README

Summary

This code solves the dynamic programming technique described in the working paper `The shadow costs of illiquidity' by Kristy A.E. Jansen and Bas J.M. Werker (link).

File structure

1. The file `shadow_costs_illiquidity.m' is the main file.
2. The files `focs_NT.m' and `focs_T.m' are two additional functions and called by the main file.
   - `focs_NT.m': computes the first order derivatives for consumption and the allocation to the liquid risky asset for the illiquid case (2 first-order conditions).
   - `focs_T.m': computes the first order derivatives for consumption, and both the allocation to the liquid and illiquid risky assets for the liquid case (3 first-order conditions).

The derivations of the first-order conditions and the H-functions can be found in the paper.

Running the code

- Simply press run into the main file `shadow_costs_illiquidity.m' .
- Licenses used:
  - Matlab (MATLAB_R2019b)
  - Optimization toolbox
  - Statistics toolbox
  - Curve fitting toolbox

Important notes

- In case of extreme parameter choices (correlations close to 1, high prices of risk, high liquidity shocks etc.), the choice of the initial parameter values and the options how to search for the optimal solution should be carefully reconsidered.
- Failure of the dynamic optimization problem is easily detected by evaluating the exitflag and/or sum of squared residuals at each grid search (link).
- To speed up the dynamic programming technique, the options set to solve lsqnonlin in Matlab can be changed to allow for fewer evaluations or function iterations. Of course, this comes at cost of less accuracy. This is particularly true for extreme parameter choices (correlations close to 1, high prices of risk, high liquidity shocks etc.).

Contact details

- Email: k.a.e.jansen@tilburguniversity.edu