**READ ME**

Version: 1.0

Date: March 19, 2019

Author: Michael Springborn (mspringborn@ucdavis.edu) and Amanda Faig

License: [GPL-3](https://www.gnu.org/licenses/gpl-3.0.en.html)

Operating system and software:

* This code was developed in Matlab (Version 2018a) and uses parallel computing (optional) and the Statistics and Machine Learning Toolbox (for the fitrgp function to run the Gaussian process regression).
* Software was run on a PC with an Intel Core i7-4770 Quad-Core processor (3.4 GHZ), 16GB installed RAM and a Windows 10 Enterprise operating system.

Understanding prefixes on Matlab m-file names and inline functions in the code:

* The prefix “sc\_” indicates a script file.
* The prefix “f\_” indicates a function file.
* The prefix “fi\_” (in the code) indicates an inline function.

Overall procedure:

* Run value function solution for all desired cases
  + Shock options:
    - simple model, single non-correlated growth shock (multTCorrShk = false)
    - four autocorrelated shocks model (multTCorrShk = true)
      * Can also turn any of the 4 individual shocks off
      * Set m, level of autocorrelation (e.g. 0, .95)
  + Growth model options:
    - logistic (‘base’); logistic with critical depensation (‘critdep’); Ricker (‘rick’)
  + Harvest cost model options:
    - MC decreasing in stock (default, doConstMCHarv = 0); constant MC (doConstMCHarv = 1)
  + Solution method
    - VFI [only for simple, one-shock model with single state] (multTCorrShk = false)
    - ADP
      * Value function model options:
        + Nonparametric (preferred)
        + Parametric

Select polynomial model (‘quadratic’, ‘cubic’, ‘quartic’)

* + - * A “dashboard” tracking figure is produced with each updating/regression step and is saved in: **\output\working\_figs\{casename}\dashplots***.* These figures are essential to tuning the ADP algorithm.
  + All solution files (value function, policy function, simulations) are stored to: **\output\solutions\soln\_{casename}**
* Calculate and consider policy functions
* Run simulations (for multiple shock model) to assess importance of accounting for system complexity (autocorrelation.

**Outline of code files (including scripts and functions)**

Core code files are in **bold**. Others are small helper scripts/functions. Within the code “FDP” stands for “forward dynamic programming”, which is our own labeling. We use the more common/established label “approximate dynamic programming” or “ADP” in the manuscript.

1. **main** (set options, run value function solution)
   1. **sc\_set\_bioecon\_params\_sf** (sets bioecon parameters)
   2. sc\_setparllwrks (set up parallel pool)
   3. sc\_comparemodels (to compare output from multiple FDP solutions (to check reliability))
   4. Solution runs either with VFI (simple model only: ~multTCorrShk)
      1. **sc\_VFI\_sf**
      2. **sc\_FDP\_parallel\_sf**
         1. Setup
            1. **sc\_FDP\_setup** (Setup: dynamic step size, simulation parameters, and regression model)
            2. **sc\_FDP\_shocks** (Setup: exogenous random shocks and variable paths)
         2. Profit and dynamics: conditional on action, get profit and new state
            1. f\_nextperiod\_sf
            2. f\_nextperiod\_multshock\_sf
         3. Regression:
            1. sc\_param\_reg\_sf (parametric)
            2. **sc\_nonpar\_reg\_sf\_fitrgp** (nonparametric)
         4. Plots:
            1. **sc\_plots\_sf** (figures to visualize elements of solution process)
         5. Save output if parfor fails (for debugging)
            1. f\_parForFailSave
2. Generate results and plots
   1. Simple single shock model
      1. **sc\_plot\_V\_and\_policy\_final** (generate value and policy plots for a given systmodel for simple single growth shock model)
   2. Multiple correlated shock model
      1. **sc\_analysis\_of\_solution\_sf** (Look at multiple correlated shock (multCorrShk) model output. Generate policy function; run sims for performance comparison; generate policy function figs.)
         1. **sc\_getpolfun\_sf** (Use value function solution to calculate policy function solution. Functions are continuous but policy function is calculated at a discrete set of nodes for plotting/examination.)
         2. f\_cleanPolicy (ensures regularity of policy function. Ultimately did not use (except to reshape policy function array).)
         3. sc\_setparllwrks; (see previous use under **main.m**)
         4. **sc\_runSerCorrSims**;
            1. f\_nextperiod\_multshock\_sf (see previous use under **main.m**)
         5. **sc\_PolFunFigs\_multTCorrShk** (generate policy function figs for multTCorrShk case)