4. EXERCISE

1.

a) A simple model for the US population is

$$P_t = \frac{310273000}{1 + e^{-0.0353(t-1913.25)}},$$

where t is the date in years. Plot the population for years $t \in [1790, 2000]$. What population is predicted in year 2020?

b) Plot a 3D surface of the function $z = x^2 + 2\sin(y)$ for $x \in [-5, 5]$ and $y \in [-10, 10]$. Use subplot to draw the contours of the function in the same figure.

c) A football/soccer player shoots $n = 7$ times in a match with a probability of goal $p = 0.25$ in each shot (he/she is very good!). Number of goals then follows binomial distribution. Plot the distribution and its normal approximation (mean np and variance $np(1-p)$) in the same figure. Tip: use *binopdf* to get binomial probabilities, *bar* to plot the binomial distribution, to get normal approximation use *pdf* function.

2.

a) Write the following system of equations in standard form

$$\begin{aligned} 25(x_1 - x_2) + 15x_3 + x_4 - 7 &= 0 \\ x_2 + x_3 &= 1 \\ x_1 - 5x_2 - 3x_3 - x_4 + 16 &= 0 \\ x_1/x_4 &= 2. \end{aligned}$$

Use Matlab command rank to determine whether this system can be solved or not? Try finding a solution. Tip: solving a system $Ax = b$ can be done by command A\b.

b) Consider the following system (from Higham's testmatrix toolbox)

$$\begin{bmatrix} 1 \times 10^{-10} & 0.9 & -0.4 \\ 0 & 0.9 & -0.4 \\ 0 & 0 & 1 \times 10^{-10} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Obtain the solution by hand and compare this to the solution that Matlab gives (use *format long*). Why are the two solutions different?

c) Compute the stationary distribution of the stochastic matrix

$$A = \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix}.$$

Tip: use Matlab function *eig*.

3.

a) Write down a Matlab function that takes a vector of payoffs u and a discount factor δ as its inputs, and gives the discounted sum of payoffs $\sum_{k=0}^{n-1} \delta^k u_k$ as its output. If possible do not use any loops.

b) The present value of an annuity (a yearly sum of money) may be computed from the formula

$$PVA = A[(1 - (1/(1 + r)^n))/r],$$

where A is the annuity, r is the nominal yearly interest rate (in decimal form), n is the number of years over which the annuity is paid and PVA is the present value.

- Write down a Matlab function that calculates PVA for given A, r, and n.
- Say that you were offered to choose between 500 000 today or 50 000 per year for 20 years. What would you choose if the interest rate was 5% per year?
- Determine the indifference point for the interest rate which makes the present value of 50 000 equal to 500 000. Use the *fzero* function.

4.
Assume that the price of an asset follows the process $p_{t+\Delta t} = p_t + \Delta P$, where $\Delta P = p_t(\mu\Delta t + \sigma\Delta X_t)$, where Δt is the time step and ΔX_t is a normally distributed (iid) random variable with mean zero and variance Δt.

a) Write a function (or a script) which simulates the prices of the asset for a given number of periods and for given parameters μ, Δt, and σ, and for a given initial value p_0. Test your code for different values of parameters and plot the outcome paths, e.g., use $\mu = 0.04$, $\sigma = 0.3$, $p_0 = 100$, and different time steps Δt. Tip: use *randn* function to generate normally distributed random numbers.

b) Modify your code to include the number of assets in each period when the amount A is invested in the asset in each period starting from the first period.

c) Compare two different investment scenarios for parameters $\Delta t = 1$, $\mu = 0.04$, $\sigma = 0.3$, $p_0 = 100$, $n = 20$ (number of periods): (1) invest $A = 100$ in each period, (2) invest the present value of scenario (1) investment stream in date $t = 0$ with interest rate $r = \mu$. Loop over 10 000 simulations and compare the means and variances of the final wealth (value of asset holdings) in the two scenarios. Tip: use *mean* and *var* functions for mean and variance.

5.

Consider the following New Keynesian model, where the endogenous variables are the nominal interest rate i, the inflation rate π, the natural real interest rate r, and the output gap x. The equations of the model (macroeconomic demand curve=IS curve, Phillips curve, monetary policy rule, real rate, and technology) are

$$\begin{aligned} x_t &= \mathbb{E}[x_{t+1}] - \sigma^{-1}(i_t - \mathbb{E}[\pi_{t+1}] - r_t) \\ \pi_t &= \beta\mathbb{E}[\pi_{t+1}] + \kappa x_t \\ i_t &= \rho + \phi_\pi \pi_t + \phi_x x_t \\ r_t &= \rho - \sigma\psi(1-\rho_a)a_t \\ a_t &= \rho_a a_{t-1} + \varepsilon_t, \end{aligned}$$

where a_t is the technology governed by the exogenous shock ε. Below is a template for a Dynare model file, fill in the missing details and run the model.

```
var x, pi, i, a, r;
varexo ae;
parameters alph, bet, thet, sig,  rho, phi, phipi, phix, rhoa, lam, kappa, psi;
alph=0.5;
bet = 0.99;
thet = 0.75;
sig = 1;
phi = 1;
rho = -log(bet);
phipi=1.5;
phix=0.125;
rhoa=0.9;
lam=1/thet*(1-thet)*(1-bet*thet);
psi=(1+phi)/(sig*(1-alph)+phi+alph);
kappa=lam*(sig+psi);

model; //FILL IN!
end;

initval;
x = 0;
pi= 0;
i = rho;
r = rho;
a=0;
ae=0;
end;

steady;

shocks;
var ae=1;
end;

stoch_simul(periods=10000, order=1);
```

6. (extra)
Find out how ”lookup” function works in Matlab/Octave. Program a function called fastsetdiff that takes as its arguments two (sorted) vectors x and y and returns the elements of x that are not contained in y, i.e., the function computes the set difference of sets of elements in two vectors. Avoid loops. Can you expect your code to perform faster than ”setdiff” function in Matlab/Octave, if so why?