

**Replication codes for the solution of the New Keynesian model subject to the
zero lower bound in “Occbin: a toolkit for solving dynamic models with
occasionally binding constraints easily”
by Luca Guerrieri and Matteo Iacoviello**

There are two folders in this replication packet.

The folder **runsim_sticky_zlb_1poly** contains fortran code for the model solution.

The file `make.inc` and `Makefile` need to be adapted to the local installation of Fortran. They are set up for compilation using `gfortran`. The last successful compilation was on a Silicon Mac with `gfortran 13.2` for Sonoma (macOS 14).

As set, `Makefile` can be used to wipe the old compilation and produce a new one; to that purpose, invoke these three commands:

```
make clean  
make minpack  
make
```

The main program is **runsim_sticky.f90**. The compiled program can be run invoking:
`./runsim`

Runtime is about 10 minutes on a Mac with an M2 chip. The program will save a decision rule which can be used to plot impulse response functions and solution errors using the matlab programs provided (described below).

The solution algorithm is sketched out in the appendix, also included in this packet.

The folder **matlab_sticky_zlb_1poly** contains Matlab programs that can be used to load the decision rule saved by the Fortran programs to compute impulse response function, model errors, or plot the decision rule.

The program **runsim_matlab** reproduces figure A of the appendix, which shows the approximation errors for the three expectation functions that characterize the solution. On line 14, set the switch **load_fortran_solution** to 0 to compute the solution in Matlab instead of loading it from the fortran program.

The Matlab program relies on the same solution algorithm implemented in Fortran, but the running times are longer.

To test the algorithm in Matlab, it can run more speedily using a coarse solution. For that purpose, for instance, choose second-order polynomials by setting `n=2` on line 65 and choose 11 states on line 71 --- the fortran solution is set to use polynomials of order 6 and a grid of 51 states.

Computing the coarse solution in Matlab only takes a couple of minutes on a Mac with an M2 chip.