Question 1. Choleski identification and monetary policy shocks.

In the dataset, in the first sheet of `data_ps3.xlsx`, you will find the following variables: log(real GDP), log(Price deflator) and Federal Funds Rate for the US from 1960 to 2010 at quarterly frequency. Estimate a VAR on those variables (do not worry about the potential non-stationarity), identify the monetary policy shock with a triangular (Cholesky) identification structure with the ordering above, and compute impulse responses and variance decompositions for the monetary policy shock. Compute confidence intervals around the point estimates for the impulse responses using a *bootstrap*.

(Hint: a bootstrap is similar a Monte Carlo, but instead of sampling from a known distribution, you draw from the estimated residuals. The steps are the following:

1. *Estimate the VAR (suppose with a constant) on the N variables in y_t and store the coefficients (with the constants collected in the $(N \times 1)$ vector $\hat{c}$ and the $(N \times N)$ autoregressive coefficients in the matrix $\hat{A}$. If there are p lags, you can express the VAR(p) as a VAR(1) with the companion form) and the $(T \times N)$ residuals $\hat{\varepsilon}$ (note that T is the number of estimated residuals, those obtained once one eliminated the lags lost in the estimation).*
2. *Sample* ***with replacement*** *from the estimated residuals so to form a new series of residuals $\tilde{\varepsilon}$ of dim $(T \times N)$ (note 1: sample with replacement; note 2: sample one entire row (of dim $(1 \times N)$) of the matrix $\hat{\varepsilon}$); One way of doing it is generate T random integers from 1 to T (for example, call the random draw of integers `PER`) and then set $\tilde{\varepsilon} = \hat{\varepsilon}(PER, :)$; don't use the command `permute` (why?).*
3. *Use the newly generated residuals and the estimated coefficients to construct new series: $\tilde{y}_t = \hat{c} + \hat{A}\,\tilde{y}_{t-1} + \tilde{\varepsilon}_t$. The starting values are the first values of y_t (in the case of a VAR(1), just y_1).*
4. *Estimate a VAR on the new series $\tilde{y}_t$; identify shocks and compute impulse responses and variance decompositions. Store the impulse responses and variance decompositions.*
5. *Repeat steps 2. to 4. K times (say, $K = 1000$);*
6. *At the end you have a set of 1000 impulse responses. Plot the 2.5% and 97.5% percentile (command `prctile`) of that empirical distribution. This is your 95% confidence interval.)*

Solution.

Cholesky identification and monetary policy shocks

1.1 Data and objective

In the first sheet of the dataset `data_ps3.xlsx` we observe three U.S. quarterly macroeconomic time series from 1960Q1 to 2010Q4: the log of real GDP, the log of the price deflator, and the Federal Funds Rate. We collect these variables in the 3×1 vector

$$\underline{\mathbf{y}}_t = \begin{pmatrix} \log Y_t \\ \log P_t \\ i_t \end{pmatrix}, \qquad t = 1960\text{Q1}, \ldots, 2010\text{Q4}.$$

The goal of this exercise is to estimate a VAR model for $\underline{\mathbf{y}}_t$, identify a monetary policy shock by means of a triangular (Cholesky) identification consistent with the ordering $(\log Y_t, \log P_t, i_t)$, compute impulse responses and forecast error variance decompositions for this shock, and construct bootstrap confidence intervals for the impulse responses following the algorithm described in the problem set.

```

%% =============================================================
%  LOAD DATA
%  Variables in sheet 1:
%    - LOG_GDP_  : log real GDP
%    - LOG_P_    : log price deflator
%    - FFR       : federal funds rate
%% =============================================================

TBL = readtable('data_ps3.xlsx','Sheet',1);

logY = TBL.LOG_GDP_;
logP = TBL.LOG_P_;
FFR  = TBL.FFR;

% Stack series: y_t = [logY_t, logP_t, FFR_t]
y = [logY, logP, FFR];            % T x N matrix
[T, N] = size(y);                 % N = 3

p = 4;                            % VAR(4), quarterly data

% Build dependent and regressor matrices
Y = y(p+1:end, :);                % dependent variables (Teff x N)
T_eff = size(Y,1);

X = [];
for i = 1:p
X = [X, y(p+1-i:end-i,:)];  % lags y_{t-1},...,y_{t-p}
end
X = [ones(T_eff,1), X];           % add constant
k = size(X,2);                    % number of regressors (1 + N*p)
```

1.2 Reduced-form VAR(4)

1.2.1 Model specification

We model the joint dynamics of the three variables with a VAR of order $p = 4$ and a constant term:

$$\underline{\mathbf{y}}_t = \underline{\mathbf{c}} + A_1\underline{\mathbf{y}}_{t-1} + A_2\underline{\mathbf{y}}_{t-2} + A_3\underline{\mathbf{y}}_{t-3} + A_4\underline{\mathbf{y}}_{t-4} + \underline{\mathbf{u}}_t, \tag{1}$$

where

- $\underline{\mathbf{c}}$ is a 3×1 vector of intercepts;
- A_j are 3×3 coefficient matrices, $j = 1, \dots, 4$;
- $\underline{\mathbf{u}}_t$ is a 3×1 vector of reduced-form residuals with covariance matrix
$$\Sigma_u = \mathbb{E}[\underline{\mathbf{u}}_t\underline{\mathbf{u}}_t'] \in \mathbb{R}^{3\times 3}.$$

1.2.2 Stacked regression form and estimation

Let T denote the total number of observations and $p = 4$ the lag length. We define the effective sample size $T_{\text{eff}} = T - p$ and stack the dependent variables as

$$Y = \begin{bmatrix} \underline{\mathbf{y}}'_{p+1} \\ \underline{\mathbf{y}}'_{p+2} \\ \vdots \\ \underline{\mathbf{y}}'_T \end{bmatrix} \in \mathbb{R}^{T_{\text{eff}}\times 3}.$$

The regressor matrix X collects a constant and four lags of $\underline{\mathbf{y}}_t$:

$$X = \begin{bmatrix} 1 & \underline{\mathbf{y}}'_p & \underline{\mathbf{y}}'_{p-1} & \underline{\mathbf{y}}'_{p-2} & \underline{\mathbf{y}}'_{p-3} \\ 1 & \underline{\mathbf{y}}'_{p+1} & \underline{\mathbf{y}}'_p & \underline{\mathbf{y}}'_{p-1} & \underline{\mathbf{y}}'_{p-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \underline{\mathbf{y}}'_{T-1} & \underline{\mathbf{y}}'_{T-2} & \underline{\mathbf{y}}'_{T-3} & \underline{\mathbf{y}}'_{T-4} \end{bmatrix} \in \mathbb{R}^{T_{\text{eff}} \times 13},$$

where $13 = 1 + 3 \times 4$ is the number of regressors (constant plus four lags of three variables).

The reduced-form VAR can be written compactly as

$$Y = XB + U,$$

with $B \in \mathbb{R}^{13\times 3}$ the coefficient matrix and $U \in \mathbb{R}^{T_{\text{eff}}\times 3}$ the matrix of residuals. Ordinary least squares gives

$$\widehat{B} = (X'X)^{-1}X'Y, \qquad \widehat{U} = Y - X\widehat{B}.$$

From $\widehat{B}$ we extract:

- the intercept vector $\widehat{\underline{\mathbf{c}}}$ (first row of $\widehat{B}$, transposed, dimension 3×1);
- the lag coefficient matrices $\widehat{A}_1, \dots, \widehat{A}_4$ (each 3×3), obtained by reshaping the remaining 12 rows;
- the residual covariance matrix with degrees-of-freedom correction,

$$\widehat{\Sigma}_u = \frac{1}{T_{\text{eff}} - k}\widehat{U}'\widehat{U}, \qquad k = 13.$$

```

%% ============================================================
%  POINT 1 Estimate VAR, store coefficients + residuals
%% ============================================================

% OLS estimation: Y = X * B + U
B = X \ Y;                          % coefficient matrix (k x N)
U = Y - X*B;                        % residuals (Teff x N)

% Extract constant and lag coefficient matrices
c_hat = B(1,:)';                    % N x 1 constant vector
A_hat = zeros(N, N*p);              % [A1 A2 A3 A4]

for i = 1:p
rows = 1 + (i-1)*N + (1:N);                     % rows in B for lag i
A_hat(:, (1+(i-1)*N):(i*N)) = B(rows,:)';      % each block is N x N
end

% Residual covariance matrix with DoF correction
SigmaU = (U' * U)/(T_eff - k);  % N x N
```

1.2.3 Estimated coefficients and residual covariance

The estimated coefficient matrix $\widehat{B}$ (constant and four lags of $\log Y_t$, $\log P_t$ and i_t) is:

Table 1-1: Estimated VAR(4) coefficients: matrix $\widehat{B}$

	$\log Y_t$	$\log P_t$	i_t
Constant	0.1601	-0.0281	3.2571
$\log Y_{t-1}$	1.2114	0.0039	28.4197
$\log P_{t-1}$	0.2913	1.5062	42.0864
i_{t-1}	0.0004	0.0005	1.1296
$\log Y_{t-2}$	-0.0450	-0.0021	-13.9429
$\log P_{t-2}$	-0.3006	-0.3661	10.9574
i_{t-2}	-0.0037	-0.0002	-0.5062
$\log Y_{t-3}$	-0.2125	0.0258	-14.2392
$\log P_{t-3}$	-0.2156	0.0654	-91.3392
i_{t-3}	0.0035	-0.0001	0.4770
$\log Y_{t-4}$	0.0222	-0.0225	-0.8865
$\log P_{t-4}$	0.2399	-0.2098	38.8586
i_{t-4}	-0.0010	0.0000	-0.2023

The residual covariance matrix $\widehat{\Sigma}_u$ is estimated as

$$\widehat{\Sigma}_u = \begin{pmatrix} 0.0001 & -0.0000 & 0.0009 \\ -0.0000 & 0.0000 & 0.0004 \\ 0.0009 & 0.0004 & 0.7182 \end{pmatrix}.$$

The off-diagonal elements document substantial contemporaneous correlation between the residuals of the interest rate equation and those of the macro variables, motivating the need for an explicit structural identification.

1.3 Structural identification via Cholesky (Phase A)

1.3.1 Economic assumptions and structural shocks

The residuals $\underline{\mathbf{u}}_t$ from (1) are reduced-form innovations: they are linear combinations of underlying structural shocks and are contemporaneously correlated. To interpret innovations in the Federal Funds Rate as monetary policy shocks, we impose a recursive structure consistent with the ordering

$$(\log Y_t, \log P_t, i_t).$$

This implies that real activity reacts contemporaneously to all shocks, prices react contemporaneously to real shocks but not to monetary policy shocks, and the central bank can react contemporaneously to both output and prices.

Operationally, we factorize the estimated covariance matrix via a Cholesky decomposition:

$$\widehat{\Sigma}_u = PP',$$

where P is lower triangular and 3×3. We then define the vector of structural shocks as

$$\underline{\boldsymbol{\varepsilon}}_t = P^{-1}\underline{\mathbf{u}}_t, \qquad \mathbb{E}[\underline{\boldsymbol{\varepsilon}}_t \underline{\boldsymbol{\varepsilon}}_t'] = I_3.$$

The third component of $\underline{\boldsymbol{\varepsilon}}_t$ is interpreted as the monetary policy shock.

1.3.2 Companion form and impulse responses

We obtain impulse responses from the companion representation of the VAR. Define the 12×1 state vector

$$\underline{\mathbf{z}}_t = \begin{pmatrix} \underline{\mathbf{y}}_t \\ \underline{\mathbf{y}}_{t-1} \\ \underline{\mathbf{y}}_{t-2} \\ \underline{\mathbf{y}}_{t-3} \end{pmatrix},$$

and the 12×12 companion matrix

$$F = \begin{pmatrix} \widehat{A}_1 & \widehat{A}_2 & \widehat{A}_3 & \widehat{A}_4 \\ I_3 & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 \end{pmatrix}.$$

Ignoring the constant for impulse-response analysis, the state-space form is

$$\underline{\mathbf{z}}_t = F\underline{\mathbf{z}}_{t-1} + \underline{\tilde{\mathbf{u}}}_t, \qquad \underline{\tilde{\mathbf{u}}}_t = \begin{pmatrix} P\underline{\boldsymbol{\varepsilon}}_t \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

Let

$$J = \begin{bmatrix} I_3 & 0_{3\times 9} \end{bmatrix}$$

be the 3×12 selection matrix that maps the state vector back to the observable variables: $\underline{\mathbf{y}}_t = J\underline{\mathbf{z}}_t$. For a one-standard-deviation structural shock, the impulse response at horizon h is

$$IRF(h) = JF^h J' P, \qquad h = 0, 1, \ldots, H,$$

where we set $H = 20$ quarters. The column j of $IRF(h)$ gives the responses to structural shock j; in particular, the monetary policy shock corresponds to $j = 3$.

```
%% ============================================================
%  BASELINE STRUCTURAL IRFs FROM ORIGINAL VAR (PHASE A)
%% ============================================================

% Cholesky factor (triangular identification): SigmaU = P0 * P0'
P0 = chol(SigmaU, 'lower');     % N x N

% Companion form of VAR(4)
F0 = zeros(N*p);
F0(1:N, :) = A_hat;                                % first block row
F0(N+1:end, 1:N*(p-1)) = eye(N*(p-1));            % subdiagonal identity blocks

% Selector matrix J: picks y_t from the companion state vector z_t
J = [eye(N), zeros(N, N*(p-1))];                  % N x (N*p)

% IRF horizon
H = 20;                                           % 20 quarters
IRF0 = zeros(H+1, N, N);                          % IRF0(h+1, i, j)

Fpow = eye(N*p);                                  % F^0

for h = 0:H
Theta_h = J * Fpow * J' * P0;             % N x N
IRF0(h+1,:,:) = Theta_h;
Fpow = Fpow * F0;                         % update F^{h+1}
end
```

```
% Monetary policy shock = 3rd structural shock (FFR)
mp_shock = 3;
IRF_baseline_mp = squeeze(IRF0(:,:,mp_shock)); % (H+1) x N
```

1.3.3 Baseline impulse responses

The baseline impulse responses of $\log Y_t$, $\log P_t$ and i_t to a contractionary monetary policy shock (one-standard-deviation increase in the structural innovation of the Federal Funds Rate) are reported in Figure 1-1.

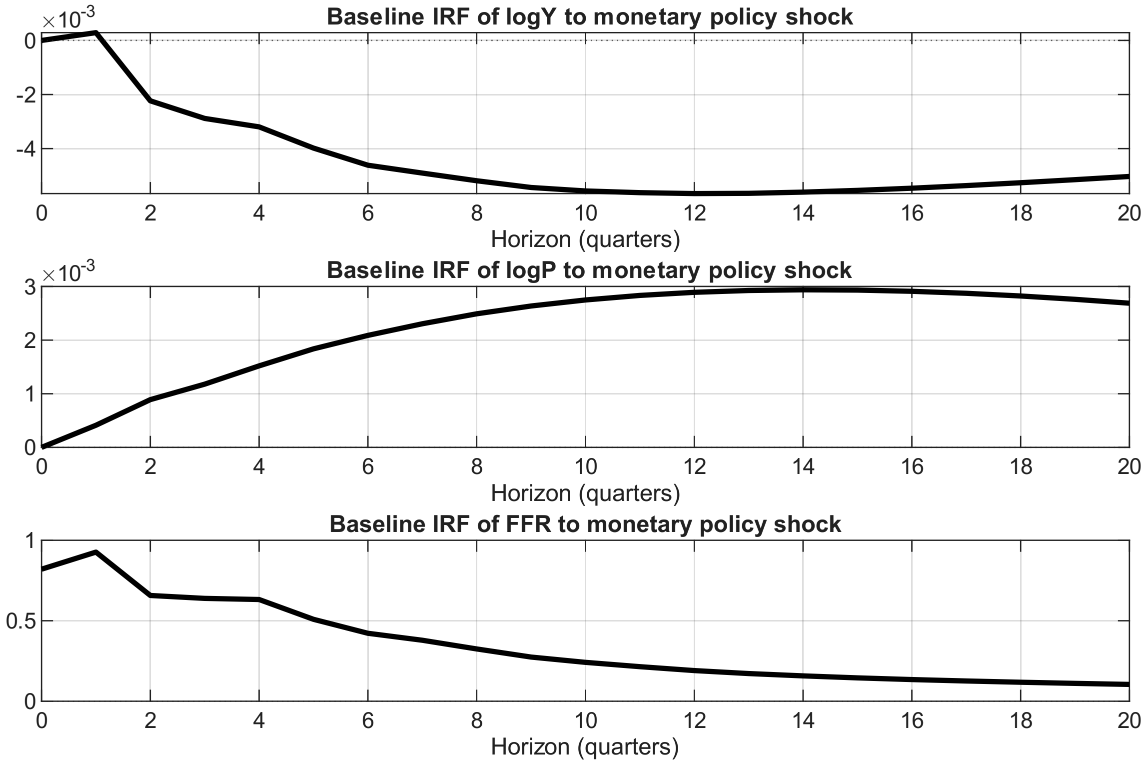


Figure 1-1: Baseline impulse responses of $\log Y_t$, $\log P_t$ and i_t to a monetary policy shock.

The responses display the usual pattern found in monetary VARs. Output reacts negatively, with a small but persistent decline peaking around two years after the shock. Prices increase slightly on impact and then gradually decelerate, eventually converging back towards zero. The Federal Funds Rate rises sharply on impact (by about 0.82 percentage points in our normalization) and then slowly declines towards its pre-shock level.

1.4 Forecast error variance decomposition

Given the structural impulse responses, we compute the share of forecast error variance of each variable explained by each structural shock at horizon h:

$$\text{FEVD}_{i,j}(h) = \frac{\sum_{s=0}^{h} IRF_{i,j}(s)^2}{\sum_{k=1}^{3}\sum_{s=0}^{h} IRF_{i,k}(s)^2}, \qquad i,j = 1,2,3.$$

We focus on the contribution of the monetary policy shock (shock $j = 3$). Table 1-2 reports the forecast error variance shares for horizons $h = 1, 4, 8, 16$ quarters.

Table 1-2: Share of forecast error variance explained by the monetary policy shock

Horizon (quarters)	$\log Y_t$	$\log P_t$	i_t
1	0.001	0.007	0.856
4	0.055	0.028	0.620
8	0.150	0.035	0.502
16	0.292	0.032	0.419

Monetary policy shocks account for the bulk of the forecast error variance of the Federal Funds Rate at short horizons (around 86% at one quarter). Their contribution to output and prices is modest initially, but rises over longer horizons: for instance, at a four-year horizon the monetary policy shock explains about 29% of the variance of $\log Y_t$, while its contribution to the variance of $\log P_t$ remains small.

```
%% =============================================================
%  BASELINE FEVD (FORECAST ERROR VARIANCE DECOMPOSITION)
%% =============================================================

FEVD0 = zeros(H+1, N, N);   % FEVD0(h+1, i, j)

for h = 0:H
irfs_0toh = IRF0(1:h+1,:,:);   % stack IRFs from 0 to h
for i = 1:N       % variable index
num = zeros(1,N);
for j = 1:N   % shock index
num(j) = sum( squeeze(irfs_0toh(:,i,j)).^2 );
end
FEVD0(h+1,i,:) = num / sum(num);
end
end

% FEVD due to the monetary policy shock (3rd shock)
fevd_baseline_mp = squeeze(FEVD0(:,:,mp_shock));  % (H+1) x N
```

1.5 Bootstrap confidence intervals (Phase B)

1.5.1 Bootstrap algorithm

To quantify the uncertainty around the impulse responses we implement the residual bootstrap procedure described in the problem set. The steps are:

1. *Store coefficients and residuals.* Estimate the VAR(4) on the original data and store $\underline{\widehat{\mathbf{c}}}$, $\widehat{A}_1, \dots, \widehat{A}_4$ and the residuals $\underline{\widehat{\mathbf{u}}}_t$.

2. *Sample residuals with replacement.* Construct a bootstrap residual series $\underline{\tilde{\mathbf{u}}}_t$ by sampling entire rows of the residual matrix with replacement. If $\mathrm{PER}(t)$ denotes the resampled index at date t, we set
$$\underline{\tilde{\mathbf{u}}}_t = \underline{\widehat{\mathbf{u}}}_{\mathrm{PER}(t)}.$$

3. *Simulate a bootstrap dataset.* Generate $\underline{\tilde{\mathbf{y}}}_t$ recursively according to
$$\underline{\tilde{\mathbf{y}}}_t = \underline{\widehat{\mathbf{c}}} + \sum_{j=1}^{4} \widehat{A}_j \underline{\tilde{\mathbf{y}}}_{t-j} + \underline{\tilde{\mathbf{u}}}_t,$$
starting from the first p observations of the original sample as initial conditions.

4. *Re-estimate the VAR and re-identify the shocks.* Estimate a VAR(4) with a constant on $\underline{\tilde{\mathbf{y}}}_t$, compute the bootstrap residual covariance matrix, apply again the Cholesky decomposition, and obtain bootstrap structural impulse responses $IRF^{(b)}(h)$.

5. *Repeat.* Repeat steps 2–4 for $K = 1000$ bootstrap replications, storing for each replication the impulse responses of $\log Y_t$, $\log P_t$ and i_t to the monetary policy shock.

1.5.2 Bootstrap confidence bands

For each variable i and horizon h, we obtain a bootstrap distribution of impulse responses $\{IRF^{(b)}_{i,3}(h)\}_{b=1}^{K}$. The 95% pointwise confidence bands are given by the percentile intervals

$$\mathrm{CI}_{95\%}(h) = \Big[\mathrm{prctile}(IRF^{(b)}_{i,3}(h),\, 2.5),\ \mathrm{prctile}(IRF^{(b)}_{i,3}(h),\, 97.5)\Big].$$

Figure 1-2 displays the baseline impulse responses together with the 95% bootstrap bands. The bands are relatively narrow for the interest rate, reflecting the fact that the monetary policy shock explains a large fraction of the variance of i_t, and somewhat wider for output and prices, especially at medium horizons.

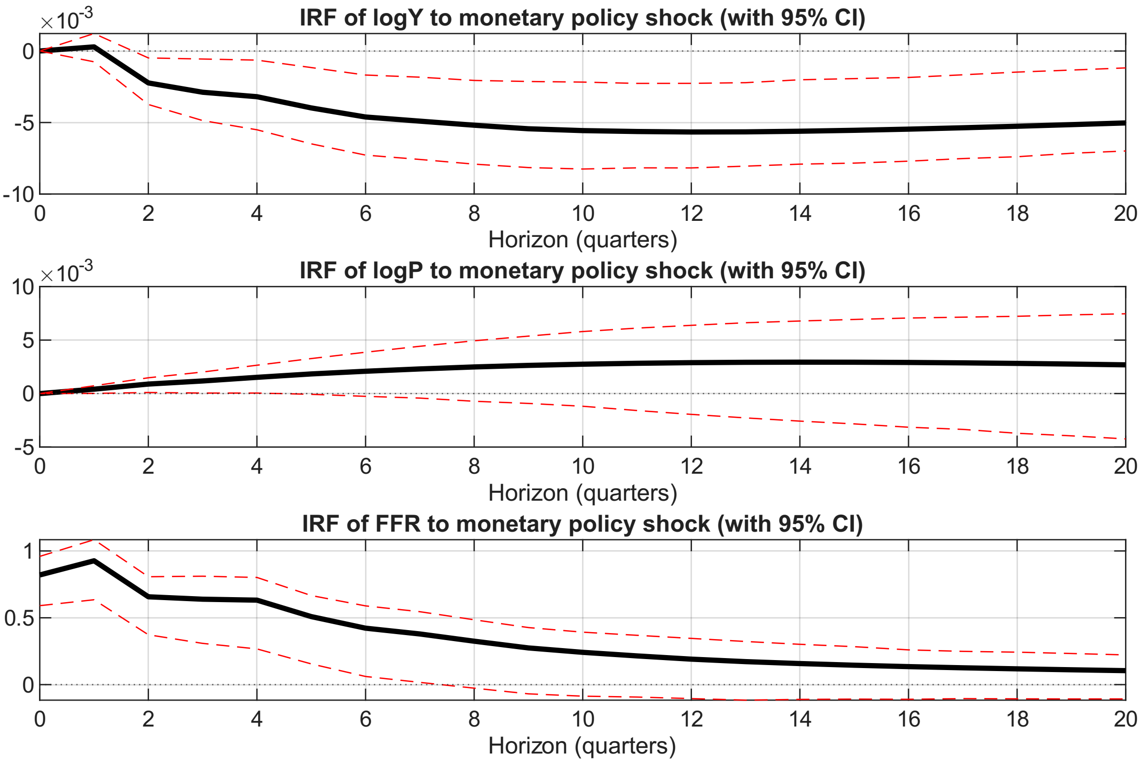


Figure 1-2: Impulse responses of $\log Y_t$, $\log P_t$ and i_t to a monetary policy shock with 95% bootstrap confidence bands.

```
%% ============================================================
%  POINT 2 Sample WITH replacement from residuals (FIRST DRAW)
%% ============================================================

PER = randi(T_eff, T_eff, 1);        % T_eff integers from 1..T_eff
U_tilde = U(PER, :);                 % bootstrap residuals

%% ============================================================
%  POINT 3 Simulate new series y_tilde (FIRST DRAW)
%% ============================================================
```

```

% Use original first p observations as starting values
y_tilde = zeros(T, N);
y_tilde(1:p, :) = y(1:p, :);        % initial conditions

for t = p+1:T
% build [1, y_{t-1},...,y_{t-p}]
x_t = 1;
for j = 1:p
x_t = [x_t, y_tilde(t-j, :)];
end
y_tilde(t, :) = x_t * B + U_tilde(t-p,:);   % simulated data
end

%% ============================================================
%  POINT 4 Estimate VAR on simulated series, compute IRFs (FIRST DRAW)
%% ============================================================

% Build Yb, Xb from y_tilde
Yb = y_tilde(p+1:end, :);
Xb = [];
for i = 1:p
Xb = [Xb, y_tilde(p+1-i:end-i,:)];
end
Xb = [ones(T_eff,1), Xb];

% Re-estimate VAR on bootstrap sample
Bb = Xb \ Yb;
Ub = Yb - Xb * Bb;
SigmaUb = (Ub' * Ub)/(T_eff - k);

% Extract A_hat_b from Bb
A_hat_b = zeros(N, N*p);
for i = 1:p
rows = 1 + (i-1)*N + (1:N);
A_hat_b(:, (1+(i-1)*N):(i*N)) = Bb(rows,:)';
end

% Cholesky identification on bootstrap sample
P_b = chol(SigmaUb, 'lower');

% Companion form for bootstrap VAR
F_b = zeros(N*p);
F_b(1:N,:) = A_hat_b;
F_b(N+1:end, 1:N*(p-1)) = eye(N*(p-1));

IRF_b = zeros(H+1, N, N);
Fpow = eye(N*p);
for h = 0:H
Theta_h = J * Fpow * J' * P_b;
IRF_b(h+1,:,:) = Theta_h;
Fpow = Fpow * F_b;
end

%% ============================================================
%  POINT 5 Repeat steps 2-4 K times, build bootstrap distribution
%% ============================================================

K = 1000;                           % number of bootstrap replications
IRF_boot = zeros(K, H+1, N, N);    % store full IRFs

disp('Running bootstrap replications...');

```

```
for k_boot = 1:K

% --- Step 2: sample residuals with replacement ---
PER = randi(T_eff, T_eff, 1);
U_tilde = U(PER, :);

% --- Step 3: simulate new data under estimated VAR ---
y_tilde = zeros(T, N);
y_tilde(1:p, :) = y(1:p, :);  % initial conditions

for t = p+1:T
    x_t = 1;
    for j = 1:p
        x_t = [x_t, y_tilde(t-j,:)];
    end
    y_tilde(t,:) = x_t * B + U_tilde(t-p,:);
end

% --- Step 4: re-estimate VAR and compute structural IRFs ---
Yb = y_tilde(p+1:end,:);
Xb = [];
for i = 1:p
    Xb = [Xb, y_tilde(p+1-i:end-i,:)];
end
Xb = [ones(T_eff,1), Xb];

Bb = Xb \\ Yb;
Ub = Yb - Xb*Bb;
SigmaUb = (Ub' * Ub)/(T_eff - k);

% extract A_hat_b
A_hat_b = zeros(N, N*p);
for i = 1:p
    rows = 1 + (i-1)*N + (1:N);
    A_hat_b(:, (1+(i-1)*N):(i*N)) = Bb(rows,:)';
end

% companion and Cholesky for bootstrap VAR
F_b = zeros(N*p);
F_b(1:N,:) = A_hat_b;
F_b(N+1:end,1:N*(p-1)) = eye(N*(p-1));

P_b = chol(SigmaUb, 'lower');

Fpow = eye(N*p);
for h = 0:H
    IRF_boot(k_boot,h+1,:,:) = J * Fpow * J' * P_b;
    Fpow = Fpow * F_b;
end

end

%% ============================================================
%  COMPUTE 95% CONFIDENCE BANDS FOR MONETARY POLICY IRFs
%% ============================================================

% Baseline IRFs (original VAR) already in IRF_baseline_mp
IRF_original = IRF_baseline_mp;   % (H+1) x N

% Extract bootstrap IRFs for monetary policy shock (3rd shock)
IRF_boot_mp = squeeze(IRF_boot(:,:,:,mp_shock));  % K x (H+1) x N

ci_low   = zeros(H+1,N);
```

```
ci_high = zeros(H+1,N);

for i = 1:N
tmp = squeeze(IRF_boot_mp(:,:,i)); % K x (H+1)
ci_low(:,i)  = prctile(tmp,  2.5, 1)';
ci_high(:,i) = prctile(tmp, 97.5, 1)';
end
```

1.6 Summary

Overall, the estimated VAR(4) with triangular identification delivers impulse responses that conform to standard macroeconomic intuition. A contractionary monetary policy shock raises the Federal Funds Rate on impact and then gradually brings it back towards its steady state, generates a small but persistent decline in real output, and has a moderate, delayed effect on prices. The forecast error variance decomposition shows that monetary policy shocks are the dominant source of short-run fluctuations in the interest rate, while they explain a non-trivial but limited portion of real activity at longer horizons. Bootstrap confidence intervals confirm that these qualitative patterns are statistically robust.

□

Question 2. Long-run identification.

Read the paper "Technology, employment, and the business cycle: do technology shocks explain aggregate fluctuations?" by Jordi Galí, *The American Economic Review*, Vol. 89, No. 1 (Mar., 1999), pp. 249–271.

Using the dataset in the second sheet of `data_ps3.xlsx`, replicate Figure 2 in the paper and compute bootstrapped confidence bands *(warning: results may slightly differ from those in the paper, as the data are not exactly the same as in the paper)*.

Solution.

Step 1: VAR estimation in differences. To replicate Galì's estimated impulse responses, we first need to import quarterly series. We mapped the code variables to standard notation in the following way:

$$x_t \equiv \text{labor productivity} \equiv \texttt{outperh}, \qquad n_t \equiv \text{hours worked} \equiv \texttt{hours}.$$

Following Galí (1999), we estimate the VAR in first differences of productivity and hours. This will allow us to implement long-run identification on levels via restrictions on which variables affect permanently productivity (Blanchard–Quah/Galí). Formally, let $\Delta \equiv 1 - L$ denote the difference operator. We construct:

$$\Delta x_t \;=\; x_t - x_{t-1}, \qquad \Delta n_t \;=\; n_t - n_{t-1}, \qquad Y_t \;=\; \begin{bmatrix} \Delta x_t \\ \Delta n_t \end{bmatrix}.$$

In the code this corresponds to `dx = diff(x)`, `dn = diff(h)`, and `Y = [dx, dn]`. If x_t and n_t are in logs, Δx_t and Δn_t can be read as quarterly growth rates of productivity and hours. The vector Y_t is the dependent variable in the reduced-form VAR estimated below; after identifying the structural shocks we will cumulate the impulse responses of Δx_t and Δn_t to obtain level responses for x_t and n_t, and we will also report the response of output $y_t = x_t + n_t$ (since $x_t = y_t - n_t$).

```
%% Import data and define VAR variables
fname = 'data_ps3.xlsx';
Tbl = readtable(fname,'Sheet','technology_shock','VariableNamingRule','preserve
    ');
Tbl.Properties.VariableNames = {'date','outperh','hours'};

time = Tbl.date;
x    = Tbl.outperh;
h    = Tbl.hours;

% differencing
dx = diff(x);
dn = diff(h);
Y  = [dx, dn];
dtime = time(2:end);
```

We then procede to estimate a VAR(p) with an intercept on the vector Y_t. Formally, the VAR with constant is:

$$Y_t = c + \sum_{\ell=1}^{p} A_\ell Y_{t-\ell} + u_t, \quad u_t \sim (0, \Sigma_u), \tag{2}$$

We set $p = 4$ for quarterly data, in line with Gali's graphical representation. For $t = p+1, \dots, T$, define the regressor vector:

$$X_t = \begin{bmatrix} 1 \\ Y_{t-1} \\ \vdots \\ Y_{t-p} \end{bmatrix}$$

We define $Y_t \equiv (Y_{p+1}, \dots, Y_T)'$ and construct a `for` loop to retrieve $X \equiv (X_{p+1}, \dots, X_T)'$. The OLS estimator of the $k \times 2$ coefficient matrix:

$$B \equiv \begin{bmatrix} c' & A_1' & \cdots & A_p' \end{bmatrix}'$$

is

$$\widehat{B} = (X'X)^{-1}X'Y_t, \qquad \widehat{U} = Y_t - X\widehat{B},$$

with $\widehat{U}$ containing the stacked reduced-form residuals $\hat{u}_t$. The residual covariance is estimated with the usual DoF correction:

$$\widehat{\Sigma}_u = \frac{1}{T_{\text{eff}} - k}\widehat{U}'\widehat{U}, \qquad T_{\text{eff}} \equiv T - p.$$

We store the intercept $\hat{c}$, the lag matrices $\{\hat{A}_\ell\}_{\ell=1}^{p}$, and the residuals $\{\hat{u}_t\}$. This will be useful for later steps.

```
%% VAR(p) on [dx dn] with constant
[Tobs, nvar] = size(Y);
p = 4;                         % quarterly data

%stacked regression form
T_eff = Tobs - p;
Yt = Y(p+1:end, :);
X  = ones(T_eff, 1);
for L = 1:p
    X = [X, Y(p+1-L:end-L, :)];
end

% OLS
B = X \ Yt;
U = Yt - X*B;
```

```
SigmaU = (U' * U) / (T_eff - size(X,2));

% parameters for later
c_hat    = B(1, :)';
A_hat    = B(2:end, :)';
A_cells = mat2cell(A_hat, nvar, nvar*ones(1,p));

resid_hat = U;
Y_init    = Y(1:p, :);
```

Step 2: Galí identification and IRFs in levels. Let I denote the 2×2 identity. We first compute the long-run multiplier of the VAR in differences:

$$\widehat{C}(1) \;\equiv\; \sum_{h=0}^{\infty} \widehat{\Phi}_h \;=\; (I - \widehat{A}_1 - \cdots - \widehat{A}_p)^{-1},$$

where $\{\widehat{\Phi}_h\}$ are the reduced-form MA coefficients implied by $\{\widehat{A}_\ell\}$. This is implemented as:

$$\texttt{A\_sum} = \sum_{\ell=1}^{p} \widehat{A}_\ell, \qquad \widehat{C}(1) = (I - \texttt{A\_sum})^{-1}.$$

We then procede to impose Galí's long-run restriction. Pick the structural shock vector $\varepsilon_t = (\varepsilon_t^{\text{tech}}, \varepsilon_t^{\text{nontech}})' \sim (0, I_2)$ and a contemporaneous mapping $u_t = B\,\varepsilon_t$ so that $\Sigma_u = BB'$. Because the VAR is in differences, the long-run effect on levels of the structural shocks is:

$$\sum_{h=0}^{\infty} \Theta_h \;=\; \sum_{h=0}^{\infty} \Phi_h B \;=\; C(1)\,B.$$

Galí assumes that *only the technology shock has a permanent effect on productivity*. With the ordering (x_t, n_t), this means the $(1,2)$ element of the long-run impact on levels is zero:

$$C(1)\,B \;=\; \begin{bmatrix} \star & 0 \\ \star & \star \end{bmatrix} \quad \text{(lower triangular)}.$$

Imposing both $BB' = \Sigma_u$ and $C(1)B$ lower-triangular is achieved by taking the Cholesky factor of the long-run covariance:

$$L \;\equiv\; \texttt{chol}(C(1)\,\Sigma_u\,C(1)') \;\; \text{(lower)}, \qquad \Rightarrow \qquad C(1)\,\Sigma_u\,C(1)' = LL'.$$

Setting

$$\widehat{B} \;=\; \widehat{C}(1)^{-1} L$$

ensures the two estimated matrices are lower-triangular, i.e., the non-technology shock has no long-run effect on productivity. This pins down the structural impact matrix and normalizes $\text{Var}(\varepsilon_t) = I_2$, so all IRFs are to one–standard-deviation structural shocks.

We then procede to compute reduced-form MA(∞) and structural IRFs. We estimate $\{\widehat{\Phi}_h\}_{h=0}^{H}$ recursively from the VAR:

$$\Phi_0 = I, \qquad \Phi_h = \sum_{j=1}^{\min(h,p)} A_j \Phi_{h-j} \quad (h \geq 1),$$

for a horizon $H = 12$ quarters (as in Galí's figure). Structural IRFs for differences are:

$$\Theta_h \;=\; \Phi_h\,\widehat{B}, \qquad h = 0, \ldots, H,$$

and level IRFs are their cumulative sums,

$$\text{LevelTheta}_h = \sum_{s=0}^{h} \Theta_s, \qquad h = 0, \dots, H,$$

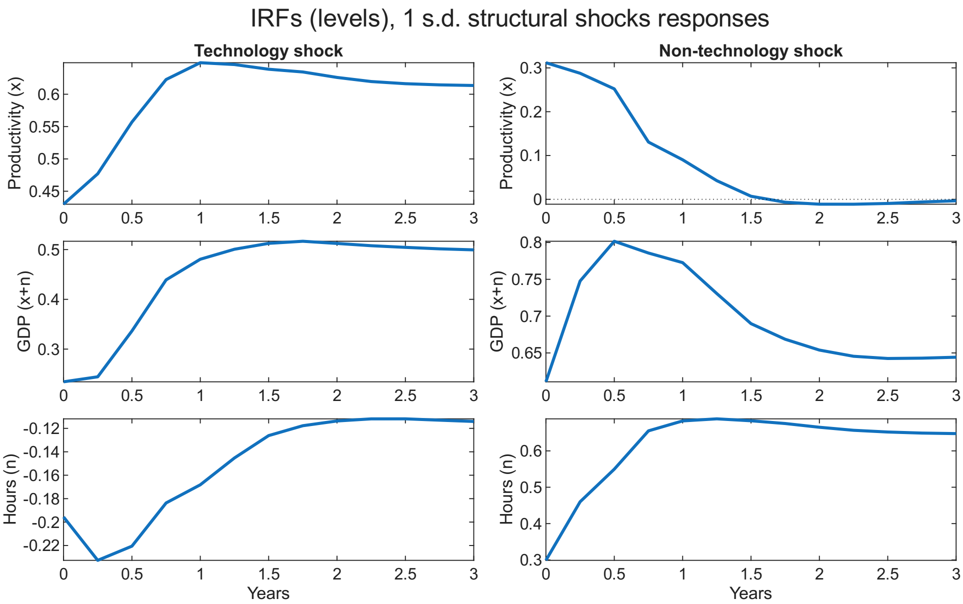
which deliver responses of (x_t, n_t) to each structural shock. We are finally able to plot the tech and nontech impulse responses. With columns of $\widehat{B}$ ordered as (tech, nontech), the code extracts:

$$\texttt{prod\_tech}(h) = [\text{LevelTheta}_h]_{1,\,\text{tech}}, \qquad \texttt{hours\_tech}(h) = [\text{LevelTheta}_h]_{2,\,\text{tech}},$$

$$\texttt{prod\_ntech}(h) = [\text{LevelTheta}_h]_{1,\,\text{nontech}}, \qquad \texttt{hours\_ntech}(h) = [\text{LevelTheta}_h]_{2,\,\text{nontech}}$$

$$\texttt{gdp\_tech}(h) = \texttt{prod\_tech}(h) + \texttt{hours\_tech}(h), \qquad \texttt{gdp\_ntech}(h) = \texttt{prod\_ntech}(h) + \texttt{hours\_ntech}(h).$$

These six series are the panels to be plotted to replicate Galí (1999), Figure 2, without bootstrapped confidence bands (yet).




```
%% Long-run identification (Gali)
nvar = size(Y,2);
I    = eye(nvar);

% long-run multipliers on \Delta y
A_sum = zeros(nvar);
for j = 1:p
    A_sum = A_sum + A_cells{j};
end
C1 = inv(I - A_sum);                     % C(1) = (I - A1 - ... - Ap)^{-1}

% identification: Gali restriction "only tech has permanent effect on productivity"
% the long-run impact on levels must be lower-triangular:
L = chol(C1 * SigmaU * C1', 'lower');
B = C1 \ L;

% reduced-form MA(\infty) for \Delta y
H = 12;                                  % horizon quarterly
Phi = zeros(nvar,nvar,H+1);
Phi(:,:,1) = eye(nvar);
for h = 1:H
```

```
    S = zeros(nvar);
    for j = 1:min(h,p)
        S = S + A_cells{j} * Phi(:,:,h-j+1);
    end
    Phi(:,:,h+1) = S;
end

% structural IRFs for \Delta y
Theta = zeros(nvar,nvar,H+1);
for h = 0:H
    Theta(:,:,h+1) = Phi(:,:,h+1) * B;    % \Delta y responses
end
LevelTheta = cumsum(Theta, 3);

% for each shock k, GDP IRF = x_IRF + n_IRF
gdp_irf = squeeze(LevelTheta(1,:,:)+LevelTheta(2,:,:));

% extract series (columns = shocks ; col 1 = tech, col 2 = nontech):
tech = 1; ntech = 2;
prod_tech   = squeeze(LevelTheta(1,tech,:));
hours_tech  = squeeze(LevelTheta(2,tech,:));
prod_ntech  = squeeze(LevelTheta(1,ntech,:));
hours_ntech = squeeze(LevelTheta(2,ntech,:));
gdp_tech    = prod_tech  + hours_tech;
gdp_ntech   = prod_ntech + hours_ntech;
```

Step 3: Bootstrapped confidence bands. We construct percentile confidence bands for the level IRFs by an i.i.d. residual bootstrap, following the instructions in the previous Exercise of the PS. This involves re-tracking the VAR dynamics and re-imposing the long-run (Galí) identification in every replication. At first, after defining the effective sample for the VARs, we preallocate storage for each IRF. Later, for $K = 1000$ replications indexed by k we:

1. Resampled reduced-form residuals in order to have an i.i.d vector bootstrap.

2. Generated a bootstrap sample under the estimated VAR. This involved initializing `Yb` and, for each t, simulating the VAR using the estimated coefficients. Starting from the observed initial conditions $Y_1^{*(k)} = Y_1, \ldots, Y_p^{*(k)} = Y_p$, we iterate

$$Y_t^{*(k)} = \hat{c} + \sum_{j=1}^{p} \hat{A}_j Y_{t-j}^{*(k)} + u_t^{*(k)}, \qquad t = p+1, \ldots, T.$$

3. Re-estimated the RF through the stacked regression form and stored the intercept into `c_hat_b` and the coefficients into `A_cells_b`

4. Recomputed the long-run multiplier and identify structural shocks. Each time, we take the Cholesky of the long-run covariance to get a lower triangular matrix L_b

5. Recursively ran reduced-form MA multipliers Φ_h from the VAR lags and mapped it to structural IRFs.

Lastly, we cumulated across horizons to get level IRFs and we stored the responses that we needed from each shock:

$\texttt{prod\_b\_tech}(h) = \left[\text{LevelTheta\_b}_h\right]_{1,\,\text{tech}}$, $\texttt{hours\_b\_tech}(h) = \left[\text{LevelTheta\_b}_h\right]_{2,\,\text{tech}}$,

$\texttt{prod\_b\_ntech}(h) = \left[\text{LevelTheta\_b}_h\right]_{1,\,\text{nontech}}$, $\texttt{hours\_b\_ntech}(h) = \left[\text{LevelTheta\_b}_h\right]_{2,\,\text{nontech}}$

$\texttt{gdp\_b\_tech}(h) = \texttt{prod\_b\_tech}(h) + \texttt{hours\_b\_tech}(h)$, $\texttt{gdp\_b\_ntech}(h) = \ldots.$

For each stored objected, we form the empirical 2.5th and 95th percentiles across $k = 1, \ldots, K$ with $\alpha = 0.05$:

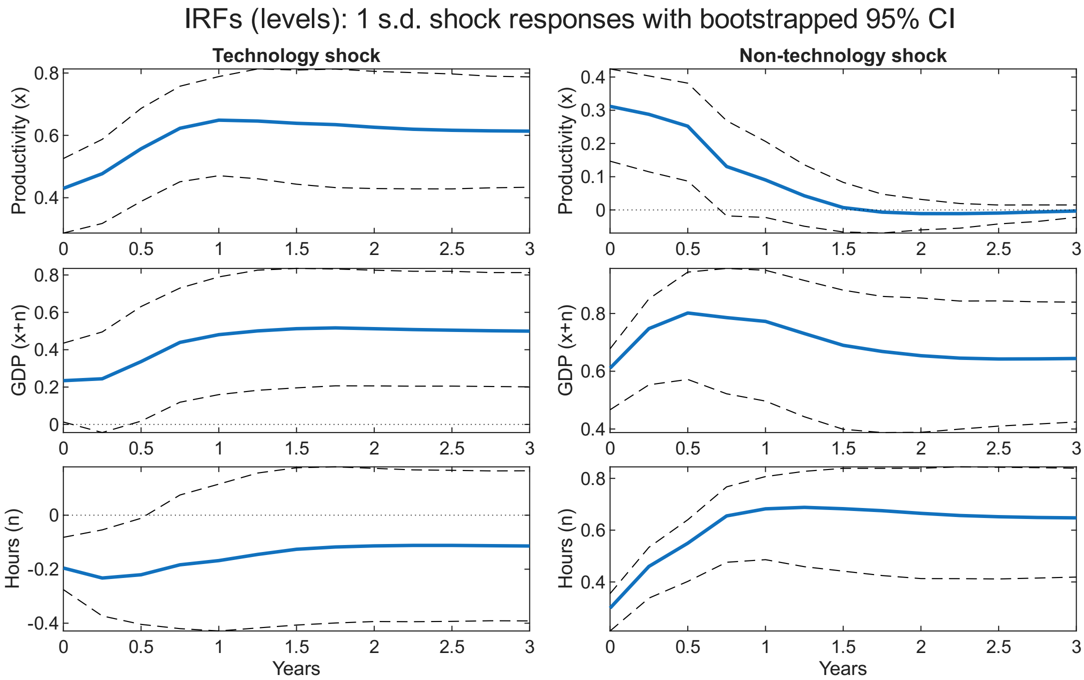
$$\mathrm{CI}_{95\%}(h) \;=\; \Big[Q_{0.025}\{\mathrm{IRF}^{*(k)}(h)\},\; Q_{0.975}\{\mathrm{IRF}^{*(k)}(h)\}\Big].$$

This yields pointwise 95% bootstrap percentile confidence bands. We finally provide the plot of our IRFs and bootstrapped CI, comparing it with Galì's graph.

```
    %% bootstrap CI
K = 1000;
[Tobs, nvar] = size(Y);
T_eff = Tobs - p;
I = eye(nvar);
boot_prod_tech   = zeros(K, H+1);
boot_gdp_tech    = zeros(K, H+1);
boot_hours_tech  = zeros(K, H+1);
boot_prod_ntech  = zeros(K, H+1);
boot_gdp_ntech   = zeros(K, H+1);
boot_hours_ntech = zeros(K, H+1);
for k = 1:K
    % resample residual rows with replacement
    idx = randi(T_eff, T_eff, 1);
    eps_tilde = resid_hat(idx, :);

    % simulate new series using \hat c, \hat A and resampled eps
    Yb = zeros(Tobs, nvar);
    Yb(1:p,:) = Y_init;
    for t = p+1:Tobs
        yhat = c_hat';
        for j = 1:p
            yhat = yhat + Yb(t-j,:) * A_cells{j}';
        end
        Yb(t,:) = yhat + eps_tilde(t-p,:);    % bootstrapped innovation
    end

    % re-estimate RF VAR(p) with constant on Yb
    [Tloc, ~] = size(Yb);
    T_eff_loc = Tloc - p;
    Yt_loc = Yb(p+1:end,:);
    X_loc = ones(T_eff_loc,1);
    for L = 1:p
        X_loc = [X_loc, Yb(p+1-L:end-L,:)];
    end
    B_loc = X_loc \ Yt_loc;
    U_loc = Yt_loc - X_loc*B_loc;
    SigmaU_b = (U_loc' * U_loc) / (T_eff_loc - size(X_loc,2));
    c_hat_b  = B_loc(1, :)';
    A_hat_loc  = B_loc(2:end, :)';
    A_cells_b  = mat2cell(A_hat_loc, nvar, nvar*ones(1,p));

    % long-run: C(1) = (I - A1 - ... - Ap)^{-1}
    A_sum_b = zeros(nvar);
    for j = 1:p, A_sum_b = A_sum_b + A_cells_b{j}; end
    C1_b = (I - A_sum_b) \ I;

    % identification: lower-triangular long-run (Gali)
    L_b = chol(C1_b * SigmaU_b * C1_b','lower');
    B_b = C1_b \ L_b;

    % RF MA multipliers
    Phi_b = zeros(nvar,nvar,H+1); Phi_b(:,:,1) = I;
    for h = 1:H
        S = zeros(nvar);
```

```
        for j = 1:min(h,p), S = S + A_cells_b{j} * Phi_b(:,:,h-j+1); end
        Phi_b(:,:,h+1) = S;
    end
    % structural IRFs for differences
    Theta_b = zeros(nvar,nvar,H+1);
    for h = 0:H, Theta_b(:,:,h+1) = Phi_b(:,:,h+1) * B_b; end
    LevelTheta_b = cumsum(Theta_b, 3);

    % store series needed for bands (columns = shocks: 1 tech, 2 non-tech)
    prod_b_tech   = squeeze(LevelTheta_b(1,1,:))';
    hours_b_tech  = squeeze(LevelTheta_b(2,1,:))';
    prod_b_ntech  = squeeze(LevelTheta_b(1,2,:))';
    hours_b_ntech = squeeze(LevelTheta_b(2,2,:))';

    boot_prod_tech(k,:)   = prod_b_tech;
    boot_hours_tech(k,:)  = hours_b_tech;
    boot_gdp_tech(k,:)    = prod_b_tech + hours_b_tech;
    boot_prod_ntech(k,:)  = prod_b_ntech;
    boot_hours_ntech(k,:) = hours_b_ntech;
    boot_gdp_ntech(k,:)   = prod_b_ntech + hours_b_ntech;
end
% percentile bands (95%)
pct = [2.5 97.5];
PT = prctile(boot_prod_tech,   pct);  prod_tech_lo   = PT(1,:)'; prod_tech_hi = PT(2,:)';
GT = prctile(boot_gdp_tech,    pct);  gdp_tech_lo    = GT(1,:)'; gdp_tech_hi = GT(2,:)';
HT = prctile(boot_hours_tech,  pct);  hours_tech_lo  = HT(1,:)'; hours_tech_hi = HT(2,:)';
PN = prctile(boot_prod_ntech,  pct);  prod_ntech_lo  = PN(1,:)'; prod_ntech_hi = PN(2,:)';
GN = prctile(boot_gdp_ntech,   pct);  gdp_ntech_lo   = GN(1,:)'; gdp_ntech_hi = GN(2,:)';
HN = prctile(boot_hours_ntech, pct);  hours_ntech_lo = HN(1,:)'; hours_ntech_hi = HN(2,:)';
```

□



VOL. 89 NO. 1 GALÍ: TECHNOLOGY, EMPLOYMENT, AND THE BUSINESS CYCLE 261

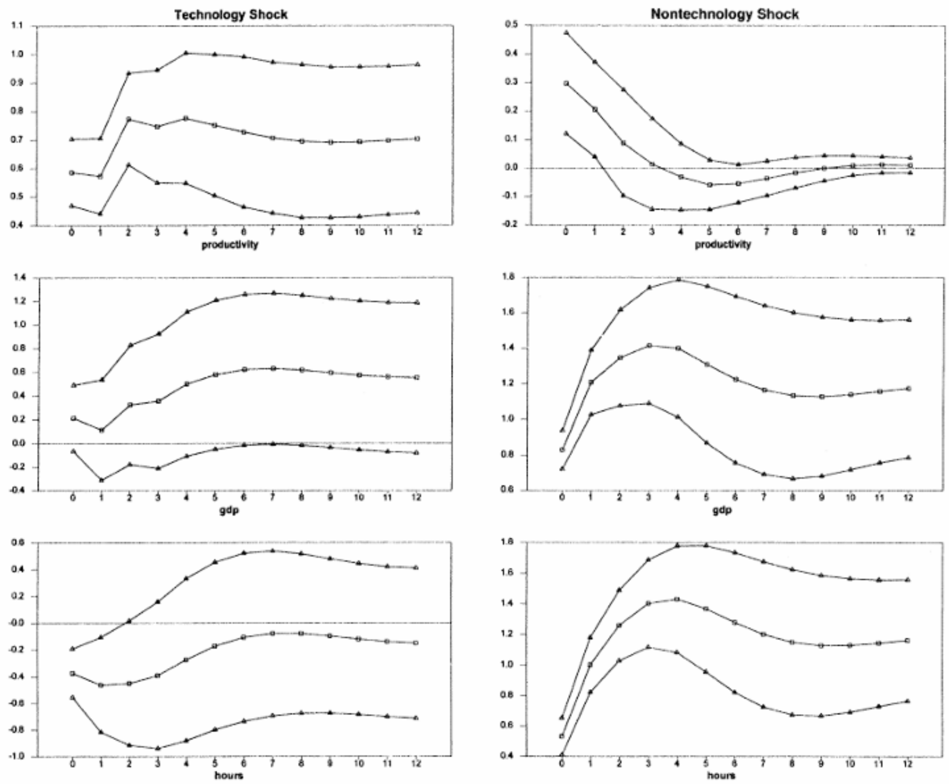


FIGURE 2. ESTIMATED IMPULSE RESPONSES FROM A BIVARIATE MODEL: U.S. DATA, FIRST-DIFFERENCED HOURS (POINT ESTIMATES AND ±2 STANDARD ERROR CONFIDENCE INTERVALS)