The code solves the steady state in Julia and uses the BKM Method to solve the dynamics and the optimal taxes in the Transition.

The steady state of the model uses the Endogenous Grid Method (EGM), using the code of Alidstair McKay (All errors are ours).

**I - Julia code to solve for the steady state, the IRF and the comparison between the methods:**

* Main file « MIT\_simul.jl » solves for the optimal taxes in the transition using the distribution in the long run. It uses the following files:
  + « Aiyagari\_solve.jl » solve Aiyagari model.
  + « Parameters.jl » parameters of the Aiyagari model.
  + « MIT\_functions.jl » functions to generate the transition and the IRF in the BKM Method
  + Be sure the following files are in the same folder to generate the comparison between the methods:
  + « initial\_dis.jld2 » initial distribution in the long run.
  + « tofigtruncation.mat » impulse response functions from the variables in the Truncation method.
  + « truncation.jld2 ».
* Main file « MIT\_simul\_low.jl » solves for the optimal taxes in the transition using the distribution in the file « intial\_dist\_low.jld2 ». It uses the following files:
  + « Aiyagari\_solve.jl » solve Aiyagari model.
  + « Parameters.jl » parameters of the Aiyagari model.
  + « MIT\_functions.jl » functions to generate the transition and the IRF in the BKM Method
  + Be sure the following files are in the same folder to generate the comparison between the methods:
  + « initial\_dis\_low.jld2 » initial distribution in the long run to generate a low value for the optimal tax.
  + « tofigtruncation.mat » impulse response functions from the variables in the Truncation method.
  + « truncation.jld2 ».
* Main file « MIT\_simul\_large.jl » solves for the optimal taxes in the transition using the distribution in the file « intial\_dist\_large.jld2 ». It uses the following files:
  + « Aiyagari\_solve.jl » solve aiyagari model.
  + « Parameters.jl » parameters of the Aiyagari model.
  + « MIT\_functions.jl » functions to generate the transition and the IRF in the BKM Method
  + Be sure the following files are in the same folder to generate the comparison between the methods:
  + « initial\_dis\_large.jld2 » initial distribution in the long run to generate the high value for the optimal tax.
  + « tofigtruncation.mat » impulse response functions from the variables in the Truncation method.
  + « truncation.jld2 ».
* Main file « MIT\_simul\_shocks.jl » solves for the IRF and compare the results among the methods. It uses the following files:
  + « Aiyagari\_solve.jl » solve Aiyagari model.
  + « Parameters.jl » parameters of the Aiyagari model.
  + « MIT\_functions.jl » functions to generate the transition and the IRF in the BKM Method
  + Be sure the following files are in the same folder to generate the comparison between the methods:
  + « initial\_dis.jld2 » initial distribution in the long run.
  + « tofigtruncation.mat » impulse response functions from the variables in the Truncation method.
  + « tofigreiter.mat » impulse response functions from the variables in the Reiter method.
  + « tofigRA.mat » impulse response functions from the variables in the Representative agent economy.
  + « truncation.jld2 ».