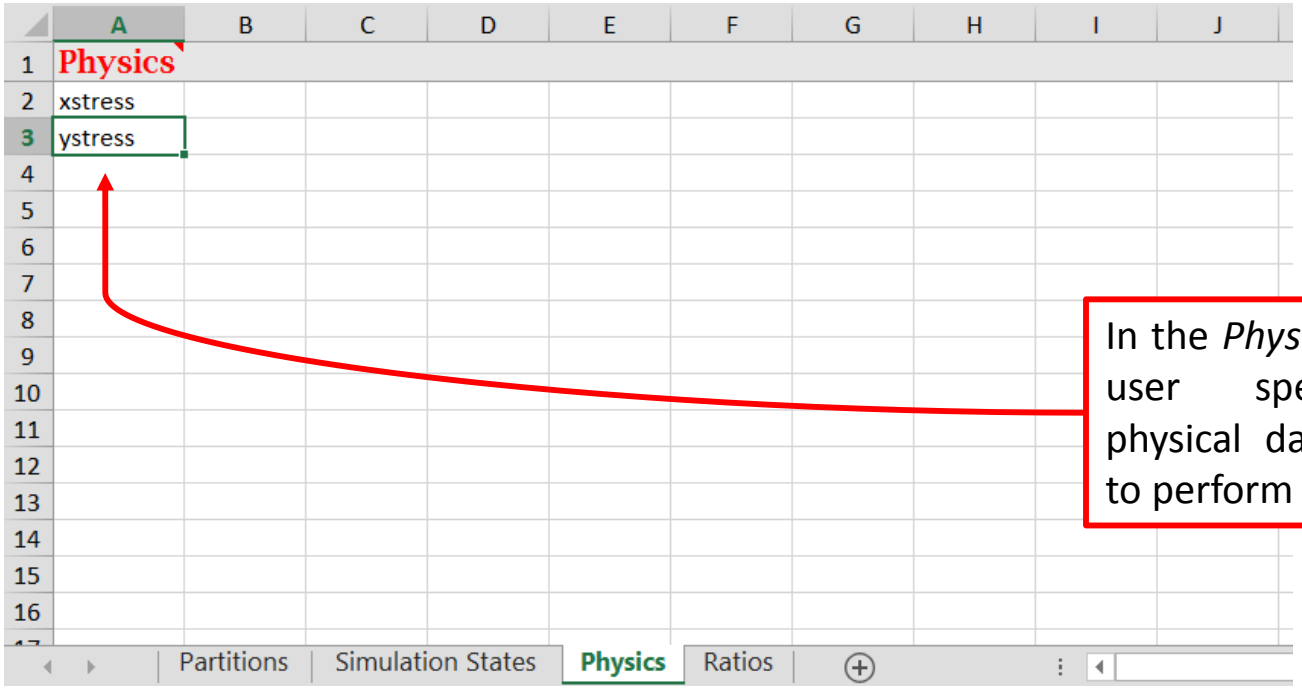


In the *Simulation States* sheet, the user can specify the time points were the analysis is necessary.

How to implement Objective Partitioning?

3. Performing Objective Partitioning through "objPart.m"

3.1.3. Selecting Physics

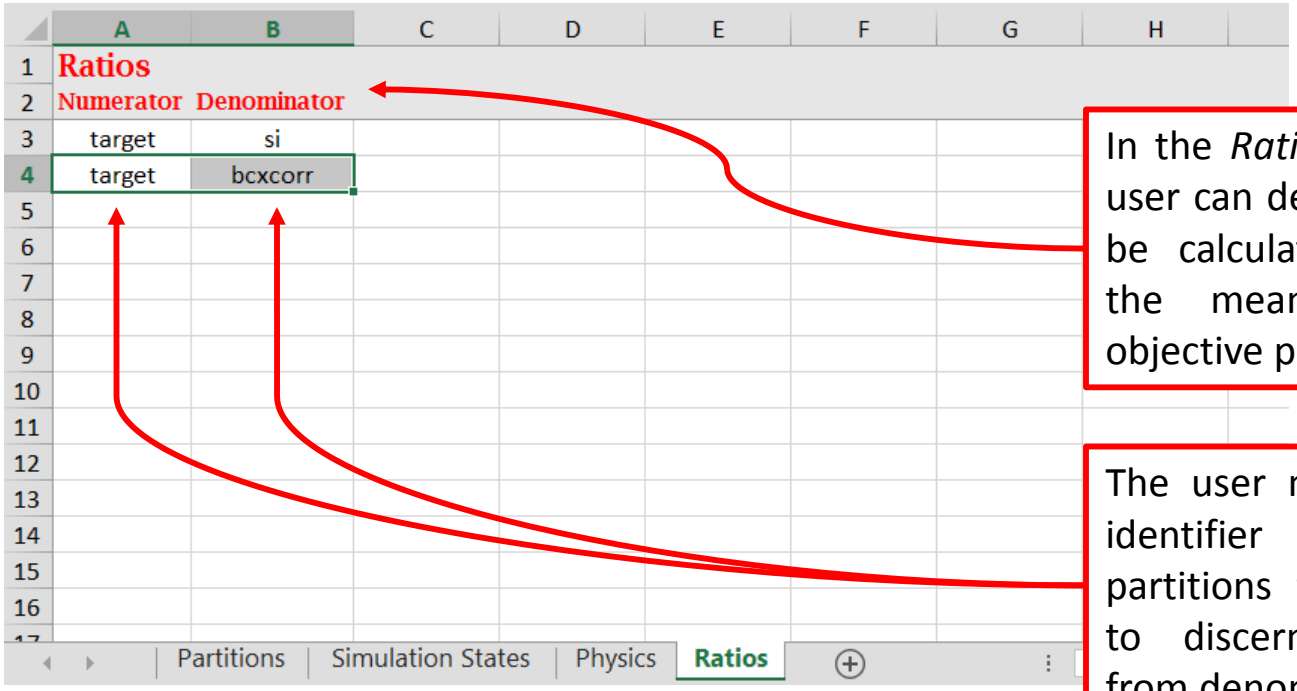


In the *Physics* sheet, the user specifies the physical data on which to perform the analysis

How to implement Objective Partitioning?

3. Performing Objective Partitioning through "objPart.m"

3.1.4. Ratios



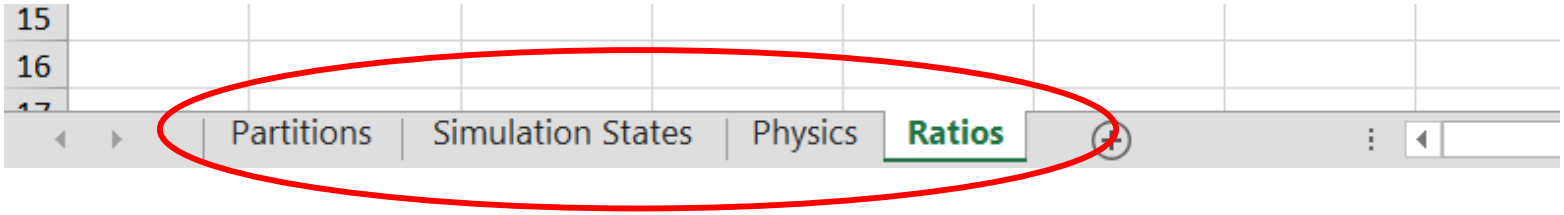
In the *Ratios* sheet, the user can define ratios to be calculated between the means of each objective partition.

The user must use the identifier of the partitions for the code to discern numerator from denominator

How to implement Objective Partitioning?

3. Performing Objective Partitioning through “objPart.m”

3.1.5. Warning!!

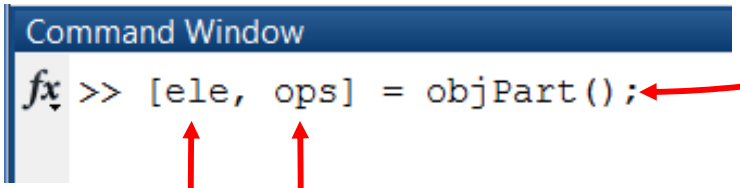


All sheets covered between sections **3.1.1.** and **3.1.4.** must have **at least one input**.
Otherwise, the whole program wont work!!

How to implement Objective Partitioning?

3. Performing Objective Partitioning through “objPart.m”

3.2. Executing code



No inputs are needed as the functions requests data through browser windows

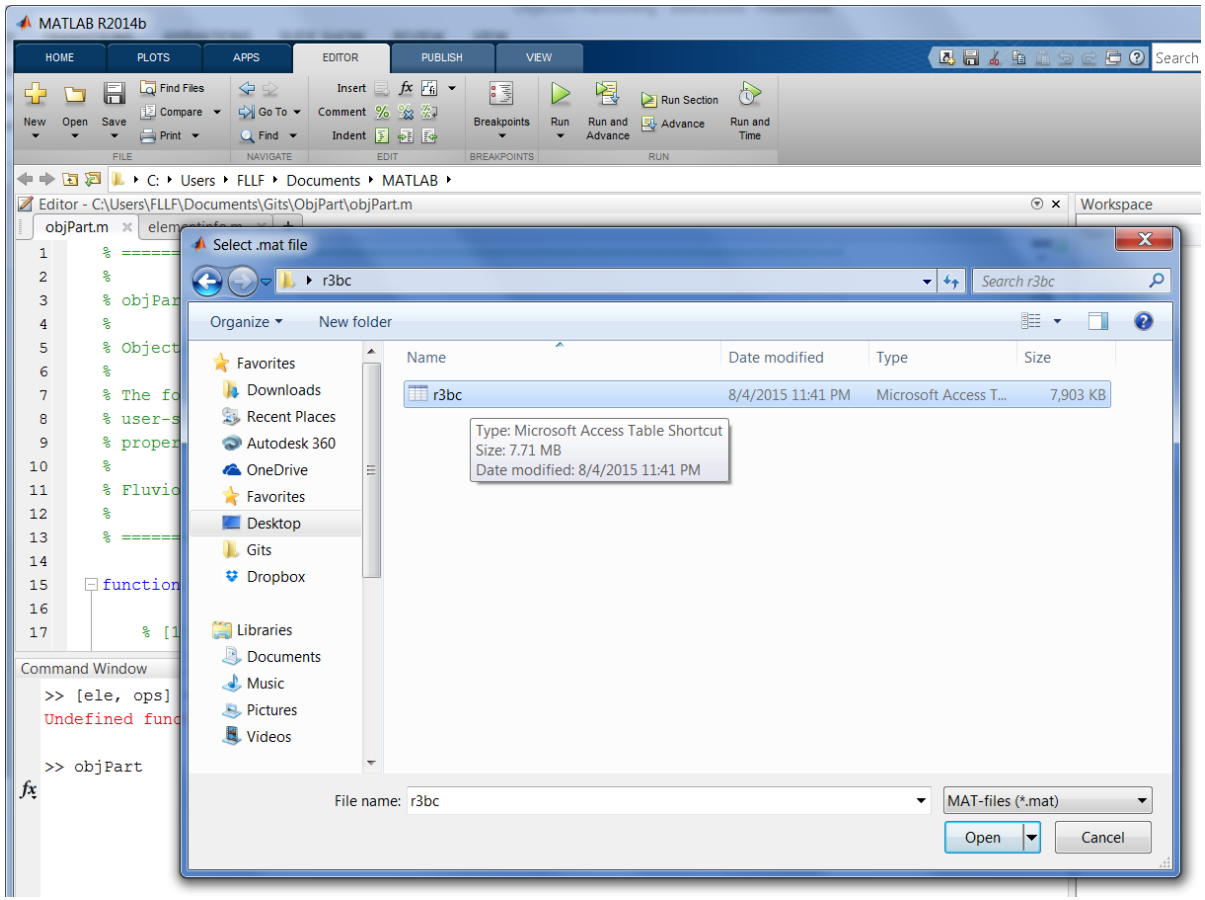
The program modifies and outputs the structure array created by “elementinfo.m”

The variable “ops” contains the information that defines the objective partitions

How to implement Objective Partitioning?

3. Performing Objective Partitioning through “ObjPart.m”

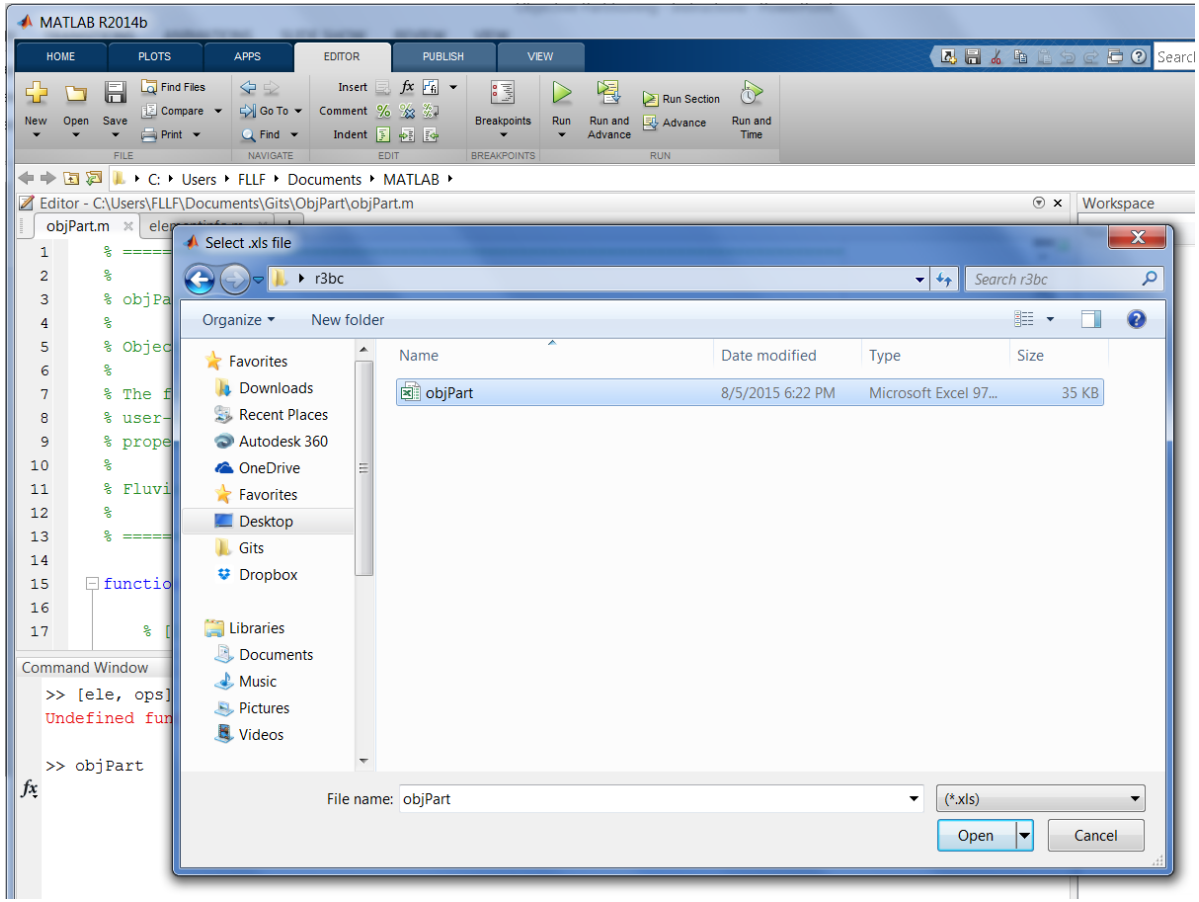
3.3. Browse for data structure array “ele”



How to implement Objective Partitioning?

3. Performing Objective Partitioning through “ObjPart.m”

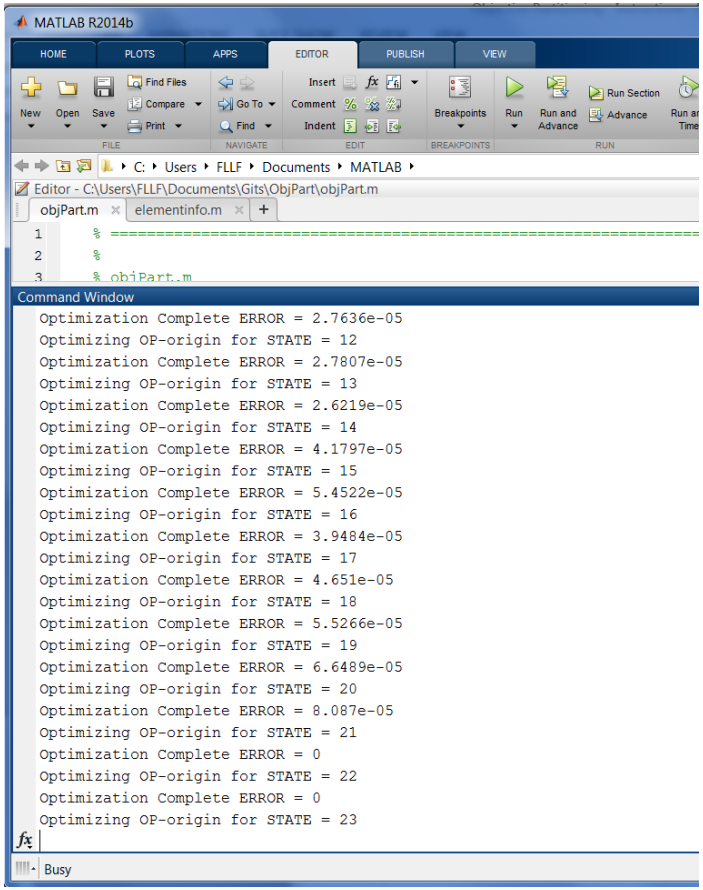
3.4. Browse for “objPart.xls”



How to implement Objective Partitioning?

3. Performing Objective Partitioning through “ObjPart.m”

3.5. Wait for optimizations to complete

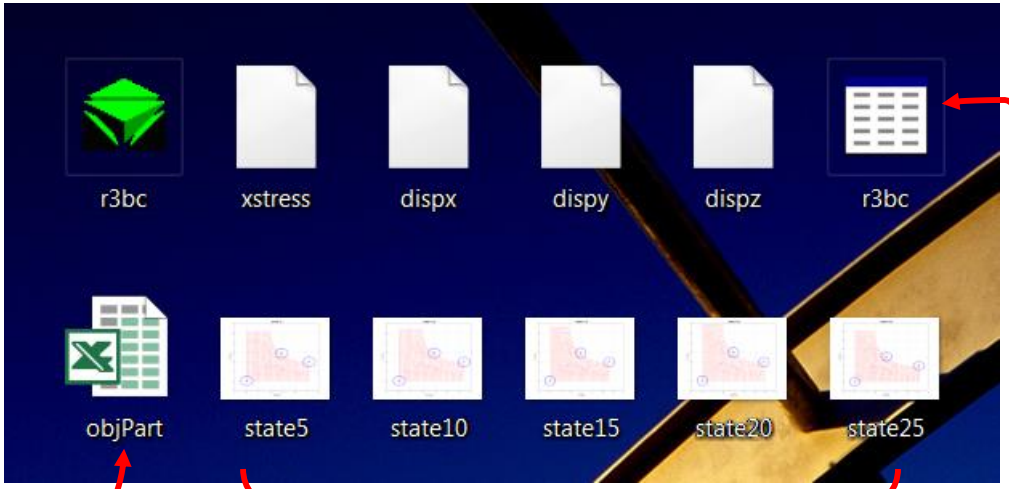


Most of the computations in objective partitioning are associated with the relocation of the partitions on the basis of the deformation of the target mesh

How to implement Objective Partitioning?

3. Performing Objective Partitioning through “ObjPart.m”

3.5. Identify outputs



objPart.xls

Output figures for visualization

Structure Array from “elementinfo.m” has been updated!

File saved in the directory of inputs