Description of Code/Data

This document describes the MATLAB code files contained in `LCSolutionMethodCode.zip`. See Galizia (2021), "Saddle Cycles: Solving Rational Expectations Models Featuring Limit Cycles (or Chaos) Using Perturbation Methods" for further details.

1 First-Order Approximations

The MATLAB function `InvSubGen.m` provides code to compute first-order approximations to candidate solutions. These are the solutions of the form

$$\begin{pmatrix} y' \\ z \end{pmatrix} = \begin{pmatrix} \pi_y \\ \psi_y \end{pmatrix} y + \begin{pmatrix} \pi_\theta \\ \psi_\theta \end{pmatrix} \theta$$

to (see equation (19)-(20) in the text) the equations

$$\Gamma_x x + \Gamma_{x'} \mathbb{E}\left[x'\right] + \Gamma_\theta \theta + \Gamma_{\theta'} \mathbb{E}\left[\theta'\right] = 0$$

$$\mathbb{E}\left[\theta'\right] = \Pi_\theta \theta \,.$$

2 Replication Code for Application in Section 5

2.1 Main Results for Perturbation Approximation

The folder `MainCode` in `LCSolutionMethodCode.zip` contains code to replicate the perturbation results for the application of Section 5. See Online Appendix for details of solution implementation. The main file to run is `BGPEx.m`, which produces all the relevant figures. `BGPEx.m` uses the following files:

- `fn_BGPEx_Sol.m`: Compute Taylor coefficients for the non-linear approximation.
- `fn_BGPEx_e.m`: Compute Taylor approximation of e for given state variables.
- `fn_BGPEx_EulErr.m`: Compute Euler errors.
- `fn_BGPEx_prsStr.m`: Convert model parameters from vector to struct variable, or the reverse.

- fn_BGPEx_SpecSim.m: Compute spectrum of e for model.
- fn_BGPEx_thsim.m: Simulate sequence of stochastic exogenous variables.
- fn_BGPEx_XYESim.m: Simulate sequence of X, Y, and e.
- The following files which were automatically generated using MATLAB's Symbolic Toolbox and are used to solve the model (see explanation of fn_BGPEx_Dervs.m below):
  - fn_BGPEx_SS.m: Get steady state of model.
  - fn_BGPEx_FirstOrderMat.m: Get coefficient matrices for first-order approximation of system.
  - fn_BGPExDerv_ord2.m, fn_BGPExDerv_ord3.m: Used to sequentially obtain 2nd- and 3rd-order derivatives for approximation of solution.
  - fn_BGPEx_QDerv_ord3.m: Get 3rd-order approximation of Q function.
- InvSubGen.m: Obtain linear approximations (see above).
- TaylExp.m: Return information about form of monomials in Taylor expansion.
- TaylMaps.m: Return information about various mappings between different forms of Taylor expansions (used in simulations).
- TaylSim_tX.m: Compute monomials of Taylor approximation for given state variables using Kronecker (tensor) product method.
- rgb.m: File used for colors of plots.

The five automatically generated functions listed above were created using the included function fn_BGPEx_Dervs.m. To re-generate those files, call this function with the argument 3 (indicating 3rd-order approximation). In principle this file can also be used to generate relevant functions for alternative approximation orders, though using the resulting output would require modifications to BGPEx.m and the functions called by it.

Also included in the folder are the following MATLAB data files:

- pars_mu.mat, pars_z.mat: Files containing parameters for the μ-shock and z-shock models, respectively.
- Spec.mat: File containing spectrum for hours data.
- EE_FE.mat: File containing policy function and Euler errors for model solved using finite elements method (see below).

2.2 Code for Finite Elements Solution

The folder `FECode` contains the code to obtain the finite elements (FE) solution. The main file to run is BGP_FE.m, which (when the appropriate options are selected) generates a `.mat`-file containing cuts of the policy function for e and of the Euler errors for the FE solution, which are used in Figures 9 and 11 of the paper.

- `fn_cm.m`: Get array of indices corresponding to all possible vectors of the form (x_1, x_2, x_3) with $x_j = 1$ or 2.
- `fn_W.m`: Compute reduced-form weights (i.e., elements of the arrays $\boldsymbol{W}_{ijk}$; see Online Appendix) and associated gridpoints, as well as several other variables that will be useful in computing the FE solution.
- `fn_Obj.m`: Compute the objective function for the FE solution method.
- `fn_Psi.m`: Compute the value of $\Psi_{ijk}(s)$ (i.e., an individual element) for given indices i, j, k and state s.
- `fn_PsiA.m`: Compute the value of a tent function for given inputs, and upper or lower bound for the support of the tent function, the length of that support interval, and a flag for which side of the tent function peak the inputs are on.
- `fn_e.m`: Compute the value of the policy function for e for given states and a candidate FE solution.
- `fn_ekap_mex.mexw64`, `fn_ekap.m`: Function that undertakes the multiplication of basis functions by the given FE coefficients to obtain e.
  - `fn_ekap.m` contains the MATLAB code for this function, which is not actually used in the included version of the code.
  - `fn_ekap_mex.mexw64` is a `mex`-function compiled for 64-bit Windows 10 machines from `fn_ekap.m` automatically using MATLAB Coder. This is the function called in the included code (called from `fn_e.m`).
  - Using the compiled `mex`-function improved computation times for e (given a candidate FE solution) by around 30%.
  - For computer architectures/operating systems other than 64-bit Windows 10, the `mex`-function would need to be re-compiled. Alternatively, if computation speed is not a priority, the call to this function in `fn_e.m` can be replaced by a call to the MATLAB function `fn_ekap.m`.
- `fn_R.m`: Compute the value of the Euler residuals for a given candidate FE solution.

- `fn_ebnd.m`: Compute the value of the policy function for e (as done in `fn_e.m`), but subject to the constraints that certain endogenous variables (e..g, e itself, as well as the argument of the utility function) must be positive. At an actual FE solution, the results from `fn_e.m` and `fn_ebnd.m` will coincide, but it is possible that, without explicitly imposing the relevant constraints, certain candidate solutions could put e out of allowable bounds, which will cause errors in the code elsewhere. `fn_ebnd.m` ensures this doesn't happen.
- `fn_spen.m`: Compute the value of the next-period endogenous state (X', Y') given the current state $s = (X, Y, M)$ and the current e.
- `fn_Om.m`: Compute the value of Ω (see Online Appendix) at given states using the candidate FE solution.
- `fn_BGPFE_EulErr.m`: Compute Euler errors for FE model.
- `fn_BGPEx_prsStr.m`: See description from perturbation solution above.
- `fn_BGPEx_SS.m`: See description from perturbation solution above.
- `legzo.m`: Function to compute nodes and weights of a given Legendre polynomial (used for Gauss-Legendre quadrature).

Also included in the folder are the following MATLAB data files:

- `pars_mu.mat`: Files containing parameters for the μ-shock model.
- `kap_20_20_15.mat`: FE solution shown in paper for μ-shock model.
- `GH.mat`: Gauss-Hermite quadrature nodes and weights.
- `LCvc.mat`: File containing ranges of state variables to use for policy function and Euler error cuts.