ASSIGNMENT I

Vision: This project teaches you to solve for the *stationary equilibrium* and *transition path* in a heterogeneous agent model and analyze the economic results.

- **Problem:** The problem consists of
    1. A number of questions (page 2)
    2. A model (page 3-4)
- **Code:** The problem is designed to be solved with the *GEModelTools* package.
- **Structure:** Your project should consist of
    1. A single self-contained pdf-file with all results
    2. A single Jupyter notebook showing how the results are produced
    3. Well-documented *.py* files
- **Hand-in:** Upload a single zip-file on Absalon (and nothing else)
- **Deadline:** 6th of October 2023
- **Exam:** Your Assignment I will be a part of your exam portfolio.
  You can incorporate feedback before handing in the final version.

HANC with multiple types of labor

a) **Setup.** Define the stationary equilibrium, the transition path and create a DAG for the model.

b) **Solve for the stationary equilibrium.** Discuss and illustrate which factors determines wealth inequality.

c) **Compute and inspect the Jacobians of the household block wrt. φ_1.**

d) **Solve for the transition path when φ_{1t} is 10 percent higher for 10 periods.** Discuss which types of agents this benefits.

e) **Solve for the transition path when φ_{1t} is *permanently* 10 percent higher.** Discuss which types of agents this benefits.

1. Model

Households. The model has a continuum of infinitely lived households indexed by $i \in [0,1]$. Households are *ex ante* heterogeneous in terms of their discount factors, β_i, and their ability, χ_i. The discount factors are drawn with equal probabilities from a three element set, $\beta_i \in \{\breve{\beta} - \sigma_\beta, \breve{\beta}, \breve{\beta} + \sigma_\beta\}$. The abilities are either low or high, $\chi_i \in \{0,1\}$, with probabilities $\frac{2}{3}$ and $\frac{1}{3}$.

Households choose consumption and exogenously supply two types of labor, η_i^j for $j \in \{0,1\}$ with associated productivity φ_t^j. Savings is in terms of capital, which is rented out to firms at the rental rate, r_t^K. There are no possibilities to borrow. Households are *ex post* heterogeneous in terms of their stochastic labor productivity, s_{it}, and their (end-of-period) savings, a_{it-1}. The distribution of households over idiosyncratic states is denoted $\underline{\boldsymbol{D}}_t$ before shocks are realized and $\boldsymbol{D}_t$ afterwards. The real wages are w_t^j, and real-profits are Π_t.

The household problem is

$$\begin{aligned}
v_t(s_{it}, a_{it-1}) &= \max_{c_t} \frac{c_{it}^{1-\sigma}}{1-\sigma} - \nu \frac{\left(\sum_{j=0}^{1} \eta_i^j\right)^{1+\varepsilon}}{1+\varepsilon} + \beta_i \mathbb{E}_t\left[v_{t+1}(s_{it+1}, a_{it})\right] \qquad (1) \\
\text{s.t. } a_{it} + c_{it} &= (1 + r_t^K - \delta) a_{it-1} + \sum_{j=0}^{1} w_t^j \varphi_t^j \eta_i^j s_{it} + \Pi_t \\
\log s_{it+1} &= \rho_s \log s_{it} + \psi_{it+1}, \;\; \psi_{it+1} \sim \mathcal{N}(\mu_\psi, \sigma_\psi),\ \mathbb{E}[s_{it}] = 1 \\
a_{it} &\geq 0.
\end{aligned}$$

The Euler-equation is

$$c_{it}^{-\sigma} = \beta_i \mathbb{E}\left[v_{a,it+1}(s_{it+1}, a_{it})\right] \qquad (2)$$

$$v_{a,it} = (1 + r_t^K - \delta) c_{it}^{-\sigma}. \qquad (3)$$

The aggregate quantities of central interest are

$$C_t^{hh} = \int c_{it} d\boldsymbol{D}_t \qquad (4)$$

$$L_t^{j,hh} = \int \varphi_t^j \eta_i^j s_{it} d\boldsymbol{D}_t \;\text{ for } j \in \{0,1\} \qquad (5)$$

$$A_t^{hh} = \int a_{it} d\boldsymbol{D}_t \qquad (6)$$

Firms. A representative firm rents capital, K_{t-1}, and hires both types of labor, L_t^1 and L_t^2, to produce goods, with the production function

$$Y_t = \Gamma_t K_{t-1}^{\alpha} \Pi_{j=0}^{1} \left(L_t^j \right)^{\frac{1-\alpha}{2}} \tag{7}$$

where Γ_t is technology and α is the Cobb-Douglas weight parameter on capital. Capital depreciates with the rate $\delta \in (0,1)$. The real rental price of capital is r_t^K and the real wages are w_t^j. Profits are $\Pi_t = Y_t - \sum_{j=0}^{1} w_t^j L_t^j - r_t^K K_{t-1}$. The households own the representative firm in equal shares.

The law-of-motion for capital is $K_t = (1-\delta)K_{t-1} + I_t$.

Market clearing. Market clearing implies

1. Asset market: $K_t = A_t^{hh}$
2. Labor market: $L_t^j = L_t^{j,hh}$
3. Goods market: $Y_t = C_t^{hh} + I_t$

2. Calibration

1. **Preferences:** $\sigma = 2$, $\breve{\beta} = 0.975$, $\sigma_\beta = 0.01$, $\nu = 0.5$, $\varepsilon = 1.0$
2. **Labor supply**: $\eta_i^j = \begin{cases} 1 & \text{if } \chi_i = j \\ 0 & \text{else} \end{cases}$, $\varphi_{ss}^0 = 1$, $\varphi_{ss}^1 = 2$
3. **Income process:** $\rho_s = 0.95$, $\sigma_\psi = 0.30\sqrt{(1-\rho_s^2)}$,
4. **Production:** $\Gamma_{ss} = 1$, $\alpha_{ss} = 0.36$, $\delta = 0.10$

Implementation hints

The model can be implemented in many different equally good ways. Below are some hints for one such implementation.

Compared to the baseline HANC model, the first new thing is additional heterogeneity in η^0 and η^1 (through χ).

This calls for extending the code with something like:

- In method **setup()**:

```
Nfix = 6
par.phi0_ss = 1.0 # steady state productivity of labor type 0
par.phi1_ss = 2.0 # steady state productivity of labor type 1
```

- In method **allocate()**:

```
par.beta_grid = np.zeros(par.Nfix)
par.eta0_grid = np.zeros(par.Nfix)
par.eta1_grid = np.zeros(par.Nfix)
```

- In function **obj_ss()**:

```
ss.phi0 = par.phi0_ss
ss.phi1 = par.phi1_ss
ss.L0 = ss.phi0*2/3 # from equation (5) combined with P(chi=0) = 2/3
ss.L1 = ss.phi1*1/3 # from equation (5) combined with P(chi=1) = 1/3
```

- In function **prepare_hh_ss()**:

```
par.beta_grid[:] = np.tile(beta_grid,2)
par.eta0_grid[:] = np.hstack((np.ones(par.Nbeta),np.zeros(par.Nbeta)))
par.eta1_grid[:] = np.hstack((np.zeros(par.Nbeta),np.ones(par.Nbeta)))
ss.Dbeg[:3,:,0] = z_ergodic*2/3*1/3 # ergodic at a_lag = 0.0
ss.Dbeg[:3,:,1:] = 0.0 # none with a_lag > 0.0
ss.Dbeg[3:,:,0] = z_ergodic*1/3*1/3 # ergodic at a_lag = 0.0
ss.Dbeg[3:,:,1:] = 0.0 # none with a_lag > 0.0
```

- In function **solve_hh_backwards()**:

```
l0[i_fix,i_z,:] = phi0*par.eta0_grid[i_fix]*par.z_grid[i_z]
l1[i_fix,i_z,:] = phi1*par.eta1_grid[i_fix]*par.z_grid[i_z]
```

Next, the definition of *shocks* and *unknowns* must be changed. In **method settings()**, we should have something like:

```
self.shocks = ['Gamma','phi0','phi1'] # exogenous shocks
self.unknowns = ['K','L0','L1'] # endogenous unknowns
```

It is important to check your results step by step:

1. Check you can solve the household_problem.
   (for your choice of `ss.phi0, ss.phi1, ss.r, ss.w0, ss.w1`)
2. Check all markets clear after completing `find_ss()`.
3. Run `model.test_path()` after `find_ss()` to find errors in `blocks.py`.

Finally, some extra tips:

1. **Permanent shock:** You should use the `ini=` option in `find_transition_path()`
2. **Utility:** You might want to look at household utility. In solve_hh_backwards():

   ```
   u[i_fix,i_z] = c[i_fix,i_z]**(1-par.sigma)/(1-par.sigma) - par.nu
   ```

   In the notebook you can calculate expected discounted utility as:

   ```
   v = np.sum([par.beta_grid[i_fix]**t*
   np.sum(path.u[t,i_fix]*path.D[t,i_fix]/np.sum(path.D[t,i_fix]))
   for t in range(par.T)])
   ```