Lab 5

Introduction to Python Programming for Economics & Finance

Richard Foltyn
University of Glasgow

June 2, 2023

Contents

1 Household problem with endogenous labour supply

Consider the following infinite-horizon household problem with an endogenous labour supply choice which is an extension of the material we covered in the lecture,

$$
\begin{aligned}
V(y,a) &= \max_{c,\ell,a'} \left\{ u(c,\ell) + \beta \mathbb{E}\left[V(y',a') \mid y\right] \right\} \\
\text{s.t.} \quad c + a' &= (1+r)a + yw\ell \\
y' &= \rho y + \nu' \qquad \nu' \sim N\left(0,\sigma^2\right) \\
c &\geq 0,\ a' \geq 0,\ \ell \in [0,1]
\end{aligned}
$$

where utility is given by

$$
u(c,\ell) = \frac{c^{1-\gamma}-1}{1-\gamma} + \chi \log(1-\ell)
$$

In addition to c and a', the household now also chooses labour $\ell \in [0,1]$ and values leisure $1-\ell$ in its utility function. The interest rate r and the wage rate w are taken as given by households.

This problem has an additional intratemporal first-order condition which relates optimal labour supply to optimal consumption and is given by

$$
\frac{\frac{\chi}{1-\ell}}{c^{-\gamma}} = wy
$$

As usual, this equates the MRS to relative prices where the price of consumption is unity by assumption. For a given consumption level we can therefore obtain the optimal *interior* labour supply as

$$
\ell = 1 - \frac{\chi}{wy} c^{\gamma}
$$

if this expression is in $[0,1]$. Otherwise, labour supply assumes one of the boundary solutions.

You are asked to modify the solution algorithm from the lecture to solve this problem numerically using the following parametrisation:

Parameter	Description	Value
β	Discount factor	0.96

Parameter	Description	Value
γ	Coef. of relative risk aversion	1.5
χ	Weight on leisure term	1.0
r	Interest rate	0.04
w	Wage	1.0
ρ	Autocorrelation of labour income	0.95
σ	Conditional std. dev. of labour income	0.20
N_y	Number of states for Markov chain	3

As before, you should discretise the AR(1) labour income process to three states using the Rouwenhorst method.

Modified Parameters class

Update the definition of the `Parameters` class to include the additional parameters:

```
@dataclass
class Parameters:
    ....
    chi = 1.0
    w = 1.0
    ....
```

VFI with grid search

The most straightforward way to implement endogenous labour supply is to use grid search over a 2-dimensional choice set.

To this end, create a grid of candidate labour supply levels,

```
grid_l = np.linspace(0.0, 1.0, 11)
```

and add it a an additional attribute to the `Parameters` class:

```
@dataclass
class Parameters:
    ...
    grid_l = None       # Grid for labour supply choices
    ...
```

Modify the grid search algorithm we covered in the lecture to accommodate this additional choice. The easiest way to do that is to insert an innermost loop over candidate ℓ:

```
for il, lab in enumerate(par.grid_l):
    # perform same steps as before
```

and for each `il` save the expected utility and the saving choice. After going through all labour supply choices, but the one with yields the highest utility and save it as the optimal choice.

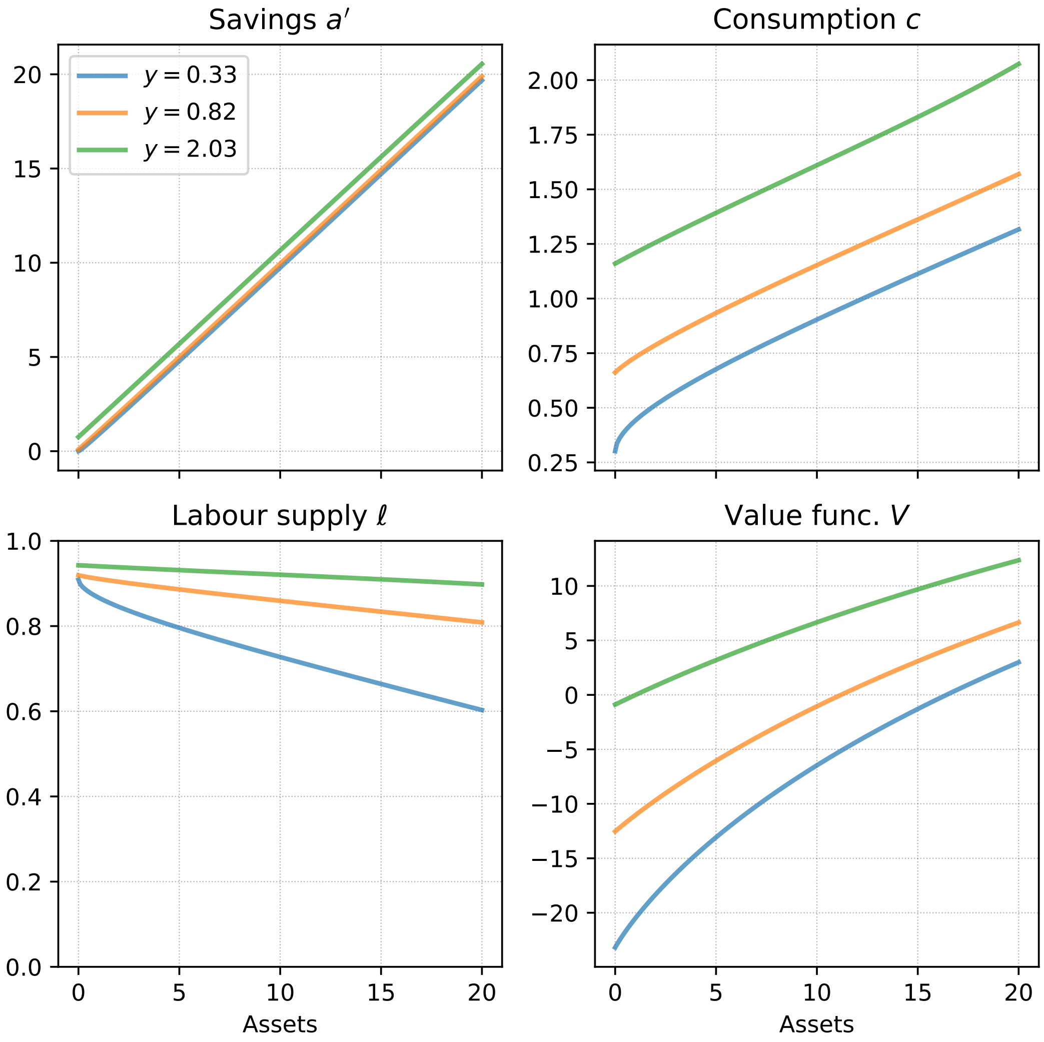
Alternatively, to solve the problem in a vectorised fashion, you can evaluate all choices over (a', ℓ) jointly as a 2-dimensional matrix.

Plotting the solution

Augment the plotting code we used in the lecture to additional show the optimal labour supply choice as a function of assets.

1.0.1 Solution

Your final graphs should resemble the ones shown below. These were created using VFI with quadratic interpolation, so your grid search solution will not be as smooth.


Savings a'
Consumption c
Labour supply ℓ
Value func. V
$y = 0.33$
$y = 0.82$
$y = 2.03$
Assets
Assets