Computational Macroeconomics
Exercises

Week 3

Willi Mutschler
willi@mutschler.eu

Version: June 8, 2022

Contents

1. Symbolic Math Toolbox in MATLAB **1**

2. The Algebra of New Keynesian Models **5**

3. Basic New Keynesian Model: preprocessing and steady-state in Dynare **9**

A. Solutions **11**

1. Symbolic Math Toolbox in MATLAB

MATLAB's Symbolic Math Toolbox provides a set of functions for solving, plotting, and manipulating symbolic math equations. It enables you to perform symbolic computations by defining a special data type — symbolic objects. Functions are called using the familiar MATLAB syntax and are available for integration, differentiation, simplification, equation solving, and other mathematical tasks.

Creating symbolic numbers, matrices, and variables

Symbolic numbers are exact representations (like we do on paper). Calculations on symbolic numbers and variables are exact, unlike floating-point numbers which are only accurate to machine precision. [1]

In MATLAB you can create symbolic objects by using either `syms` or `sym`. The `syms` command is actually shorthand for the `sym` syntax, but the two functions handle assumptions differently. For example `syms x` creates a symbolic variable x in the MATLAB workspace with the value x, whereas `z = sym('y')` creates a symbolic variable z with the value y.

1. Create the symbolic number $sym(1/3)$ and compare it to the floating number 1/3. How does MATLAB display these numbers in the terminal?

2. Demonstrate the exactness of symbolic numbers by computing $\sin(\pi)$ symbolically and numerically.

3. You can create multiple variables in one command with `syms x y z`. Create the variables x, y, and z.

4. If you want to create an array of numbered symbolic variables, the `syms` syntax is inconvenient. Use `sym('a',[1 20])` instead to create a row vector containing the symbolic variables $a_1, ..., a_20$ and assign it to the MATLAB variable A.

5. By combining `sym` and `syms`, you can create many symbolic variables with corresponding variables name in the MATLAB workspace. Clear the workspace. Create the symbolic variables $a_1, ..., a_10$ and assign them the MATLAB variable names $a_1, ..., a_10$, respectively, by running `syms(sym('a', [1 10]))`. Have a look in the workspace which variables are created.

6. Create the following symbolic matrix

$$A = \begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix}$$

   Check if the sum of the elements of the first row equals the sum of the elements of the second column.

7. The `sym` function also lets you define a symbolic matrix or vector without having to define its elements in advance. In this case, the `sym` function generates the elements of a symbolic matrix at the same time that it creates a matrix. The function presents all generated elements using the same form: the base (which must be a valid variable name), a row index, and a column index. Run the following code

```
clearvars
A = sym('a', [2 4])
B = sym('b_%d_%d', [2 4])
```

   and confirm that you understood the syntax by looking at the created variables.

[1] Granted MATLAB also offers variable-precision floating-point arithmetic (VPA).

Perform symbolic computations

With the Symbolic Math Toolbox software, you can also perform symbolic computations like derivatives, integrals or solving equations.

8. Have a look at the output of the following commands:

```
clearvars
syms a b c x;
f = a*x^2 + b*x + c
g(x) = a*x^2 + b*x + c
g(4)
solve(g)
solve(f==2,a)
```

   What does `solve(g)` and `solve(f,a)` do?

9. To differentiate a symbolic expression, use the `diff` or `jacobian` commands. Take the first derivative of f with respect to x and a, respectively:

```
clearvars
syms a b c x;
f = a*x^2 + b*x + c
diff(f,x)
diff(f,a)
diff(f,b)
diff(f,c)
diff(f,x,a)
jacobian(f,[a;b;c;x])
```

10. To compute the integral use the `int` function. Compute $\int_{-\infty}^{\infty} ax^2+bx+cdx$ and $\int_{-2}^{1} ax^2+bx+cdx$.

Symbolic simplifications

The Symbolic Math Toolbox provides a set of simplification functions allowing you to manipulate the output of a symbolic expression, e.g. `simplify`, `expand`, `factor`, `horner` or `subs`. Symbolic simplification is not always straightforward as different problems require different forms of the same mathematical expression.

11. Define the following polynomial of the golden ratio:

```
phi = (1 + sqrt(sym(5)))/2;
golden_ratio = phi^2 - phi - 1
```

   Now run `simplify(golden_ratio)`.

12. To show the order of a polynomial or to symbolically differentiate or integrate a polynomial the standard polynomial form with all the parentheses multiplied out and all the similar terms summed up is useful. You can do this by using `expand`:

```
clearvars
syms x
f = (x^2-1)*(x^4+x^3+x^2+x+1)*(x^4-x^3+x^2-x+1);
disp(f);
expand(f)
```

13. To show the polynomial roots a factor simplification is useful:

```
clearvars
syms x
g = x^3 + 6*x^2 + 11*x + 6;
disp(g)
factor(g)
```

14. The nested (Horner) representation of a polynomial is often most efficient for numerical evaluations:

```
clearvars
syms x
h = x^5 + x^4 + x^3 + x^2 + x;
disp(h)
horner(h)
```

15. You can substitute a symbolic variable with another symbolic variable or a numeric value by using the subs function. When your expression contains more than one variable, you can specify the variable for which you want to make the substitution. For example:

```
clearvars
syms x y
f = x^2*log(y) + 5*x*sqrt(y);
disp(f)
subs(f, x, 3)
subs(f, x, y)
```

You can also substitute a set of elements in a symbolic matrix:

```
syms a b c
alpha = sym('alpha');
beta = sym('beta');
A = [a b c; c a b; b c a];
disp(A);
A(2,1) = beta;
subs(A,b,alpha);
B = subs(A,b,alpha);
disp(A);
disp(B)
```

Writing out symbolic expressions to script files for numerical evaluation

Often we want to do some symbolic manipulations and then evaluate those expressions numerically. Evaluating symbolic variables, however, is computationally expensive in terms of memory allocation and speed of numerical evaluation. One can write out the symbolic expressions to script files and evaluate these, because evaluating functions is computationally very cheap and fast. Consider the following example, where we want to compute the Jacobian of f:

```
clearvars
var_names = ["x";"y"];
param_names = ["ALPHA";"BETA"];
syms(sym(var_names));
syms(sym(param_names));
```

```
f(1,1) = x^ALPHA*log(y) + BETA*x*sqrt(y);
f(2,1) = x^2-y + BETA/ALPHA*sqrt(x^3)*(y/2);
f(3,1) = x^ALPHA + BETA*x;
f(4,1) = log(y) + BETA*sqrt(y);
df = jacobian(f,[x;y]);
```

16. Have a look at the help file and the examples for `fprintf` which is able to print formatted data to a text file.

17. Run the following commands:

```
NameOfFunction = 'num_df';
NameOfFile = strcat(NameOfFunction,'.m');
NameOfOutput = 'df';
if exist(NameOfFile,'file') > 0
    delete(NameOfFile);
end
fileID = fopen(NameOfFile,'w');
fprintf(fileID,'function %s = %s(%s)\n',...
          NameOfOutput,NameOfFunction,strjoin([var_names;param_names],','));
fprintf(fileID,'%s = zeros(%d, %d);\n',NameOfOutput,size(df,1),size(df,2));
[nonzero_row,nonzero_col,nonzero_vals] = find(df);
for j = 1:size(nonzero_vals,1)
    fprintf(fileID,'%s(%d,%d) = %s;\n',...
          NameOfOutput,nonzero_row(j),nonzero_col(j),char(nonzero_vals(j)));
end
fprintf(fileID,'\nend %% function end \n');
fclose(fileID);
```

Note that a new file called `num_df.m` is created in your working directory. Open it and try to understand what the commands actually do.

18. Now let's compare the runtime of computing *df* a 100 times for random data by (i) substituting the symbolic expressions or (ii) evaluating the script files:

```
rng(123); % get same random numbers by setting seed
tic;
for j=1:100
    num_df(randn(),randn(),rand(),rand());
end
toc

rng(123); % get same random numbers by setting seed
tic;
for j=1:100
    double(subs(df,[x y ALPHA BETA],[randn(),randn(),rand(),rand()]));
end
toc
```

Readings

- `https://mathworks.com/help/symbolic/getting-started-with-symbolic-math-toolbox.html`
- `https://mathworks.com/help/symbolic/symbolic-computations-in-matlab.html`

2. The Algebra of New Keynesian Models

Consider the basic New Keynesian (NK) model without capital, a linear production function and Calvo (1983) price frictions.

Model description

Households The economy is assumed to be inhabited by a large number of identical households. The representative household maximizes present as well as expected future utility

$$\max E_t \sum_{j=0}^{\infty} \beta^j U(c_{t+j}, n_{t+j}^s, z_{t+j})$$

with $\beta < 1$ denoting the discount factor and E_t is the expectation operator conditional on information at time t. The contemporaneous utility function

$$U(c_t, n_t^s, z_t) = \left(\frac{c_t^{1-\sigma}}{1-\sigma} - \frac{(n_t^s)^{1+\varphi}}{1+\varphi} \right) z_t$$

has three arguments: a consumption index c_t, a labor index n_t which corresponds to either hours worked or employed household members, and an exogenous preference shifter z_t. Note that the marginal utility of consumption is positive, whereas more labor reduces utility. The inverse of σ is the intertemporal elasticity of substitution, whereas the inverse of φ is the Frisch elasticity of labor. Note that the exogenous preference shifter influences only intertemporal decisions but not intratemporal ones. The consumption index is formed by assuming a continuum of goods represented by the interval $h \in [0, 1]$ and aggregated to a single consumption good using a Dixit and Stiglitz (1977) aggregation technology:

$$c_t = \left(\int_0^1 c_t(h)^{\frac{\epsilon-1}{\epsilon}} dh \right)^{\frac{\epsilon}{\epsilon-1}}$$

That is, $c_t(h)$ denotes the quantity of good h consumed by the household in period t. $\epsilon > 1$ is an elasticity parameter measuring the *love-of-variety*. The household decides how to allocate its consumption expenditures among the different goods by taking the price $P_t(h)$ of good h as given and maximizing the consumption index c_t for any given level of expenditures. Similarly, in each period the household takes the nominal wage W_t as given and supplies perfectly elastic labor service to the firm sector. In return she receives nominal labor income $W_t n_t^s$ and, additionally, nominal profits and dividends $P_t \int_0^1 div_t(f) df$ from each firm $f \in [0, 1]$ in the intermediate goods sector, because the firms are owned by the household. Moreover, the household purchases a quantity of one-period nominally risk-less bonds B_t at price Q_t. The bond matures the following period and pays one unit of money at maturity. Income and wealth are used to finance consumption expenditures. In total this defines the (nominal) budget constraint of the household

$$\int_0^1 P_t(h) c_t(h) dh + Q_t B_t \leq B_{t-1} + W_t n_t^s + P_t \int_0^1 div_t(f) df$$

In addition, it is assumed that the household is subject to a solvency constraint that prevents it from engaging in Ponzi-type schemes:

$$\lim_{T \to \infty} E_t \left\{ \Lambda_{t,T} \frac{B_T}{P_T} \right\} \geq 0$$

for all periods t, where

$$\Lambda_{t,T} = \beta^{T-t} \frac{\partial U(c_T, n_T^s, z_T)/\partial c_T}{\partial U(c_t, n_t^s, z_T)/\partial c_t} \tag{1}$$

denotes the stochastic discount factor.

Furthermore, let $\Pi_t = P_t/P_{t-1}$ denote the gross inflation rate, then the following relationships for the nominal interest rate R_t and the real interest rate r_t hold:

$$Q_t = \frac{1}{R_t} \tag{2}$$

$$R_t = r_t E_t \Pi_{t+1} \tag{3}$$

1. Explain the economics behind equation (2) that determines the nominal interest rate and equation (3) that determines the real interest rate.

2. Is there debt in this model? In other words, what is the optimal path for B_t in this model?

3. Explain the difference between the solvency constraint $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} \geq 0$ and the transversality condition $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} = 0$ which holds in the optimum allocation.

4. Show that cost minimization of consumption expenditures implies

$$c_t(h) = \left(\frac{P_t(h)}{P_t}\right)^{-\epsilon} c_t$$

$$P_t = \left(\int_0^1 P_t(h)^{1-\epsilon} dh\right)^{\frac{1}{1-\epsilon}}$$

   Interpret these equations. What does this imply for the budget constraint?

5. Derive the intratemporal and intertemporal optimality conditions:

$$w_t := \frac{W_t}{P_t} = -\frac{\frac{\partial U(c_t, n_t^s, z_t)}{\partial n_t^s}}{\frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t}} \tag{4}$$

$$\frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t} = \beta E_t \left[\frac{\partial U(c_{t+1}, n_{t+1}^s, z_{t+1})}{\partial c_{t+1}} r_t\right] \tag{5}$$

   where w_t denotes the real wage and $\Pi_{t+1} = P_{t+1}/P_t$ the gross inflation rate. Interpret these equations.

Firms The economy is populated by a continuum of firms indexed by $f \in [0, 1]$ that produce differentiated goods $y_t(f)$. The technology for transforming these intermediate goods into the final output good y_t has the Dixit and Stiglitz (1977) form:

$$y_t = \left[\int_0^1 y_t(f)^{\frac{\epsilon-1}{\epsilon}} df\right]^{\frac{\epsilon}{\epsilon-1}} \tag{6}$$

where $\epsilon > 1$ is the substitution elasticity between inputs, the so-called *love-of-variety*.

Intermediate firm f uses the following linear production function to produce their differentiated good

$$y_t(f) = a_t n_t^d(f) \tag{7}$$

where a_t denotes the common technology level available to all firms. Firms face perfectly competitive factor markets for hiring labor $n_t^d(f)$. Real profit of firm f is equal to revenues from selling its differentiated good at price $P_t(f)$ minus costs from hiring labor at wage w_t

$$div_t(f) = \frac{P_t(f)}{P_t} y_t(f) - w_t n_t^d(f) \tag{8}$$

The objective of the firm is to choose contingent plans for $P_t(f)$ and $n_t^d(f)$ so as to maximize the present discounted value of nominal dividend payments given by

$$E_t \sum_{j=0}^{\infty} \Lambda_{t,t+j} P_{t+j} div_{t+j}(f)$$

where household's stochastic discount factor $\Lambda_{t,t+j}$ takes into account that firms are owned by the household.

Prices of intermediate goods are determined by nominal contracts as in Calvo (1983) and Yun (1996). In each period firm f faces a constant probability $1-\theta, 0 \leq \theta \leq 1$, of being able to re-optimize the price $P_t(f)$ of its good $y_t(f)$. The probability is independent of the time it last reset its price. Formally:

$$P_t(f) = \begin{cases} \widetilde{P}_t(f) & \text{with probability } 1-\theta \\ P_{t-1}(f) & \text{with probability } \theta \end{cases} \tag{9}$$

where $\widetilde{P}_t(f)$ is the re-optimized price in period t. Accordingly, when a firm cannot re-set its price for j periods, its price in period $t+j$ is given by $\widetilde{P}_t(f)$ and stays there until the firm can optimize it again. Hence, the firm's objective in t is to set $\widetilde{P}_t(f)$ to maximize expected profits until it can re-optimize the price again in some future period $t+j$. The probability to be stuck at the same price for j periods is given by θ^j.

6. Show that profit maximization in the final goods sector implies:

$$y_t(f) = \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} y_t$$
$$P_t = \left[\int_0^1 P_t(f)^{1-\epsilon} df\right]^{\frac{1}{1-\epsilon}}$$

Interpret these equations.

7. Derive the following expression for the stochastic discount factor:

$$\Lambda_{t,t+1+j} = \beta \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{-1} \Lambda_{t+1,t+1+j}$$

8. Show that the optimal labor demand schedule of intermediate good firm f is given by:

$$w_t = mc_t(f) a_t = mc_t(f) \frac{y_t(f)}{n_t^d(f)} \tag{10}$$

where $mc_t(f)$ are real marginal costs of firm f. What does this imply for aggregate real marginal costs $mc_t = \int_0^1 mc_t(f) df$?

9. Denote $\widetilde{p}_t := \frac{\widetilde{P}_t(f)}{P_t}$ and show that optimal price setting of intermediate firms must satisfy:

$$\widetilde{p}_t \cdot s_{1,t} = \frac{\epsilon}{\epsilon - 1} \cdot s_{2,t} \tag{11}$$
$$s_{1,t} = y_t \frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t} + \beta\theta E_t \Pi_{t+1}^{\epsilon-1} s_{1,t+1} \tag{12}$$
$$s_{2,t} = mc_t y_t \frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t} + \beta\theta E_t \Pi_{t+1}^{\epsilon} s_{2,t+1} \tag{13}$$

Explain why firms that reset prices set the same price, i.e. $\widetilde{P}_t(f) = \widetilde{P}_t$ or in other words we can drop the f.

10. Show that the law of motion for the optimal re-set price $\widetilde{p}_t = \frac{\widetilde{P}_t(f)}{P_t}$ is given by:

$$1 = \theta \Pi_t^{\epsilon-1} + (1-\theta) \widetilde{p}_t^{1-\epsilon} \tag{14}$$

Monetary Policy The central bank adjusts the nominal interest rate R_t according to an interest rate rule in response to deviations of (i) gross inflation Π_t from a target Π^* and (ii) output y_t from steady-state output y:

$$R_t = R\left(\frac{\Pi_t}{\Pi^*}\right)^{\phi_\pi}\left(\frac{y_t}{y}\right)^{\phi_y} e^{\nu_t} \tag{15}$$

where R denotes the nominal interest rate in steady state, ϕ_π the sensitivity parameter to inflation deviations, ϕ_y the feedback parameter of the output gap and ν_t an exogenous deviation to the rule.

Exogenous variables and stochastic shocks The exogenous preference shifter z_t, the level of technology a_t and the exogenous deviations ν_t from the monetary rule evolve according to

$$\log z_t = \rho_z \log z_{t-1} + \varepsilon_{z,t} \tag{16}$$

$$\log a_t = \rho_a \log a_{t-1} + \varepsilon_{a,t} \tag{17}$$

$$\nu_t = \rho_\nu \nu_{t-1} + \varepsilon_{\nu,t} \tag{18}$$

with persistence parameters ρ_z, ρ_a and ρ_ν. The preference shock $\varepsilon_{z,t}$, the productivity shock $\varepsilon_{a,t}$ and the monetary policy shock $\varepsilon_{\nu,t}$ are iid Gaussian:

$$\begin{pmatrix}\varepsilon_{z,t}\\ \varepsilon_{a,t}\\ \varepsilon_{\nu,t}\end{pmatrix} \sim N\left(\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}\sigma_z^2 & 0 & 0\\ 0 & \sigma_a^2 & 0\\ 0 & 0 & \sigma_\nu^2\end{pmatrix}\right)$$

Market clearing

11. What does market clearing imply for private bonds B_t?
12. Explain why labor market clearing implies:

$$n_t^s = \int_0^1 n_t^d(f)df = n_t$$

 where n_t is the equilibrium amount of labor.
13. Show that aggregate real profits are given by

$$div_t \equiv \int_0^1 div_t(f)df = y_t - w_t n_t \tag{19}$$

14. Show that aggregate demand is given by

$$y_t = c_t \tag{20}$$

15. Denote $p_t^* = \int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} df$. Show that aggregate supply is given by

$$p_t^* y_t = a_t n_t \tag{21}$$

 Explain why p_t^* is called the price inefficiency distortion.
16. Derive the law of motion for the price inefficiency distortion p_t^*:

$$p_t^* = (1-\theta)\,\widetilde{p}_t^{-\epsilon} + \theta \Pi_t^\epsilon p_{t-1}^* \tag{22}$$

Readings

- Galí (2015, Ch.3)
- Heijdra (2017, Ch.19)
- Romer (2019, Ch.7)
- Walsh (2017, Ch.8)
- Woodford (2003, Ch.3)

3. New Keynesian Model: preprocessing and steady-state in Dynare

Consider the just derived basic New Keynesian (NK) model without capital and a linear production function. Assume that the parameters have the following values:

Table 1: Parameter Values

Parameter	**Value**
β	0.990
ρ_a	0.900
ρ_z	0.500
ρ_ν	0.500
σ	1.000
φ	5.000
θ	0.750
ϵ	9.000
ϕ_π	1.500
ϕ_y	0.125
Π^*	1.005

Write a Dynare mod file for this model and compute the steady state of the model either analytically or numerically. Hint: The model equations are (2), (3), (4), (5), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), and (22). Additionally, for reporting purposes, add the following equations:

$$\begin{aligned}
\widehat{y}_t &= \log(y_t) - \log(y) & \widehat{w}_t &= \log(w_t) - \log(w) & \widehat{n}_t &= \log(n_t) - \log(n) \\
\widehat{\Pi}_t^{an} &= 4\left(\log(\Pi_t) - \log(\Pi)\right) & \widehat{R}_t^{an} &= 4\left(\log(R_t) - \log(R)\right) & \widehat{r}_t^{an} &= 4\left(\log(r_t) - \log(r)\right) \\
\widehat{mc}_t &= \log(mc_t) - \log(mc) & \widehat{a}_t &= \log(a_t) - \log(a) & \widehat{z}_t &= \log(z_t) - \log(z)
\end{aligned}$$

where variables without a time subscript denote steady-state values.

References

Calvo, Guillermo A. (Sept. 1983). “Staggered Prices in a Utility-Maximizing Framework”. In: *Journal of Monetary Economics* 12.3, pp. 383–398. ISSN: 03043932. DOI: 10.1016/0304-3932(83)90060-0.

Dixit, Avinash K. and Joseph E. Stiglitz (1977). “Monopolistic Competition and Optimum Product Diversity”. In: *The American Economic Review* 67.3, pp. 297–308. ISSN: 00028282.

Galí, Jordi (2015). *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework and Its Applications*. Second edition. Princeton ; Oxford: Princeton University Press. ISBN: 978-0-691-16478-6.

Heijdra, Ben J. (2017). *Foundations of Modern Macroeconomics*. Third edition. Oxford: Oxford university Press. ISBN: 978-0-19-878413-5.

Kimball, Miles S. (1995). “The Quantitative Analytics of the Basic Neomonetarist Model”. In: *Journal of Money, Credit and Banking* 27.4, pp. 1241–1277. ISSN: 00222879, 15384616. DOI: 10.2307/2078048.

Romer, David (2019). *Advanced Macroeconomics*. Fifth Edition. The McGraw-Hill Series in Economics. Dubuque: McGraw-Hill Education. ISBN: 978-1-260-18521-8.

Smets, Frank and Rafael Wouters (2007). “Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach”. In: *American Economic Review* 97.3, pp. 586–606. DOI: 10.1257/aer.97.3.586.

Walsh, Carl E. (2017). *Monetary Theory and Policy*. Fourth edition. London, England ; Cambridge, Massachusetts: MIT Press. ISBN: 978-0-262-03581-1.

Woodford, Michael (2003). *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton, N.J. ; Woodstock, Oxfordshire [England]: Princeton University Press. ISBN: 978-0-691-01049-6.

Yun, Tack (Apr. 1996). “Nominal Price Rigidity, Money Supply Endogeneity, and Business Cycles”. In: *Journal of Monetary Economics* 37.2, pp. 345–370. ISSN: 03043932. DOI: 10.1016/S0304-3932(96)90040-9.

A. Solutions

1 Solution to Symbolic Math Toolbox in MATLAB

progs/matlab/symbolic_toolbox_illustation.m

```
%% 1
sym(1/3)
1/3
% The symbolic number is represented in exact rational form, while the floating-point
    number is a decimal approximation.
% Note that the symbolic result is not indented, while the standard MATLAB result is
    indented.

%% 2
sin(sym(pi))
sin(pi)
% The symbolic result is exact, while the numeric result is an approximation.


%% 3
syms x
z=sym('y')
syms x y z

%% 4
A = sym('a', [1 20]);
disp(A)
% A is a symbolic variable with size 1x20

%% 5
clearvars
A = sym('a', [1 10])
% The MATLAB workspace contains 1 variable A which is symbolic and has 10 entries
clearvars
syms(sym('a', [1 10]))
% The MATLAB workspace contains 10 MATLAB variables whie are symbolic and each has 1
    entry

%% 6
syms a b c;
A = [ a b c;
      c a b;
      b c a];
isequal(sum(A(1,:)), sum(A(:,2)))

%% 7
clearvars
A = sym('a', [2 4])
B = sym('b_%d_%d', [2 4])

%% 8
clearvars
syms a b c x;
f = a*x^2 + b*x + c
g(x) = a*x^2 + b*x + c
g(4)
solve(g)% solves the equation a*x^2 + b*x + c = 0 for the variable x, i.e. this is the
    famous abc formula for quadratic equations
solve(f==2,a) % solves the equation a*x^2 + b*x + c=2 for the variable a. Note that if
    you don't specify a variable, then MATLAB takes the variable which name is closest
    to x

%% 9
diff(f,x)
```

```
diff(f,a)
diff(f,b)
diff(f,c)
diff(f,x,a)
jacobian(f,[a;b;c;x])


%% 10
subs(int(f,x),x,-2)
int(f,x,-2,1)

%% 11
phi = (1 + sqrt(sym(5)))/2;
golden_ratio = phi^2 - phi - 1
simplify(golden_ratio)

%% 12
clearvars
syms x
f = (x^2-1)*(x^4+x^3+x^2+x+1)*(x^4-x^3+x^2-x+1);
disp(f)
expand(f)

%% 13
syms x
g = x^3 + 6*x^2 + 11*x + 6;
disp(g)
factor(g)

%% 14
clearvars
syms x
h = x^5 + x^4 + x^3 + x^2 + x;
disp(h)
horner(h)

%% 15
clearvars
syms x y
f = x^2*log(y) + 5*x*sqrt(y);
disp(f)
subs(f, x, 3)
subs(f, x, y)

syms a b c
alpha = sym('alpha');
beta = sym('beta');
A = [a b c; c a b; b c a];
disp(A);
A(2,1) = beta;
subs(A,b,alpha);
B = subs(A,b,alpha);
disp(A);
disp(B)
%The subs function does not change the original expression f:

%% 16
doc fprintf

%% 17
clearvars
var_names = ["x";"y"];
param_names = ["ALPHA";"BETA"];
syms(sym(var_names));
syms(sym(param_names));
```

```
f(1,1) = x^ALPHA*log(y) + BETA*x*sqrt(y);
f(2,1) = x^2-y + BETA/ALPHA*sqrt(x^3)*(y/2);
f(3,1) = x^ALPHA + BETA*x;
f(4,1) = log(y) + BETA*sqrt(y);
df = jacobian(f,[x;y]);

NameOfFunction = 'num_df';
NameOfFile = strcat(NameOfFunction,'.m');
NameOfOutput = 'df';
if exist(NameOfFile,'file') > 0
    delete(NameOfFile); % Delete old version of file (if it exists)
end
fileID = fopen(NameOfFile,'w');
fprintf(fileID,'function %s = %s(%s)\n',NameOfOutput,NameOfFunction,strjoin([var_names;
    param_names],','));

fprintf(fileID,'\n%% Evaluate the jacobian of f=x^ALPHA*log(y) + BETA*x*sqrt(y)\n');
fprintf(fileID,'\n%% Initialize %s\n',NameOfOutput);
fprintf(fileID,'%s = zeros(%d, %d);\n',NameOfOutput,size(df,1),size(df,2));
[nonzero_row,nonzero_col,nonzero_vals] = find(df);
for j = 1:size(nonzero_vals,1)
    fprintf(fileID,'%s(%d,%d) = %s;\n',NameOfOutput,nonzero_row(j),nonzero_col(j),char(
    nonzero_vals(j)));
end
fprintf(fileID,'\nend %% function end \n');
fclose(fileID);

%% 18
% Compare time to evaluate either script files or symbolic expressions
rng(123); % get same random numbers by setting seed
tic;
for j=1:100
    num_df(randn(),randn(),rand(),rand());
end
toc

rng(123); % get same random numbers by setting seed
tic;
for j=1:100
    double(subs(df,[x y ALPHA BETA],[randn(),randn(),rand(),rand()]));
end
toc
```

2 Solution to The Algebra of New Keynesian Models

1. Equation (2) captures that bond prices are inversely related to interest rates. When the interest rate goes up, the price of bonds falls. Intuitively, this makes sense because if you are paying less for a fixed nominal return (at par), your expected return should be higher. More specifically to our model, we consider so-called zero-coupon bonds or discount bonds. These bonds don't pay any interest but derive their value from the difference between the purchase price and the par value (or the face value) paid at maturity. On maturity the bondholder receives the face value of his investment. So instead of interest payments, you get a large discount on the face value of the bond, that is the price is lower than the face value. In other words, investors profit from the difference between the buying price and the face value, contrary to the usual interest income. In our model, we consider zero-coupon bonds with a face value of 1. So suppose that you buy such a bond at a price of 0.8, then although the bond pays no interest, your compensation is the difference between the initial price and the face value. Let R_t denote the *gross yield to maturity* of a zero-coupon bond, that is the discount rate that sets the present value of the promised bond payments equal to the current market price of the bond. So the price of a Zero-Coupon bond is equal to

$$Q_t = \frac{1}{R_t}$$

In our example this would imply $R_t = 1.25$. As there are no other investment opportunities in this model R_t is also equal to the nominal interest rate in the economy.

Equation (3) is the so-called Fisherian equation which states that the gross real return on a bond r_t is equivalent to the gross nominal interest rate divided by the expected gross inflation rate. Inflation expectations are responsible for the difference between nominal and real interest rates, showing that future expectations matter for the economy.

2. In equilibrium, bond-holding is always zero in all periods: $B_t = 0$. This is due to the fact that in this model we have a representative agent and only private bonds. If all agents were borrowing, there would be nobody they could be borrowing from. If all were lenders, nobody would like to borrow from them. So bonds across all agents need to be in 0 net supply as markets need to clear in equilibrium. Note, though, that this bond market clearing condition is imposed *after* you derive the households optimality conditions as household savings behavior in equilibrium still needs to be consistent with the bond market clearing.

3. The *No-Ponzi-Game* or *solvency* condition is an external constraint imposed on the individual by the market or other participants. You forbid your agent from acquiring infinite debt that is never re-paid, a so-called Ponzi-scheme. That is, the individual is restricted from financing consumption by raising debt and then raising debt again to re-pay the previous debt and finance again consumption and so on. The individual would very much like to violate it, though, so we need to impose this constraint. In short: the *solvency* condition prevents that households consume more than they earn and refinance their additional consumption with excessive borrowing. The *transversality condition* is an optimality condition that states that it is not optimal to start accumulating assets and never consume them, i.e. $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} \leq 0$. But with respect to optimality you would still want to run a Ponzi-scheme if allowed one. $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} \leq 0$ combined with $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} \geq 0$ yields $\lim_{T\to\infty} E_t \left\{\Lambda_{t,T}\frac{B_T}{P_T}\right\} = 0$. This condition must be satisfied in order for the individual to maximize intertemporal utility implying that at the limit wealth should be zero. In other words, if at the limit wealth is positive it means that the household could have increased its consumption without necessarily needing to work more hours, thus implying that consumption was not maximized and therefore contradicting the fact that the household behaves optimally. In short: transversality conditions make sure that households do no have any leftover savings (in terms of bonds or capital) as this does not correspond to an optimal path of utility-enhancing consumption.

   We never need to actually include this condition into our codes, but implicitly we use it to pick a certain steady state or trajectory. For instance in the RBC model we have three possible steady states $(k_t = 0, c_t = 0)$, $k_t > 0, c_t = 0$ or $k_t > 0, c_t > 0$. We do not consider the first one because in this case the economy does not exist. All the trajectories leading to the second one violate the transversality condition, so finally we select the third steady state as the *good one* and this is exactly the one that is most interesting from an economic point of view.

   Coming back to our model, both the *solvency* and *transversality condition* are actually full-filled already as bond-holding is always zero in all periods including the hypothetical asymptotic end of life: $B_t = 0$ for all t. So these conditions are rather trivial in this model setting, but are important in more sophisticated models.

4. The household minimizes her consumption expenditures $\int_0^1 P_t(h)c_t(h)dh$ by choosing $c_t(h)$ and taking the aggregation technology into account. That is, the Lagrangian is given by:

$$£^c = \int_0^1 P_t(h)c_t(h)dh + P_t\left(c_t - \left[\int_0^1 c_t(h)^{\frac{\epsilon-1}{\epsilon}}dh\right]^{\frac{\epsilon}{\epsilon-1}}\right)$$

   where P_t denotes the Lagrange multiplier, i.e. the cost of an additional unit in the index c_t.

Setting the derivative with respect to $c_t(h)$ equal to zero yields:

$$\frac{\partial \pounds^c}{\partial c_t(h)} = P_t(h) - P_t \left(\frac{\epsilon}{\epsilon-1}\right) \underbrace{\left[\int_0^1 c_t(h)^{\frac{\epsilon-1}{\epsilon}} dh\right]^{\frac{\epsilon}{\epsilon-1}-1}}_{c_t^{1/\epsilon}} \left(\frac{\epsilon-1}{\epsilon}\right) c_t(h)^{\frac{\epsilon-1}{\epsilon}-1} = 0$$

which can be simplified to:

$$c_t(h) = \left(\frac{P_t(h)}{P_t}\right)^{-\epsilon} c_t$$

Note that this is the demand function for each consumption good $c_t(h)$. Accordingly, ϵ is the (constant) demand elasticity. Other aggregation technologies can change this, see e.g. the Kimball (1995) aggregation technology in Smets and Wouters (2007)-type models.

Plugging this expression into the aggregation technology yields:

$$\begin{aligned}
c_t^{\frac{\epsilon-1}{\epsilon}} &= \int_0^1 c_t(h)^{\frac{\epsilon-1}{\epsilon}} dh = \int_0^1 \left(\left(\frac{P_t(h)}{P_t}\right)^{-\epsilon} c_t\right)^{\frac{\epsilon-1}{\epsilon}} dh = c_t^{\frac{\epsilon-1}{\epsilon}} P_t^{\epsilon-1} \int_0^1 P_t(h)^{1-\epsilon} dh \\
\Leftrightarrow P_t &= \left[\int_0^1 P_t(h)^{1-\epsilon} dh\right]^{\frac{1}{1-\epsilon}} \\
\Leftrightarrow 1 &= \int_0^1 \left(\frac{P_t(h)}{P_t}\right)^{1-\epsilon} dh
\end{aligned}$$

Similar to the aggregation technology for the consumption index c_t, P_t can be interpreted as the aggregation technology for the different prices $P_t(h)$.

In the budget constraint, we can now get rid of one integral $\int_0^1 c_t(h)P_t(h)dh = P_t c_t$, because:

$$\int_0^1 c_t(h)P_t(h)dh = \int_0^1 \left(\frac{P_t(h)}{P_t}\right)^{-\epsilon} c_t P_t(h) dh = P_t c_t \underbrace{\int_0^1 \left(\frac{P_t(h)}{P_t}\right)^{1-\epsilon} dh}_{=1} = P_t c_t$$

That is, conditional on optimal behavior of households, total consumption expenditures can be rewritten as the product of the aggregate price index times the aggregate consumption quantity index.

5. Due to our assumptions, the solvency and transversality conditions as well as the concave optimization problem, we can rule out corner solutions and neglect the non-negativity constraints in the variables and the budget constraint; hence, we only need to focus on the first-order conditions. The Lagrangian for the household's problem is:

$$\begin{aligned}
\pounds^{HH} = E_t \sum_{j=0}^{\infty} \beta^j \left\{U\left(c_{t+j}, n_{t+j}^s, z_{t+j}\right)\right\} \\
+ \beta^j \lambda_{t+j} \left\{ \int_0^1 div_{t+j}(f)df + w_{t+j} n_{t+j}^s + \underbrace{\frac{B_{t-1+j}}{P_{t-1+j}}}_{b_{t-1+j}} \underbrace{\frac{P_{t-1+j}}{P_{t+j}}}_{\Pi_{t+j}^{-1}} - Q_{t+j} \underbrace{\frac{B_{t+j}}{P_{t+j}}}_{b_{t+j}} - c_{t+j} \right\}
\end{aligned}$$

where $\beta^j \lambda_{t+j}$ are the Lagrange multipliers corresponding to period $t+j$'s **real** budget constraint (be aware of the difference between nominal and real variables and constraints; for instance, $b_t = B_t/P_t$ is real debt). The problem is not to choose $\{c_t, n_t^s, b_t\}_{t=0}^{\infty}$ all at once in an open-loop policy, but to choose these variables sequentially given the information at time t in a closed-loop policy, i.e. at period t decision rules for $\{c_t, n_t^s, b_t\}$ given the information set at period t; at period $t+1$ decision rules for $\{c_{t+1}, n_{t+1}^s, b_{t+1}\}$ given the information set at period $t+1$, etc.

First-order condition with respect to c_t

$$\lambda_t = \frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t} = z_t c_t^{-\sigma} \tag{23}$$

This is the marginal utility function, i.e. the shadow price of an additional unit of revenue (e.g. dividends or labor income) in the budget constraint.

First-order condition with respect to n_t^s

$$w_t = -\frac{\partial U(c_t, n_t^s, z_t)/\partial n_t^s}{\lambda_t} = -\frac{\partial U(c_t, n_t^s, z_t)/\partial n_t^s}{\partial U(c_t, n_t^s, z_t)/\partial c_t} = n_t^{\varphi} c_t^{\sigma} \tag{24}$$

This is the intratemporal optimality condition, i.e. labor supply of the household. Note that the preference shifter z_t has no effect on the intratemporal decision.

First-order condition with respect to b_t

$$\lambda_t Q_t = \beta E_t \left[\lambda_{t+1} \Pi_{t+1}^{-1}\right] \tag{25}$$

Combined with (2) and (3) this yields the so-called Euler equation, i.e. the intratemporal choice between consumption and saving:

$$\frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t} = \beta E_t \left[\frac{\partial U(c_{t+1}, n_{t+1}^s, z_{t+1})}{\partial c_{t+1}} R_t \Pi_{t+1}^{-1}\right] \tag{26}$$

$$z_t c_t^{-\sigma} = \beta E_t \left[z_{t+1} c_{t+1}^{-\sigma}\right] r_t \tag{27}$$

In words, intertemporal optimality is characterized by an indifference condition: An additional unit of consumption yields either marginal utility today in the amount of $\frac{\partial U(c_t, n_t^s, z_t)}{\partial c_t}$ (left-hand side). Or, alternatively, this unit of consumption can be saved given the real interest rate r_t. This saved consumption unit has a present marginal utility value of $\beta E_t \left[z_{t+1} c_{t+1}^{-\sigma}\right] r_t$ (right-hand side). An optimal allocation equates these two choices.

6. The output packers maximize profits $P_t y_t - \int_0^1 P_t(f) y_t(f) df$ subject to (6). The Lagrangian is

$$\pounds^p = P_t y_t - \int_0^1 P_t(f) y_t(f) df + \Lambda_t^p \left\{ \left[\int_0^1 y_t(f)^{\frac{\epsilon-1}{\epsilon}} df \right]^{\frac{\epsilon}{\epsilon-1}} - y_t \right\}$$

where Λ_t^p is the Lagrange multiplier corresponding to the aggregation technology. The first-order condition w.r.t y_t is

$$P_t = \Lambda_t^p \tag{28}$$

Λ_t^p is the gain of an additional output unit, hence, equal to the aggregate price index P_t.

The first-order condition w.r.t. $y_t(f)$ yields:

$$\frac{\partial \pounds^p}{\partial y_t(f)} = -P_t(f) + \Lambda_t^p \frac{\epsilon}{\epsilon-1} \left[\int_0^1 y_t(f)^{\frac{\epsilon-1}{\epsilon}} df \right]^{\frac{\epsilon}{\epsilon-1}-1} \frac{\epsilon-1}{\epsilon} y_t(f)^{\frac{\epsilon-1}{\epsilon}-1} = 0 \tag{29}$$

Note that $\left[\int_0^1 y_t(f)^{\frac{\epsilon-1}{\epsilon}} df\right] = y_t^{\frac{\epsilon-1}{\epsilon}}$ and $\Lambda_t^p = P_t$. Therefore:

$$P_t(f) = P_t \left[y_t^{\frac{\epsilon-1}{\epsilon}} \right]^{\frac{\epsilon-\epsilon+1}{\epsilon-1}} y_t(f)^{\frac{\epsilon-1-\epsilon}{\epsilon}} = P_t \left(\frac{y_t(f)}{y_t} \right)^{\frac{-1}{\epsilon}} \tag{30}$$

Reordering yields

$$y_t(f) = \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} y_t \tag{31}$$

This is the demand curve for intermediate good $y_t(f)$. Again we see that ϵ is the constant demand elasticity.

The aggregate price index is implicitly determined by inserting the demand curve (31) into the aggregator (6)

$$y_t = \left[\int_0^1 \left(\left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} y_t\right)^{\frac{\epsilon-1}{\epsilon}} df\right]^{\frac{\epsilon}{\epsilon-1}} \tag{32}$$

$$\Leftrightarrow P_t = \left[\int_0^1 P_t(f)^{1-\epsilon} df\right]^{\frac{1}{1-\epsilon}} \tag{33}$$

7. As the firms are owned by the households, the nominal stochastic discount factor, $\Lambda_{t,t+j}$, between t and $t+j$ is derived from the Euler equation (27) of the households $\lambda_t = \beta E_t \left[\lambda_{t+1} R_t \Pi_{t+1}^{-1}\right]$ which implies for the stochastic discount factor:

$$E_t \Lambda_{t,t+j} = E_t 1/R_{t+j} = E_t \beta^j \frac{\lambda_{t+j}}{\lambda_t} \frac{P_t}{P_{t+j}}$$

From here, we can establish the following relationships:

$$\Lambda_{t,t} = 1 \tag{34}$$

$$\Lambda_{t+1,t+1+j} = \beta^j \frac{\lambda_{t+1+j}}{\lambda_{t+1}} \frac{P_{t+1}}{P_{t+1+j}} \tag{35}$$

$$\Lambda_{t,t+1+j} = \beta^{j+1} \frac{\lambda_{t+1+j}}{\lambda_t} \frac{P_t}{P_{t+1+j}} = \beta \frac{\lambda_{t+1}}{\lambda_t} \frac{P_t}{P_{t+1}} \beta^j \frac{\lambda_{t+1+j}}{\lambda_{t+1}} \frac{P_{t+1}}{P_{t+1+j}} = \beta \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{-1} \Lambda_{t+1,t+1+j} \tag{36}$$

We will need the last relationship later to derive the recursive nonlinear price setting equations.

8. The Lagrangian of the intermediate firm is

$$\pounds^f = E_t \sum_{j=0}^{\infty} \Lambda_{t,t+j} P_{t+j} \left[\frac{P_{t+j}(f)}{P_{t+j}} y_{t+j}(f) - w_{t+j} n_{t+j}^d(f) + mc_{t+j}(f)\left(a_{t+j} n_{t+j}^d(f) - y_{t+j}(f)\right)\right]$$

$mc_t(f)$ denotes the Lagrange multiplier which is the shadow price of producing an additional output unit in the optimum; obviously, this is our notion of real marginal costs. Taking the derivative wrt $n_t^d(f)$ actually boils down to a static problem (as we only need to evaluate for $j=0$) and yields:

$$w_t = mc_t(f) a_t = mc_t(f) \frac{y_t(f)}{n_t^d(f)} \tag{37}$$

where we substituted the production function (7) for a_t. This is the labor demand function, which implies that the labor-to-output ratio is the same across firms and equal to a_t. Note that all firms face the same factor prices and all have access to the same production technology a_t; hence, from the above equation it is evident that marginal costs are identical across firms

$$mc_t(f) = \frac{w_t}{a_t} \tag{38}$$

This means that aggregate marginal costs are also equal to the ratio between the real wage and technology:

$$mc_t = \int_0^1 mc_t(f) df = \frac{w_t}{a_t}$$

9. The Lagrangian of the intermediate firm is

$$\pounds^f = E_t \sum_{j=0}^{\infty} \Lambda_{t,t+j} P_{t+j} \left[\left(\frac{P_{t+j}(f)}{P_{t+j}} \right)^{1-\epsilon} y_{t+j} - w_{t+j} n^d_{t+j}(f) + mc_{t+j}(f) \left(a_{t+j} n^d_{t+j}(f) - \left(\frac{P_{t+j}(f)}{P_{t+j}} \right)^{-\epsilon} y_{t+j} \right) \right] \tag{39}$$

where (compared to above) we used the demand curve (31) to substitute for $y_t(f)$. When firms decide how to set their price they need to take into account that due to the Calvo mechanism they might get stuck at $\widetilde{P}_t(f)$ for a number of periods $j = 1, 2, ...$ before they can re-optimize again. The probability of such a situation is θ^j. Therefore, when firms are able to change prices in period t, they take this into account and the above Lagrangian of the expected discounted sum of nominal profits becomes:

$$\pounds^{f^c} = E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P_{t+j} \left[\left(\frac{\widetilde{P}_t(f)}{P_{t+j}} \right)^{1-\epsilon} y_{t+j} - w_{t+j} n^d_{t+j}(f) + mc_{t+j} \left(a_{t+j} n^d_{t+j}(f) - \left(\frac{\widetilde{P}_t(f)}{P_{t+j}} \right)^{-\epsilon} y_{t+j} \right) \right] \tag{40}$$

$$= E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P^{\epsilon}_{t+j} y_{t+j} \left[\widetilde{P}_t(f)^{1-\epsilon} - P_{t+j} \cdot mc_{t+j} \cdot \widetilde{P}_t(f)^{-\epsilon} \right] + ... \tag{41}$$

where in the second line we focus only on relevant parts for the optimization wrt to $\widetilde{P}_t(f)$. Moreover, we took into account that $mc_t(f) = mc_t$.

The first-order condition of maximizing $\pounds^{f^c}$ wrt to $\widetilde{P}_t(f)$ is

$$0 = E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P^{\epsilon}_{t+j} y_{t+j} \left[(1-\epsilon) \cdot \widetilde{P}_t(f)^{-\epsilon} + \epsilon \cdot P_{t+j} \cdot mc_{t+j} \widetilde{P}_t(f)^{-\epsilon-1} \right] \tag{42}$$

As $\widetilde{P}_t(f) > 0$ does not depend on j, we multiply by $\widetilde{P}_t(f)^{\epsilon+1}$:

$$0 = E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P^{\epsilon}_{t+j} y_{t+j} \left[(1-\epsilon) \cdot \widetilde{P}_t(f) + \epsilon \cdot P_{t+j} \cdot mc_{t+j} \right] \tag{43}$$

Rearranging

$$\widetilde{P}_t(f) \cdot E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P^{\epsilon}_{t+j} y_{t+j} = \frac{\epsilon}{\epsilon - 1} \cdot E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} P^{\epsilon+1}_{t+j} y_{t+j} mc_{t+j} \tag{44}$$

Dividing both sides by $P_t^{\epsilon+1}$

$$\underbrace{\frac{\widetilde{P}_t(f)}{P_t}}_{\widetilde{p}_t} \cdot \underbrace{E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t} \right)^{\epsilon} y_{t+j}}_{S_{1,t}} = \frac{\epsilon}{\epsilon - 1} \cdot \underbrace{E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t} \right)^{\epsilon+1} y_{t+j} mc_{t+j}}_{S_{2,t}} \tag{45}$$

Note that all firms that reset prices face the same problem and therefore set the same price, $\widetilde{P}_t(f) = \widetilde{P}_t$. This is also evident by looking at the infinite sums, $S_{1,t}$ and $S_{2,t}$, because these do not depend on f. Therefore, we can drop the f in $\widetilde{P}_t(f)$ and define $\widetilde{p}_t := \frac{\widetilde{P}_t}{P_t}$. The first-order condition can thus be written compactly:

$$\widetilde{p}_t \cdot S_{1,t} = \frac{\epsilon}{\epsilon - 1} \cdot S_{2,t} \tag{46}$$

Moreover, the two infinite sums can be written recursively. For this we make use of the relationships for the stochstic discount factor (34) and (36). The first recursive sum can be written as:

$$
\begin{aligned}
S_{1,t} &= E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t}\right)^{\epsilon} y_{t+j} \\
&= y_t + E_t \sum_{j=1}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t}\right)^{\epsilon} y_{t+j} \\
&= y_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \Lambda_{t,t+j+1} \left(\frac{P_{t+j+1}}{P_t}\right)^{\epsilon} y_{t+j+1} \\
&= y_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \Lambda_{t,t+j+1} \left(\frac{P_{t+j+1}}{P_{t+1}}\frac{P_{t+1}}{P_t}\right)^{\epsilon} y_{t+j+1} \\
&= y_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \beta \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{-1} \Lambda_{t+1,t+1+j} \left(\frac{P_{t+j+1}}{P_{t+1}}\Pi_{t+1}\right)^{\epsilon} y_{t+j+1} \\
&= y_t + \theta\beta E_t \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{\epsilon-1} \underbrace{E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t+1,t+1+j} \left(\frac{P_{t+j+1}}{P_{t+1}}\right)^{\epsilon} y_{t+j+1}}_{=S_{1,t+1}}
\end{aligned}
$$

The second recursive sum can be written as

$$
\begin{aligned}
S_{2,t} &= E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t}\right)^{\epsilon+1} y_{t+j} mc_{t+j} \\
&= y_t mc_t + E_t \sum_{j=1}^{\infty} \theta^j \Lambda_{t,t+j} \left(\frac{P_{t+j}}{P_t}\right)^{\epsilon+1} y_{t+j} mc_{t+j} \\
&= y_t mc_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \Lambda_{t,t+j+1} \left(\frac{P_{t+j+1}}{P_t}\right)^{\epsilon+1} y_{t+j+1} mc_{t+j+1} \\
&= y_t mc_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \Lambda_{t,t+j+1} \left(\frac{P_{t+j+1}}{P_{t+1}}\frac{P_{t+1}}{P_t}\right)^{\epsilon+1} y_{t+j+1} mc_{t+j+1} \\
&= y_t mc_t + E_t \sum_{j=0}^{\infty} \theta^{j+1} \beta \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{-1} \Lambda_{t+1,t+1+j} \left(\frac{P_{t+j+1}}{P_{t+1}}\Pi_{t+1}\right)^{\epsilon+1} y_{t+j+1} mc_{t+j+1} \\
&= y_t mc_t + \theta\beta E_t \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{\epsilon} \underbrace{E_t \sum_{j=0}^{\infty} \theta^j \Lambda_{t+1,t+1+j} \left(\frac{P_{t+j+1}}{P_{t+1}}\right)^{\epsilon+1} y_{t+j+1} mc_{t+j+1}}_{=S_{2,t+1}}
\end{aligned}
$$

To sum up:

$$
S_{1,t} = y_t + \theta\beta E_t \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{\epsilon-1} S_{1,t+1} \tag{47}
$$

$$
S_{2,t} = y_t mc_t + \theta\beta E_t \frac{\lambda_{t+1}}{\lambda_t} \Pi_{t+1}^{\epsilon} S_{2,t+1} \tag{48}
$$

10. The law of motion for $\widetilde{p}_t = \frac{\widetilde{P}_t}{P_t}$ is given by the aggregate price index (33) which can be re-arranged to

$$
1 = \int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{1-\epsilon} df \tag{49}
$$

Due to the Calvo mechanism, we note that $(1-\theta)$ firms are able re-set their price to $\widetilde{P}_t$, whereas the remaining θ firms cannot do so. Therefore:

$$1 = \int_{optimizers} \left(\frac{P_t(f)}{P_t}\right)^{1-\epsilon} df + \int_{non-optimizers} \left(\frac{P_t(f)}{P_t}\right)^{1-\epsilon} df \tag{50}$$

$$1 = (1-\theta)\left(\frac{\widetilde{P}_t}{P_t}\right)^{1-\epsilon} + \theta \int_0^1 \left(\frac{P_{t-1}(f)}{P_t}\frac{P_{t-1}}{P_{t-1}}\right)^{1-\epsilon} df \tag{51}$$

$$1 = (1-\theta)\widetilde{p}_t^{1-\epsilon} + \theta\left(\frac{P_{t-1}}{P_t}\right)^{1-\epsilon} \int_0^1 \left(\frac{P_{t-1}(f)}{P_{t-1}}\right)^{1-\epsilon} df \tag{52}$$

$$1 = (1-\theta)\widetilde{p}_t^{1-\epsilon} + \theta\Pi_t^{1-\epsilon} \underbrace{\int_0^1 \left(\frac{P_{t-1}(f)}{P_{t-1}}\right)^{1-\epsilon} df}_{\overset{(49)}{=}1} \tag{53}$$

$$1 = (1-\theta)\widetilde{p}_t^{1-\epsilon} + \theta\Pi_t^{\epsilon-1} \tag{54}$$

11. Private bonds B_t are in zero net supply on the budget constraint. Note that this condition can only be imposed after taking first order conditions. It would be invalid to eliminate bonds already in the budget constraint of the household. Even if bonds are in zero net supply, households savings behavior in equilibrium still needs to be consistent with the bond market clearing.

12. In an equilibrium, labor demand from the intermediate firms needs to be equal to the labor supply of the households; hence:

$$n_t^s = \int_0^1 n_t^d(f)df = n_t \tag{55}$$

We label the equilibrium hours worked with n_t.

13. Given the demand for good $y_t(f)$ and the Dixit-Stiglitz aggregation technology, we get:

$$\int_0^1 y_t(f)P_t(f)df = \int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} y_t P_t(f)df = P_t y_t \underbrace{\int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{1-\epsilon} df}_{=1} = P_t y_t$$

Moreover, from the labor market we have $n_t = \int_0^1 n_t(f)df$. Plugging both expressions into aggregate real profits:

$$div_t = \int_0^1 div_t(f)df = \int_0^1 \frac{P_t(f)}{P_t} y_t(f)df - \int_0^1 w_t n_t^d(f)df = y_t - w_t n_t$$

14. Revisit the budget constraint in real terms:

$$\int_0^1 \frac{P_t(h)}{P_t} c_t(h)dh + Q_t b_t \leq b_{t-1}\Pi_t^{-1} + w_t n_t^s + \int_0^1 div_t(f)df$$

which becomes

$$c_t = w_t n_t + (y_t - w_t n_t) = y_t$$

in an optimal allocation with cleared markets. This is the aggregate demand equation.

15. Define $y_t^{sum} = \int_0^1 y_t(f)df$. Using the production function (7) and labor market clearing we get:

$$y_t^{sum} = \int_0^1 a_t n_t^d(f)df = a_t n_t \tag{56}$$

Furthermore, due to the demand for intermediate good $y_t(f)$ in (31) we get:

$$y_t^{sum} = y_t \underbrace{\int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} df}_{=p_t^*} \tag{57}$$

Equating both yields:

$$p_t^* y_t = a_t n_t \tag{58}$$

This is the aggregate supply equation. Price frictions, however, imply that resources will not be efficiently allocated as prices are too high because not all firms can re-optimize their price in every period. This inefficiency is measured by $p_t^* \leq 1$.

16. The law of motion for the inefficiency distortion p_t^* is given due to the Calvo price mechanism, i.e.:

$$\begin{aligned}
p_t^* &= \int_0^1 \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} df \\
p_t^* &= \int_{optimizers} \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} df + \int_{non-optimizers} \left(\frac{P_t(f)}{P_t}\right)^{-\epsilon} df \\
p_t^* &= (1-\theta)\widetilde{p}_t^{-\epsilon} + \theta \int_0^1 \left(\frac{P_{t-1}(f)}{P_t}\right)^{-\epsilon} df \\
p_t^* &= (1-\theta)\widetilde{p}_t^{-\epsilon} + \theta \int_0^1 \left(\frac{P_{t-1}(f)}{P_t}\frac{P_{t-1}}{P_{t-1}}\right)^{-\epsilon} df \\
p_t^* &= (1-\theta)\widetilde{p}_t^{-\epsilon} + \theta \left(\frac{P_{t-1}}{P_t}\right)^{-\epsilon} \int_0^1 \left(\frac{P_{t-1}(f)}{P_{t-1}}\right)^{-\epsilon} df \\
p_t^* &= (1-\theta)\widetilde{p}_t^{-\epsilon} + \theta \Pi_t^{\epsilon} \underbrace{\int_0^1 \left(\frac{P_{t-1}(f)}{P_{t-1}}\right)^{-\epsilon} df}_{=p_{t-1}^*} \\
p_t^* &= (1-\theta)\widetilde{p}_t^{-\epsilon} + \theta \Pi_t^{\epsilon} p_{t-1}^*
\end{aligned}$$

3 Solution to Basic New Keynesian Model: preprocessing and steady-state in Dynare

Steady-State:

The steady-state can be computed analytically for all model variables.

In the non-stochastic steady-state $\varepsilon_{z,t} = \varepsilon_{a,t} = \varepsilon_{\nu,t} = 0$. Accordingly, equations (16), (17) and (18) become:

$$z = 1, \qquad a = 1, \qquad \nu = 0$$

From equation (15) we can infer

$$\Pi = \Pi^*$$

Note that the following derivations would simplify immensely if one considers price stability in steady-state, i.e. $\Pi^* = 1$. However, in the following we'll allow for trend inflation, e.g. an annualized inflation target of 2% which for quarterly models would equal to $\Pi^* = 1.005$.

Anyways, from (14) we get:

$$\widetilde{p} = \left(\frac{1 - \theta\,\Pi^{\epsilon-1}}{1-\theta}\right)^{\frac{1}{1-\epsilon}}$$

Obviously, for $\Pi^* = 1$ we'd get $\widetilde{p} = 1$.

Now comes the messy part. Evaluating (12) and (13) in steady-state, we can combine them to get an expression for s_1/s_2. Evaluating (11) in steady-state in terms of s_1/s_2, we can rearrange to get marginal costs in steady-state:

$$mc = \frac{\frac{\epsilon-1}{\epsilon}\,\widetilde{p}\,(1 - \theta\,\beta\,\Pi^{\epsilon})}{1 - \theta\,\beta\,\Pi^{\epsilon-1}}$$

Again, for $\Pi^* = 1$, this looks much more familiar as $mc = \frac{\epsilon-1}{\epsilon}$.

Now we are able to evaluate (22) in steady state, which becomes:

$$p^* = \widetilde{p}^{-\epsilon}\,\frac{1-\theta}{1-\theta\,\Pi^{\epsilon}}$$

Obviously, for $\Pi^* = 1$ this would imply $p^* = 1$.

Now comes the easy part: evaluating equations (5), (2) and (3) in steady state yields:

$$Q = \frac{\beta}{\Pi}, \qquad R = \frac{1}{Q}, \qquad r = \frac{R}{\Pi}$$

Steady-state wages can be computed from (10) (as $mc_t(f) = mc_t$):

$$w = mc \cdot a$$

To get labor in steady-state, we start with evaluating labor supply (4) in steady-state:

$$w = -\frac{-zn^{\varphi}}{zc^{-\sigma}} = n^{\varphi}c^{\sigma} \overset{(20)}{=} n^{\varphi}y^{\sigma} \overset{(21)}{=} n^{\varphi}\left(\frac{an}{p^*}\right)^{\sigma} = n^{\varphi+\sigma}\left(\frac{a}{p^*}\right)^{\sigma} \tag{59}$$

$$\Leftrightarrow n = \left(\frac{w}{\left(\frac{a}{p^*}\right)^{\sigma}}\right)^{\frac{1}{\varphi+\sigma}} \tag{60}$$

The remaining variables follow accordingly:

$$y = an/p^*, \qquad c = y, \qquad div = y - n\,w, \qquad s_1 = \frac{y\,c^{-\sigma}}{1 - \Pi^{\epsilon-1}\,\theta\,\beta}, \qquad s_2 = \frac{mc \cdot y\,c^{-\sigma}}{1 - \theta\,\beta\,\Pi^{\epsilon}}$$

Lastly, all hat variables have a steady-state of 0 by definition.

`new_keynesian_common.inc:`

```
%-------------------------------------------------------------------------
% Copyright Notice
%-------------------------------------------------------------------------
% This file implements the baseline New Keynesian model of
% Jordi Gali (2015): Monetary Policy, Inflation, and the Business Cycle,
% Princeton University Press, Second Edition, Chapter 3
%
% THIS MOD-FILE REQUIRES DYNARE 4.5 OR HIGHER
%
% This implementation was written originally by Johannes Pfeifer and
% adapted by Willi Mutschler (willi@mutschler.eu).
%
% Copyright (C) 2016-2021 Johannes Pfeifer
% Copyright (C) 2021-2022 Willi Mutschler
%
% This is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% It is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% For a copy of the GNU General Public License,
% see <http://www.gnu.org/licenses/>.

%-------------------------------------------------------------------------
% Declaration of endogenous variables
%-------------------------------------------------------------------------
var
c           ${c}$                     (long_name='consumption')
w           ${w}$                     (long_name='real wage')
pie         ${\pi}$                   (long_name='gross inflation')
n           ${n}$                     (long_name='hours worked')
R           ${R}$                     (long_name='nominal interest rate')
r           ${r}$                     (long_name='real interest rate')
y           ${y}$                     (long_name='output')
div         ${div}$                   (long_name='real profits')
Q           ${Q}$                     (long_name='bond price')
mc          ${mc}$                    (long_name='real marginal costs')
pstar       ${p^*}$                   (long_name='price inefficiency distortion')
ptilde      ${\widetilde{p}}$         (long_name='Optimal reset price')
s1          ${s_1}$                   (long_name='aux. sum 1 recursive price setting')
s2          ${s_2}$                   (long_name='aux. sum 2 recursive price setting')
a           ${a}$                     (long_name='technology level')
z           ${z}$                     (long_name='discount factor shifter')
nu          ${\nu}$                   (long_name='monetary policy shifter')
ahat        ${\widehat{a}}$           (long_name='technology level (log dev from ss)')
zhat        ${\widehat{z}}$           (long_name='preference shifter (log dev from ss)'
    )
yhat        ${\widehat{y}}$           (long_name='output (log dev from ss)')
what        ${\widehat{w}}$           (long_name='real wage (log dev from ss)')
nhat        ${\widehat{n}}$           (long_name='hours worked (log dev from ss)')
piehat_an   ${\widehat{\pi}^{ann}}$   (long_name='annualized inflation rate (log dev
    from ss)')
Rhat_an     ${\widehat{R}^{ann}}$     (long_name='annualized nominal interest rate (log
     dev from ss)')
rhat_an     ${\widehat{r}^{ann}}$     (long_name='annualized real interest rate (log
    dev from ss)')
mchat       ${\widehat{mc}}$          (long_name='real marginal costs (log dev from ss)
    ')
```

```
;
%----------------------------------------------------------------------------
% Declaration of exogenous variables (i.e. shocks)
%----------------------------------------------------------------------------
varexo
eps_a     ${\varepsilon_a}$      (long_name='technology shock')
eps_z     ${\varepsilon_z}$      (long_name='preference shock')
eps_nu    ${\varepsilon_\nu}$    (long_name='monetary policy shock')
;
%----------------------------------------------------------------------------
% Declaration of parameters
%----------------------------------------------------------------------------
parameters
BETA      ${\beta}$          (long_name='discount factor')
RHO_A     ${\rho_a}$         (long_name='autocorrelation technology process')
RHO_NU    ${\rho_{\nu}}$     (long_name='autocorrelation monetary policy process')
RHO_Z     ${\rho_{z}}$       (long_name='autocorrelation preference shock')
SIGMA     ${\sigma}$         (long_name='inverse elasticity of intertemporal substitution
    ')
VARPHI    ${\varphi}$        (long_name='inverse Frisch elasticity')
PHI_PIE   ${\phi_{\pi}}$     (long_name='inflation feedback Taylor Rule')
PHI_Y     ${\phi_{y}}$       (long_name='output feedback Taylor Rule')
EPSILON   ${\epsilon}$       (long_name='demand elasticity')
THETA     ${\theta}$         (long_name='Calvo parameter')
PIESTAR   ${\pi^*}$          (long_name='inflation target')
;

%----------------------------------------------------------------------------
% Model Equations
%----------------------------------------------------------------------------
%model(block,bytecode); % you can also use this
model;
%marginal utilities
#Un = -z*n^VARPHI;
#Uc = z*c^(-SIGMA);
#Ucp = z(+1)*c(+1)^(-SIGMA);

[name='labor supply']
w = -Un/Uc;

[name='Euler equation']
Uc = BETA*Ucp*r;

[name='optimal price setting']
ptilde*s1 = EPSILON/(EPSILON-1)*s2;

[name='optimal price setting auxiliary recursion 1']
s1 = y*Uc + BETA*THETA*pie(+1)^(EPSILON-1)*s1(+1);

[name='optimal price setting auxiliary recursion 2']
s2 = mc*y*Uc + BETA*THETA*pie(+1)^EPSILON*s2(+1);

[name='law of motion for optimal reset price']
1=THETA*pie^(EPSILON-1)+(1-THETA)*ptilde^(1-EPSILON);

[name='marginal costs / labor demand']
mc=w/a;

[name='real profits']
div=y-w*n;

[name='aggregate demand']
c=y;

[name='aggregate supply']
```

```
pstar*y = a*n;

[name='law of motion for price inefficiency distortion']
pstar = (1-THETA)*ptilde^(-EPSILON) + THETA*pie^EPSILON*pstar(-1);

[name='price of a zero-coupon bond']
Q=1/R;

[name='Fisher equation']
R = r*pie(+1);

[name='monetary policy rule']
R=steady_state(R)*(pie/PIESTAR)^PHI_PIE*(y/steady_state(y))^PHI_Y*exp(nu);

[name='preference shifter']
log(z) = RHO_Z*log(z(-1)) + eps_z;

[name='technology process']
log(a) = RHO_A*log(a(-1)) + eps_a;

[name='monetary policy shock process']
nu = RHO_NU*nu(-1) + eps_nu;

[name='definition log output (dev. from ss)']
yhat = log(y) - log(steady_state(y));

[name='definition log real wage (dev. from ss)']
what = log(w) - log(steady_state(w));

[name='definition log hours worked (dev. from ss)']
nhat = log(n) - log(steady_state(n));

[name='definition log annualized inflation (dev. from ss)']
piehat_an = 4*(log(pie)-log(steady_state(pie)));

[name='definition log annualized nominal interest rate (dev. from ss)']
Rhat_an = 4*(log(R)-log(steady_state(R)));

[name='definition log annualized real interest rate (dev. from ss)']
rhat_an = 4*(log(r)-log(steady_state(r)));

[name='definition log technology (dev. from ss)']
ahat = log(a) - log(steady_state(a));

[name='definition log discount factor shifter (dev. from ss)']
zhat = log(z) - log(steady_state(z));

[name='definition log real marginal costs (dev. from ss)']
mchat = log(mc) - log(steady_state(mc));
end;

%----------------------------------------------------------------
%  Steady state model
%----------------------------------------------------------------
steady_state_model;
z   = 1;
a   = 1;
nu  = 0;
pie = PIESTAR;
ptilde = ( (1-THETA*pie^(EPSILON-1))/(1-THETA) )^(1/(1-EPSILON));
mc = (EPSILON-1)/EPSILON * ptilde * (1-BETA*THETA*pie^EPSILON) / (1-BETA*THETA*pie^(
    EPSILON-1)) ;
pstar = (1-THETA)/(1-THETA*pie^EPSILON) * ptilde^(-EPSILON);
Q = BETA/pie;
R = 1/Q;
```

```
r = R/pie;
w = mc*a;
n = (w/(a/pstar)^SIGMA)^(1/(VARPHI+SIGMA));
y = a*n/pstar;
c = y;
div = y-w*n;
s1 = c^(-SIGMA)*y/(1-BETA*THETA*pie^(EPSILON-1));
s2 = c^(-SIGMA)*y*mc/(1-BETA*THETA*pie^EPSILON);

yhat=0;what=0;nhat=0;piehat_an=0;Rhat_an=0;rhat_an=0;ahat=0;zhat=0;mchat=0;
end;
```

new_keynesian_steady.mod:

progs/dynare/new_keynesian_steady.mod

```
@#include "new_keynesian_common.inc"

%--------------------------------------------------------------------------
% Parametrization for quarterly models (follows mostly Gali (2015) Chapter 3)
%--------------------------------------------------------------------------
ALPHA    = 1/4;  % implies steady state labor income share of 0.75
BETA     = 0.99; % implies quarterly gross interest rate of 1.0101 (under price
    stability)
RHO_A    = 0.9;  % high persistent TFP process
RHO_Z    = 0.5;  % mild persistent preference shifter process
RHO_NU   = 0.5;  % mild persistent monetary policy shock
SIGMA    = 1;    % log utility, i.e. substitution effect equals income effect
VARPHI   = 5;    % inverse Frisch elasticity
PHI_PIE  = 1.5;  % original value used by Taylor, needs to be above 1
PHI_Y    = 0.5/4;% original value used by Taylor
THETA    = 3/4;  % Calvo probability, implies average duration of 1/(1-THETA)=4
    quarters
EPSILON  = 9;    % implies (under price stability target) gross markup of EPSILON/(
    EPSILON-1)=1.125
PIESTAR  = 1+0.02/4; % central bank targets 2% annualy

%------------------------------------------------------------------
% compute steady state
%------------------------------------------------------------------
model_diagnostics;
steady;

%------------------------------------------------------------------
% optionally: compile latex
%------------------------------------------------------------------
write_latex_definitions;
write_latex_parameter_table;
write_latex_original_model;
write_latex_steady_state_model;
collect_latex_files;

path_to_pdflatex = '';
%path_to_pdflatex = '/usr/local/bin/'; % sometimes you need to adjust the path where
    pdflatex is, depending on your operating system
if system([path_to_pdflatex 'pdflatex -halt-on-error -interaction=batchmode ' M_.fname
     '_TeX_binder.tex'])
    warning('TeX-File did not compile; you need to compile it manually')
end
```

Run the mod file.