

# Readme for Model Programs for “Labor Market Policy in the Presence of a Participation Externality”

Benjamin Griffy\*      Adrian Masters†

Aug. 2021

## 1 Main code

The code is divided among many functions, organized in a top-down structure. There is a single “Main.m” file that contains run options and can be used to change the set of results generated by the code. “Main.m” contains for structs (Matlab’s name for dictionary data structures) that change how the code will run. These structs are

- `opts`: choose how the code is run. This includes whether parameters are estimated (“SMM”), which versions of the model are solved (“solveLaissezFaire,” “solvePlanner,” etc.), which policy experiments are performed (“decompExperiments,” “policy UI,” etc.), and which results are generated (“SaveEstimOutput,” “CompDecompPolicies,” etc.). This also chooses the solution technique for the global parameter search (“solver”), whether to load the previous distance minimizing set of parameters (“params” to “loadparams”), and defines the paths for input and output.
- `params`: define each parameter to be used solving and simulating the model. Note that changing the “params” option in the “opts” struct to “loadparams” will change the “`smm_params`” to their distance minimizing values.

---

\*[bgriffy@albany.edu](mailto:bgriffy@albany.edu). Department of Economics, University at Albany SUNY, 1400 Washington Avenue, Albany, NY 12222, USA

†[amasters@albany.edu](mailto:amasters@albany.edu). Department of Economics, University at Albany SUNY, 1400 Washington Avenue, Albany, NY 12222, USA

- `smm_params`: the set of parameters used to minimize the distance in the SMM objective function.
- `SMMopts`: set the upper and lower bounds, and (in principle) the equality and inequality bounds for the SMM procedure.

After setting these structs, the code reads in the targeted moments and non-targeted quantities data using the “`get_data.m`” file. Then the “`get_params.m`” file reads in the previous “`SMM.csv`” results and finds the distance minimizing `smm_params` if the “`loadparams`” option is selected and replaces them in the struct “`params`”. If not, this returns the default parameters. Finally, this file runs “`Main_Wrapper`,” which implements these options and calls the appropriate files to execute the desired results.

## 2 Solving the model

We solve several versions of the model corresponding to different policies and market conditions. These are self-explanatory and distinguished by their suffixes (i.e., “`ModelUI.m`” is for the UI policy, “`ModelVac.m`” for the vacancy subsidy, and “`Model.m`” for everything else). There are also a set of programs that start with “`ModelMaxWelf...`” whose names conclude with their corresponding policies. These calculate total welfare over the transition from the initial steady state to the new steady state after a change in policy using the policy rules solved for in the “`Model...`” files.

## 3 Calibrating the model

The model is calibrated using the code contained in the “`SMM.m`” file. There is an initial “`SMM_wrapper.m`” that checks a couple boundary conditions. “`SMM.m`” first solves the model and then calculates the moments described in the paper, the unemployment rate, not-in-the-labor-force, the average from the log-wage distribution, the standard deviation from the log-wage distribution, and the spike at the minimum wage. Last, it calculates the difference between these moments and the empirical moments and calculates the sum of squares (the SMM objective function). Finally, it appends the moments to the “`SMM.csv`” file and the difference between this and the empirical targets in the “`SSR.csv`” file.

## 4 Generating the findings

Once the distance minimizing parameters are found, the “MainFindings.m” program organizes the main results of the paper. The first several conditionals solve various versions of the model, and their decompositions described in section 6. In the “CompDecompPolicies” conditional, we create the decomposition tables from section 6.2 of the paper. In the “CompPolicyExperiments” conditional, we compare welfare across the various policies in section 6.1. The “MainFindingsAverages\_Wrapper.m” program calculates the key aggregates of interest in the paper by calling the “MainFindingsAverages.m” program. This is essentially the same as any of the “ModelMaxWelf.m” files, but calculates additional quantities of interest. The rest of the “MainFindingsAverages\_Wrapper.m” program creates the results tables in the paper. The final program called is the “MainFindingsEstimResults\_Wrapper.m” program. This creates tables that compare the model generated moments to the empirical moments, the parameters, and a table that compares the non-targeted moments. It also generates plots that compare the wage distribution to the data.

## 5 Other notes

Parameters are defined locally using the “Parameters.m” program. This is preferable over global parameters, because they can be changed in subroutines without changing their value across other functions.