This document relates to the new release of the rbMIT System Part I Software. The old version is denoted Part I SV1 (SV1, for short) and the new version is denoted Part I SV2 (SV2, for short). This version SV2 of the Part I (parametrically coercive and compliant) software greatly simplifies the user input process. No changes are effected in the actual numerical algorithms.


We indicate below the changes in the user input process with reference to the detailed procedure described in Chapter 5 of the book. All section and page numbers refer to Part I BV1 Chapter 5.

1. Software Installation (Section 5.1.2, page 157): There are no changes, except that now the directories to be downloaded are `rbMIT_Library_PartI_SV2` and `rbMIT_Aux_PartI_SV2`. Before downloading/installing the SV2 software, the user should remember to first delete or move the current (SV1) contents from the user's `\rbMIT_System\rbMIT_Library` and `\rbMIT_System\rbMIT_Aux` folders.

2. Sections 5.2.2 and 5.2.3: There is no change in the procedure. (But note that the contents of the SV2 `\rbMIT_System\rbMIT_Aux\TBCh5Ex` directory (to be copied to the user's `\rbMIT_System\TBCh5Ex` directory) will be different from the contents of the corresponding SV1 directory — reflecting the changed input process.)

3. Section 5.3: There are no changes; the user should still create the directory `\rbMIT_System\*PROBNAME`. We recall that *PROBNAME is to be read everywhere it occurs as, that is, replaced with the name of the user's problem; for instance, for the example of Chapter 5, *PROBNAME is to be read as (i.e., replaced with) TBCh5Ex.

4. Section 5.4.1

   (a) At the very beginning of this section and before entering any of the Steps, the user should — from the `\rbMIT_System\*PROBNAME` directory — do
   ```matlab
   >> addpath('..\rbMIT_Aux')
   >> probname = ' *PROBNAME '  
   % note the user should replace *PROBNAME with the name of  
   % the user’s problem
   >> rbcopier_parcoer_compliant
   ```

   Note `probname` is a MATLAB® character variable which is set to the name/label of the user's problem; for example, if the user's problem is named (i.e.,
*PROBNAME is) LinElasticBlk, then the user should set

```matlab
>> probname = 'LinElasticBlk'
% recall the '...' indicate a character string
```

During Steps 1 and 2, `probname` should always be set in the MATLAB® workspace to the particular user problem of interest/under investigation. (Thus, if the user changes directories and works on Steps 1 and 2 for another problem, the user must change `probname` in the workspace accordingly. Note Step3 is done differently; see below.)

The `rbcopier_parcoer_compliant` script copies and renames all the necessary template and function files to the user’s directory. Hence no “copy/rename/rename/replace” operations are required in SV2 (unlike SV1).

(b) Page 169, Initialize (Step1): The user can skip the first paragraph entirely; everything has been initialized automatically by the `rbcopier_parcoer_compliant` script. The user can proceed to Edit and Execute and replace all the `?` with the user’s data. (The SV2 script will of course look slightly different from the SV1 script, but the user inputs are the same as for SV1.)

(c) Page 170, Define User-Supplied Functions: The user should skip the first paragraph entirely; everything has been initialized (in the `*PROBNAME` directory) automatically by the `rbcopier_parcoer_compliant` script.

(d) Pages 170–173: There is no change to these user-supplied functions. (The user should note that in SV2 the name of the functions as stated in the `.m` file will not bear the `*PROBNAME` prefix. However, the actual names of the `.m` files in the `*PROBNAME` directory will bear the `*PROBNAME` prefix. The user does not need to change the name of the function in the `.m` file.)

(e) Page 173, Initialize (Step2): The user can skip the first paragraph entirely; everything has been initialized automatically by the `rbcopier_parcoer_compliant` script. The user can proceed to Edit and Execute and replace all the `?` with the user’s data. (The SV2 script will of course look slightly different from the SV1 script, but the user inputs are the same as for SV1.)

(f) Page 175, Initialize (Step3): The user can skip the second and third (and fourth) paragraphs. In fact, note that in SV2 there are no problem-specific Online codes, and hence no Online codes in the `*PROBNAME` directory. The `Online_RB` and `Online_RB_mq` codes in `rbMIT_Library` serve all outputs.

(g) Pages 175–176: The user should no longer declare `*PROBNAME_PROBDEF` or `*PROBNAME_ONLINE` as `GLOBAL`. It suffices to simply `load` `*PROBNAME_PROBDEF` and `*PROBNAME_ONLINE`. (Also, recall that one load suffices: `Online_RB` can then be called as many times as desired without reloading — unless for some reason the `*PROBNAME_PROBDEF` and `*PROBNAME_ONLINE` structures are corrupted or “disappear” from the MATLAB® workspace.)

(h) Page 176: To obtain `[sN, DeltaN]` in SV2 the user should — from the `\rbMIT_System\*PROBNAME` directory — do

```matlab
>> [sN, DeltaN] = Online_RB(probname, muvectorvalue, N, epsdes)
```
Notes: 

(i) There is no *PROBNAME prefix to the Online_RB code (in fact, the Online_RB code resides in \rbMIT\Library and applies to all problems).

(ii) The first argument to Online_RB is the character variable probname (= ‘*PROBNAME’ = ‘the name of the user’s problem’). It is this first argument that tells the RB Online evaluator which problem is being considered. (It then simple to consider output evaluations for several different problems simultaneously, since probname is passed through the argument list and not through the shared MATLAB® workspace.) 

(iii) The remainder of the Online_RB arguments — muvectorvalue, N, epsdes — are to be set in SV2 exactly as described in Chapter 5 (pages 176–177) for SV1.

(i) Page 177. To obtain the Multiple Queries plot, the user should — from the \rbMIT\System*PROBNAME directory — do

```matlab
>>Online_RB_mq(probname, mu_index, mu_min, mu_max, muvectorvalue, N, epsdes)
```

where (just as in Online_RB) the character variable probname specifies the problem of interest, and the remaining arguments are set as for SV1 (pages 177–178).

5. Section 5.4.2: There are no changes from SV1.

6. Section 5.5: There are no changes from SV1. (Except one change, transparent to the user: In the Step1_parcoer_compliant script, a character variable *PROBNAME_PROBDEF.type is set to ‘parcoer_compliant’. This will serve when the software for later Parts is released, in order to make sure that each problem type (‘parcoer_compliant’, ‘coer’, ‘noncoer’, ‘parabolic’) calls the routines for that particular problem type.)

-----END of Software Release Notes: PartI_SV2 (3/16/07)-----