Guide to code for "On the Optimality of Differential Asset Taxation"

Thomas Phelan
Federal Reserve Bank of Cleveland

January 3, 2025

1 Introduction

This document provides a guide to the code used to produce the figures in the paper "On the Optimality of Differential Asset Taxation." All code is written to run on Python 3.11.1 and is located on the github page for the paper. If you have questions or comments please email me at tom.phelan@clev.frb.org.

2 Notation from the paper

To explain the code construction I recall some algebra from the paper. In Section 2 of the paper I defined a candidate value function $v(u) = \overline{v}u$, where the constant $\overline{v}$ is given by

$$\overline{v} = \max_{\substack{\overline{c}, x \geq 0, x\overline{c} \leq \overline{\omega} \\ -\ln \overline{c} + x^2/2 < 1}} \frac{(Sx-1)\overline{c}}{\rho(1 + \ln \overline{c} - x^2/2)} \tag{1}$$

where $\overline{\omega} = \sqrt{\rho}\phi\sigma/(\rho\iota)$ and $S := (\Pi - \rho - \tau_I)/(\sqrt{\rho}\phi\sigma)$. In Appendix A I defined $\overline{x}$ and $\overline{\overline{x}}$ to be the solutions to $\overline{x}e^{\overline{x}^2/2} = \overline{\omega}$ and $\overline{\overline{x}}e^{\overline{\overline{x}}^2/2-1} = \overline{\omega}$, respectively. Following the explicit maximization in the proof of Proposition 2.1, the right-hand side of (1) may be written as

$$\overline{v} = \max_{x \in [0, \overline{\overline{x}}]} g(S, \overline{\omega}, x)h(S, \overline{\omega}, x) \tag{2}$$

where g and h are given by

$$\begin{aligned} g(S, \overline{\omega}, x) &= \frac{1}{\rho}(Sx - 1)e^{x^2/2} \\ h(S, \overline{\omega}, x) &= 1_{x < \overline{x}(\overline{\omega})} + 1_{x \geq \overline{x}(\overline{\omega})} \frac{(\overline{\omega}/x)e^{-x^2/2}}{1 + \ln(\overline{\omega}/x) - x^2/2}. \end{aligned} \tag{3}$$

In the paper, the stationary form of the resource constraint is given by

$$(1-\psi)\overline{C}(S)+\psi=(S\sqrt{\rho}\phi\sigma/\alpha+\rho_S/\alpha+(1/\alpha-1)\delta)(1-\psi)\overline{K}(S) \tag{4}$$

where the expressions for consumption and capital are given in Appendix B.3 of the paper

$$\overline{C}(S)=\frac{\rho_D\overline{c}(S,\overline{\omega})}{\rho_D-\mu_c(S,\overline{\omega})} \qquad \overline{K}(S)=\frac{\rho_D\overline{c}(S,\overline{\omega})x(S,\overline{\omega})}{(\rho_D-\mu_c(S,\overline{\omega}))\sqrt{\rho}\phi\sigma}.$$

The efficient value of S, denoted $\hat{S}$, is therefore a root of the equation $f(S)=0$, where

$$\begin{aligned} f(S) &= (1-\psi)\frac{\rho_D\overline{c}(S,\overline{\omega})}{\rho_D-\mu_c(S,\overline{\omega})}+\psi \\ &\quad -((S\sqrt{\rho}\phi\sigma+\rho_S)/\alpha+(1/\alpha-1)\delta)(1-\psi)\frac{\overline{c}(S,\overline{\omega})x(S,\overline{\omega})}{\sqrt{\rho}\phi\sigma}\times\frac{\rho_D}{\rho_D-\mu_c(S,\overline{\omega})}. \end{aligned} \tag{5}$$

3 Code construction

The sole class constructor for the paper is entitled captax and is located in classes.py. It includes the following 19 methods, where omegabar is $\overline{\omega}$ and Pi is Π:

- Principal-agent setting (11 methods):
  - xbar(omegabar) and xbarbar(omegabar): $\overline{x}$ and $\overline{\overline{x}}$.
  - g(S,omegabar,x) and h(S,omegabar,x): the functions in (3).
  - x(S,omegabar) and cbar(S,omegabar): $x(S,\overline{\omega})$ and $\overline{c}(S,\overline{\omega})$ from the main text.
  - mu_c(S,omegabar) and sig_c(S,omegabar): mean and volatility of consumption growth, denoted μ_c and σ_c in the main text.
  - S(Pi,phi): the variable defined in the principal's problem, $(\Pi-\rho_S)/(\sqrt{\rho}\phi\sigma)$.
  - nu_B(S,omegabar) and nu_K(S,omegabar,Pi): the wedges from the partial equilibrium setting.
- Characterization of stationary efficient allocations (3 methods):
  - f(S,omegabar,phi): defines the function (5).
  - S_hat(phi) and Pi_hat(phi): $\hat{S}$ and $\hat{\Pi}$ from Section 3 of the text.
- Quantities pertaining to decentralization (3 methods):
  - omegabar_d(S,omegabar): the constant $\overline{\omega}_{\mathrm{d}}$ in the decentralization ("d" subscript reminds the reader that this is different from the $\overline{\omega}$ value in the principal's problem).

  - r_b(Pi, phi): efficient cost of borrowing.
  - revenue_benchmark(Pi,phi,omegabar): total revenue raised in benchmark decentralization as fraction of capital income.

- Miscellaneous:

  - check1(S,omegabar) and check2(S,omegabar): two checks corresponding to the assumptions in A.1 and A.3 of the appendix, which together imply Assumption A.2 and ensure that the principal's value function is negative-valued (and hence finite valued even in when the principal can use lotteries).

4 Figures generation

There are five scripts necessary for the replication of the paper: main.py, classes.py, parameters.py, tight.py and relaxed.py.

- main.py: runs tight.py and relaxed.py.
- parameters.py: defines the parameters used in the numerical examples.
- classes.py: contains the class constructor used in the paper.
- tight.py: produces figures pertaining to the case in which collateral constraints are tight ($\bar{\iota} = 1.0$).
- relaxed.py: produces figures pertaining to the case in which collateral constraints are relaxed ($\bar{\iota} \approx 0.5$).

Each of above scripts also checks that the assumptions in the appendix are satisfied and prints this when running.