Online Appendix to "On the Optimality of Differential Asset Taxation"

Thomas Phelan
Federal Reserve Bank of Cleveland

August 22, 2022

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 in Python 3.6.5 and is located at https://github.com/tphelanECON/diff_cap_tax. If you spot errors or have questions please email me at tom.phelan@clev.frb.org.

Preliminaries

To explain the code construction I recall some algebra from the paper. In Section 2 of the paper I defined a candidate value function

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

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 2.3, 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

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

The stationary form of the resource constraint given in Assumption 3.1

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

may be simplified using the expressions in Appendix B.3,

$$\begin{aligned}\overline{C}(S) &= \frac{\rho_D\overline{c}(S,\overline{\omega})}{\rho_D-\mu_c(S,\overline{\omega})}\\ \overline{K}(S) &= \frac{\rho_D\overline{c}(S,\overline{\omega})x(S,\overline{\omega})}{(\rho_D-\mu_c(S,\overline{\omega}))\sqrt{\rho}\phi\sigma}\end{aligned} \tag{5}$$

to yield

$$\begin{aligned}&\alpha\sqrt{\rho}\phi\sigma\left((1-\psi)\frac{\rho_D\overline{c}(S,\overline{\omega})}{\rho_D-\mu_c(S,\overline{\omega})}+\psi\right)\\ &=(S\sqrt{\rho}\phi\sigma+\rho_S+(1-\alpha)\delta)(1-\psi)\frac{\rho_D\overline{c}(S,\overline{\omega})x(S,\overline{\omega})}{\rho_D-\mu_c(S,\overline{\omega})}\end{aligned} \tag{6}$$

which is how the root is found numerically.

Code construction

The sole class constructor for the paper is entitled captax and is located in classes.py. It contains the following methods (in the following, omegabar is $\overline{\omega}$):

- 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 c(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.
- omegahat(self,S,omegabar): the constant $\hat{\omega}$ in the collateral constraint in the benchmark decentralization.
- S(Pi,phi): $(\Pi-\rho_S)/(\sqrt{\rho}\phi\sigma)$.
- f(S,omegabar,phi): finds root of the equation (6).
- S_hat(phi) and Pi_hat(phi): $\hat{S}$ and $\hat{\Pi}$ from Section 3 of the text.
- r(Pi,phi): interest rate in benchmark case (no private risk-sharing).

- r_pe(Pi,phi): interest rate with private risk-sharing.
- taus(Pi,phi) and tausW(Pi,phi): entrepreneur and worker taxes in benchmark case (no private risk-sharing).
- taus_pe(Pi,phi) and tausW_pe(Pi,phi): entrepreneur and worker taxes with private risk-sharing.
- nu_B(S,omegabar) and nu_K(S,omegabar,Pi): the wedges from the partial equilibrium setting.
- check1(S,omegabar) and check2(S,omegabar): two checks corresponding to the assumptions in A.1 and A.2 of the appendix, which together ensure that the principal's value function is finite-valued.

Figures and methods computed under the assumption of private risk-sharing have a suffix _pe (for "private equity"). Figures with no suffix assume that the collateral constraint is set at the most relaxed value possible, while figures with suffix 1.0 corresponds to $\bar{\iota}$ (which gives the tightest collateral constraints possible when $\phi = 1$).

Figures generation

There are six scripts necessary for replication: main.py, classes.py, parameters.py, tight.py, relaxed.py, and private_equity.py.

- main.py: runs tight.py, relaxed.py and private_equity.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 ($\bar{\iota} = 1.0$) for the benchmark decentralization. In particular, this produces Figure 1 and Figure 2 from the main text.
- relaxed.py: produces figures pertaining to the case in which collateral constraints are relaxed ($\bar{\iota} \approx 0.5$) for the benchmark decentralization. In particular, this produces Figure 3 from the main text as well as the wedges in Figure 4 of Appendix E.

- private_equity.py: produces figures for taxes and interest rates for the market structure in which there is private risk-sharing. In particular, this produces Figure 5, 6 and 7 of Appendix E.