README for Code Packet
“Alternative Methods for Solving Heterogeneous Firm Models”
Stephen J. Terry
stephenjamesterry@gmail.com
January 2017

Basic Structure and Overview

This packet contains code written in Fortran and MATLAB which solves a simplified version of the Khan and Thomas (2008) model as well as various extensions using the KS, XPA, PARAM, REITER, and WINBERRY solution methods. Explanations of each piece of code are available at the top of each file, and a brief listing of the file structure follows. For convenience, compilation scripts for the Fortran code are provided, relying on the gfortran compiler. I cannot offer any detailed computer support beyond the inclusion of these scripts. Also, note that the REITER and WINBERRY solution algorithms rely on a call to MATLAB as well as a working copy of Chris Sims' `gensys` solver, which must be added to your MATLAB path.

Directories and Files

- KS
  - BASELINE
    - `kt_ks.f90`: main code for baseline KS solution
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
  - CONTINUOUS SHOCKS
    - `kt_ks_cont.f90`: main code for KS solution with continuous shocks
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
    - `libmacblas.a`: compiled BLAS library
    - `libmaclapack.a`: compiled LAPACK library
  - MAINTENANCE_INVESTMENT
    - `kt_ks_maint.f90`: main code for KS solution with maintenance investment
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
  - MICRO_VOLATILITY_FLUCTUATIONS
    - `kt_ks_vol.f90`: main code for KS solution with micro volatility fluctuations
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
  - SIZE_DEPENDENT_TAXES
    - `kt_ks_subsidy.f90`: main code for KS solution with size-dependent taxes

- `base_lib.f90`: module of utility functions
- `compile_script.sh`: compilation script

- MATLAB_FOR_GRAPHS_STATS
  - `graphs_and_stats.m`: code to gather results and produce figures
  - `hptrend.m`: code to HP-filter business cycle data
  - `pentle.m`: utility function for `hptrend.m`
  - `trile.m`: utility function for `hptrend.m`
- PARAM
  - `kt_param.f90`: main code for baseline PARAM solution
  - `base_lib.f90`: module of utility functions
  - `compile_script.sh`: compilation script
  - `libmacblas.a`: compiled BLAS library
  - `libmaclapack.a`: compiled LAPACK library
  - SS_CODE
    - `kt_ss.f90`: main code for steady-state solution for input to PARAM
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
    - `libmacblas.a`: compiled BLAS library
    - `libmaclapack.a`: compiled LAPACK library
- REITER
  - BASELINE
    - `kt_reiter.f90`: main code for baseline REITER solution
    - `call_gensys.m`: code for calling `gensys` solver
    - `call_simulate.m`: code for performing REITER unconditional simulation
    - `call_simulate_IRF.m`: code for performing REITER IRF simulation
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
    - `libmacblas.a`: compiled BLAS library
    - `libmaclapack.a`: compiled LAPACK library
    - SS_CODE
      - `kt_ss.f90`: main code for steady-state solution for input to REITER
      - `base_lib.f90`: module of utility functions
      - `compile_script.sh`: compilation script
      - `libmacblas.a`: compiled BLAS library
      - `libmaclapack.a`: compiled LAPACK library

  - MICRO_VOLATILITY_FLUCTUATIONS
    - `kt_reiter_vol.f90`: main code for REITER solution with micro volatility fluctuations
    - `call_gensys.m`: code for calling `gensys` solver
    - `call_simulate.m`: code for performing REITER unconditional simulation
    - `call_simulate_IRF.m`: code for performing REITER IRF simulation
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
  - SIZE_DEPENDENT_TAXES
    - `kt_reiter_subsidy.f90`: main code for REITER solution with size-dependent taxes
    - `call_gensys.m`: code for calling `gensys` solver
    - `call_simulate.m`: code for performing REITER unconditional simulation
    - `call_simulate_IRF.m`: code for performing REITER IRF simulation
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script

- WINBERRY
  - `kt_winberry.f90`: main code for baseline WINBERRY solution
  - `call_gensys.m`: code for calling `gensys` solver
  - `call_simulate.m`: code for performing WINBERRY unconditional simulation
  - `call_simulate_IRF.m`: code for performing WINBERRY IRF simulation
  - `base_lib.f90`: module of utility functions
  - `compile_script.sh`: compilation script
  - `libmacblas.a`: compiled BLAS library
  - `libmaclapack.a`: compiled LAPACK library
  - SS_CODE
    - `kt_ss.f90`: main code for steady-state solution for input to WINBERRY
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script
    - `libmacblas.a`: compiled BLAS library
    - `libmaclapack.a`: compiled LAPACK library

- XPA
  - `kt_xpa.f90`: main code for baseline XPA solution
  - `base_lib.f90`: module of utility functions
  - `compile_script.sh`: compilation script
  - SS_CODE
    - `kt_ss.f90`: main code for steady-state solution for input to XPA
    - `base_lib.f90`: module of utility functions
    - `compile_script.sh`: compilation script