1. Exactly one argument can be passed to the entry point function. The entry point function to the input MATLAB program should take exactly one argument, whose type is described by the switch `-arg_info`. However, there is no restriction on the type or shape of this argument. None or more than one arguments will result in an exception and there will be no output.

2. All the functions is the input program must be in separate files. Each function of the input Matlab program must be present in a separate `.m` file.

3. Generated `Mix10.x10` library file does not contain all the library functions used by the program. We are still in the process of building the XML file used by the builtin-handler described in Section 2 of the paper. This means that for some builtins used in the Matlab program, the X10 implementation is not populated automatically in the generated library file. However, we have provided a hand-written implementation of the builtin library functions in the directory named `library`.

4. Add the `%!parfor` annotation for parfor loops The McLab frontend currently does not support the Matlab parfor construct. However, adding the annotation ”%!parfor” just before the parfor loop in the Matlab program will trigger the MiX10 compiler to generate the parallel X10 code for the parfor construct.

5. Multiple definitions of a function may exist. Due to a known bug in the Tamer [2], some functions are defined twice in the TamerIR and thus in the generated X10 code. The McLab team is in the process of fixing it.

1.3.5 Before compiling and running the generated X10 code

Once the X10 program is generated by the MiX10 compiler, there are three steps that need to be done in order to make the generated X10 code ready to run:

1. As noted in point 3 of section 1.3.4, The hand-wriiten X10 implementation of the builtin library functions is provided in the directory named `library` inside the root directory (`MiX10`) if the artifact. The `library` folder further contains two folders named `simpleArrayLib` and `regionArrayLib`, each containing two X10 files `Helper.x10` and `Mix10.x10`. To use these libraries, replace the generated directory `simpleArrayLib` (or `regionArrayLib`) in the `output` directory with the `simpleArrayLib` (or `regionArrayLib`) in the `library` directory. For example, from within the `output directory`, you can give the following set of bash commands to replace the `simpleArrayLib`:

```
        $ rm -r ./simpleArrayLib
        $ cp -r ../library/simpleArrayLib ./
```

2. The generated X10 code contains a main() method that needs to manually edited to add a call to the entry point function with a single argument that has the type and shape specified with the `-arg_info` switch while compiling the Matlab code with the MiX10 compiler. The argument should also be real/complex as specified. The main() method contains the comment, “`//Insert Call to driver function here`”, indicating where the call should be made. For example, in the bubble sort example, a call to the entry point function can be made by adding the following line below the above mentioned comment in the main() function: