TECHNICAL ASSISTANCE REPORT

REPUBLIC OF MOZAMBIQUE

Walk through of MPC process using core-QPM

November 2025

Prepared By

Ørjan Robstad and Erlend Njølstad (short-term experts Norges bank)

Authoring Departments:

Monetary and Capital Markets Department

Nexus Reference: 18669492
Mission ID: 25MMN96
Funding Program ID: MCM_MOZ_2021_01

International Monetary Fund, IMF Publications
P.O. Box 92780, Washington, DC 20090, U.S.A.
T. +(1) 202.623.7430 • F. +(1) 202.623.7201
publications@IMF.org
IMF.org/pubs

Contents

Glossary

Admin	Administrative prices basket of goods in CPI
BM	Bank of Mozambique
BVAR	Bayesian vector auto-regressive model
Core-QPM	A core and simplified version of QPM
CPI	Consumer price index
CPI Core	CPI excluding fruits, vegetables and administered prices
DEE	Department of Economic Research
DMR	Markets and reserves Management Department
FG	Forward guidance
FPAS	Forecasting and policy analysis system
GDP	Gross domestic product
IT	Inflation targeting
MCM	Monetary and Capital Markets Department
MPC	Monetary policy committee of BM
MT	Mission team
NowMoz	Nowcasting system in DEE
QoQ	Quarterly growth rate (%)
QPM	Quarterly Projection Model
TA	Technical assistance
UIP	Uncovered interest parity
USD	US dollar
YoY	Year-on-year growth rate (%)

Preface

At the request of Bank of Mozambique (BM), a Monetary and Capital Markets (MCM) Department mission provided technical assistance from March 31 – April 11, 2025, to assist BM staff in building autonomy in the use of the new version of QPM (core-QPM) for forecasting and policy analysis purposes.

The mission team (MT) comprised of Ørjan Robstad and Erlend Njølstad, both from Norges Bank. The mission was undertaken as part of the Norwegian-funded, multi-year, central bank modernization program covering the main areas of central banking activities. The purpose of the mission was to make adjustments to the model and model output, to make it ready for practical use in the policy process. In addition, competence and autonomy on the use, maintenance, and further development of core-QPM for forecasting and policy analysis in the policy process was part of the mission.

Together with a small group of the Economic Research Department (DEE) staff, MT ran a hands-on mock forecasting and policy analysis process based on the data from the march policy round. It builds on several previous missions that involved developing, estimating and MATLAB training in core-QPM.

Executive Summary

Many of the models and tools that have been developed during earlier FPAS missions are actively used in the policy process. During the first on-site mission after the COVID-19 pandemic, however, it became clear that there was a need to enhance competence on the use of old-QPM for forecasting and policy analysis, including MATLAB scripts that handle the data feed, conditional forecasting, scenario analysis and reporting. Hence, in the follow-up mission in February 2023 and the virtual training session stretching from April to November 2023, DEE staff were introduced to some fundamental building blocks to enhance staff autonomy in the use of core macro models in the monetary policy process. As part of this effort, MT also worked hands-on with DEE staff to set up a more transparent policy model (henceforth core-QPM), based on the core of old-QPM that captures the transmission channels of the Mozambican economy. This also included adding features to meet the needs of BM and using the model to answer relevant policy questions. The model was also estimated using Mozambiquan data, improving the model's relevance for forecasting and analysis. For the past year BM staff have used the core-QPM model in parallel with the old-QPM.

The purpose of this mission was to add features and output to core-QPM so that BM is comfortable in using core-QPM as the main policy model. To this end, together with a small group of DEE staff, MT did a mock forecasting and policy analysis process. The exercises were based on the core-QPM framework that was developed during the mission in 2023 and 2024 and in subsequent virtual missions. In addition, DEE staff and the MT developed new features to improve the output and transparency of the model.

BM should move to a simpler and more transparent version of the QPM model based on the core-QPM model developed by MT in collaboration with DEE. The current version of QPM is unnecessarily complex, which makes it hard for DEE staff to reap the full benefits of having a macro model in the monetary policy process. Lack of transparency also makes it challenging to obtain the required autonomy to incorporate changes into the model, which is essential for the long-run relevance of the model as a policy tool. To ensure full autonomy in operating the core-QPM for forecasting and policy purposes, BM should transition to a simpler and more transparent core model.

The MT has provided BM with a user manual for the core-QPM version. A user manual was considered key to reduce dependence on single individuals in DEE and to increase organizational resilience against turnover. Although DEE staff overall possess the conceptual competence required, more time should be spent on taking ownership of the user manual and operational maintenance of the run-scripts and model. It is important that the tools surrounding the core-QPM are regularly maintained and updated when needed.

BM should enhance their analytical-capabilities using core-QPM and satellite models. More time should be spent on building analytical competence among BM staff to gain full advantage of having a semi-structural model for the Mozambiquan economy. This includes spending time on doing analytical exercises and adjustments in the current FPAS framework, and potentially expanding the model suite with satellite models which can shed light on topics and assumptions that are important for policy decisions. Potential satellite model extensions may be a model suite for the output gap and trends.

Next steps should start with a discussion between BM and the mission team around the key recommendations below. Going forward, the MT and BM should have a discussion of potential follow-up TA mission, topics and if they align with the recommendations in this TA report. Commitment from both BM and the MT is important to ensure further progress in modernizing the FPAS framework.

Recommendations

Table 1. Recommendations

Recommendations and Authority Responsible for Implementation	Priority	Timeframe 1/
Implementation of core-QPM and accompanying scripts as the main model for forecasting and policy analysis during the monetary policy process.	High	Near term
Take ownership and spend time on maintaining the FPAS-framework surrounding the core-QPM.	High	Near-term
Increase analytical capabilities using core-QPM and expanding with satellite models for the output gap and neutral real policy rate.	Medium	Near-term

1/ Near term: < 12 months; Medium term: 12 to 24 months.

Introduction

1. **BM is gradually modernizing its monetary policy framework based on inflation targeting (IT).** Currently, BM operates an FPAS and publishes a monetary policy report in conjunction with the monetary policy meetings. BM sets the policy interest rate, the MIMO-rate, with a view to stabilizing inflation. In addition, BM assesses risks surrounding the forecasts and provides alternative scenarios. The FPAS has gradually been expanded, and DEE is now actively using a nowcasting system (NowMoz) to produce near-term forecasts. In addition, a Bayesian vector auto-regression model (BVAR) is used to cross-check the medium-term forecasts. The medium-term forecasts and various policy analysis are produced using the old-QPM model. As noted by the TA mission in August 2022, there was still room for improvement in using the old-QPM to its full potential. During a follow-up mission in early 2023, the MT concluded that the complexity of the current version of old-QPM was an impediment to achieving staff autonomy in using the model for forecasting and policy analysis purposes. Examples of challenges with the old-QPM model are a complicated fiscal block and the lack of pre-filtering of gaps and trends prior to estimate the shock decomposition. Follow-up missions, both virtual and on-site, during 2023 and 2024 were mainly focused on developing a simplified version of the core-QPM and implementing the new model into the policy process at BM.

2. **This report builds on previous missions and their corresponding TA reports.** Various members of the MT have been involved in developing and implementing the core-QPM model. Different aspects of the model have been discussed and documented in previous TA reports, such as the TA report “Best-practice use of macro models in the Monetary Policy process” February 2024.

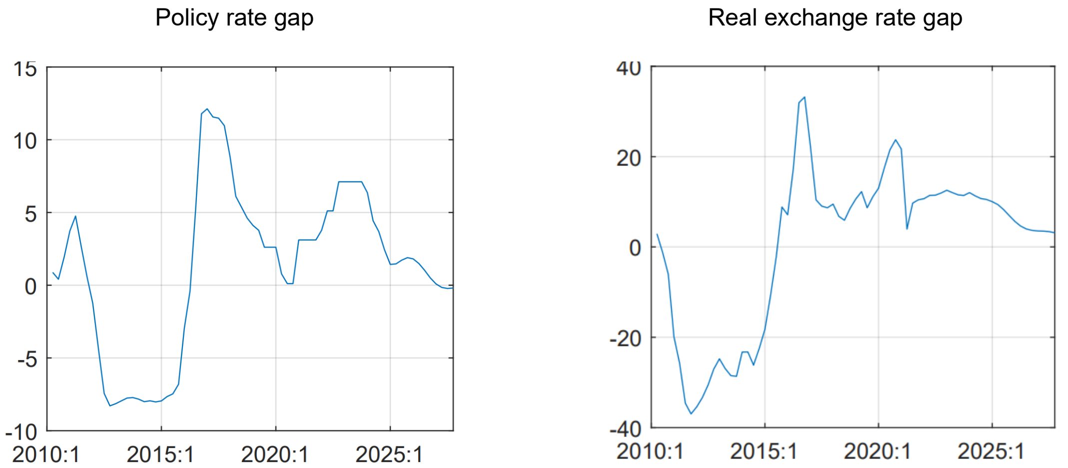
I. Forecasting and policy analysis with a simplified framework

3. **The purpose of this mission was to build further competence and autonomy on the use of core macro models for forecasting and policy analysis purposes.** During this mission, a group of DE staff and the MT did further enhancements on the structure of the core-QPM, ran shadow forecasting and policy process and made improvements to make output of the model more presentable for the MPC members. These steps have been important to help BM take well-informed decisions on the future use of core-QPM. Important differences between the core-QPM and old-QPM are reduced complexity in the core-QPM and increased transparency for the different steps involved in running, maintaining and using a macroeconomic model for forecasting and policy decisions. The DEE team consists of economists with experience in business cycle analysis, structural modelling and forecasting.

4. **Enhancements were performed by the DE staff together with the MT to increase their model understanding and capabilities.** Key assumptions in the core-QPM were discussed and adjusted. This included adjustments for the neutral level for the real exchange rate, the neutral level for the policy rate in Mozambique and the impulse from foreign variables. The trend in the real exchange rate was adjusted to a constant with a shift in 2015 when the exchange rate depreciated substantially. The neutral level for the real policy rate is slightly higher in the forecast horizon than its historical average. New assessment of trends in foreign variables ensures that the gaps are close to zero by the end of the forecast horizon. The shadow forecasting exercise covered most steps involved during a monetary policy process. The exercise was based on the same data and conditional assumption used during the latest policy process (the policy process leading up to the MPC decision in March 2025) and benchmark forecasts and alternative scenarios were produced. DEE staff familiarized themselves with different ways of doing scenario analysis inside a model framework. This included both alternative assumptions about foreign impulses and different policy responses from BM.

5. **During the mission, DEE staff and MT made improvements to the output of the model.** Important improvements during this mission included adding more forecasted variables to reporting, which makes them more accessible to model practitioners. Other improvements were adding different transformations for variables and adding graphs and tables in a presentable format. Output reports from the core-QPM will make the model more presentable internally for DEE staff and for MPC members during a monetary policy process. Figure 1 is an example of how important model variables are presented in the output report. The report includes both raw data and transformed data observable to the model, see Appendix 1-3 for further details and examples of reporting output. Expanding the reported output from the model will also increase the transparency of how the model works and key assumptions. Showing variables as gaps, as illustrated in figure 1, makes it easier for DEE staff to assess how different variables are compared to their steady state. Being able to do an assessment of gaps is important as the model interpretations of disturbances to the Mozambiquan economy is based on the gap variables, and not the level or growth rates.

Figure 1. The interest rate gap and real exchange rate gap in core-QPM



6. BM has performed an internal review of the forecasting properties based on out-of-sample forecasts from the core-QPM and the old-QPM. The review was performed by BM staff prior to the on-site mission, and thus the MT was not involved in the review. Based on BMs feedback to the MT, the review indicated that core-QPM was superior at forecasting key variables, but the MT has not been given the opportunity to see the results in detail.

II. Specific refinements, suggestions and comments

7. **The MT suggested areas for further capacity-building.** Even though considerable progress has been made in developing and using the FPAS, there is still a need for further competence-building to ensure that the required autonomy is in place to make necessary amendments to the preferred model and produce medium-term forecasts and various types of policy analysis. Given the complexity of the current QPM, the MT strongly recommends that DEE adapts the core-QPM that has been developed in 2023 and 2024 as the main framework for medium-term forecasting and policy analysis. The next step for the DEE should be to start using core-QPM as the main forecasting and policy model at BM.

8. **BM should take stronger ownership of the core-QPM and surrounding framework.** A user manual was considered key to reduce dependence on individuals in DEE and increase organizational resilience against turnover. It is strongly recommended that BM takes ownership of the user manual and operational maintenance of the entire FPAS framework. Maintenance and upkeep are required to gain the full advantages of having a semi-structural model. It is important that the tools surrounding the core-QPM are regularly maintained and updated when needed.

9. **BM should increase their analysis-capabilities using core-QPM and satellite models.** More time should be spent on building analytical competence among BM staff to gain full advantage of having a semi-structural model for the Mozambiquan economy. This includes spending time on doing analytical exercises and adjustments in the current FPAS framework, and potentially expanding the model suite with satellite models which can shed light on topics and assumptions that are important for policy decisions. Potential satellite models may be:

- A *model suite* for the output gap. The output gap is a key variable for assessing the state of the Mozambiquan economy. Extending the current FPAS framework with satellite models will allow BM to incorporate other sources of information in the judgment of the output gap. Suite of models has been discussed with BM in prior missions.

- Models for assessing trends, such as the trend in the real interest rate and real exchange rate, may be useful for enhancing the decomposition key variables in gap, trend and noise.

Next steps should be a discussion between the MT and BM about the road ahead. The central bank modernization program covering the main areas of central banking activities has made good progress during multiple on-site and virtual missions in the last years. Now is a good time to take stock and discuss potential commitment to future TA missions. Future TA missions may be linked to the recommendations highlighted in Table 1, but other topics such as a desk review of desk review of BM’s Economic Outlook and Inflation Forecasts publication may be suitable.

Annex material

An important output from this mission was new output reports from the core-QPM. Being transparent about a model's features increases transparency and credibility of the model. Three different output reports were made during this mission:

Annex 1: The main output report shows the historical data and forecast for important macroeconomic variables based on a baseline-run of the model. This includes the policy rate, inflation, exchange rate and real activity. Main variables are reported at different frequencies (QoQ, YoY etc.). Key variables are also reported with uncertainty bands based on empirical estimates of uncertainty. Forecasts are also reported in table format at different frequencies.

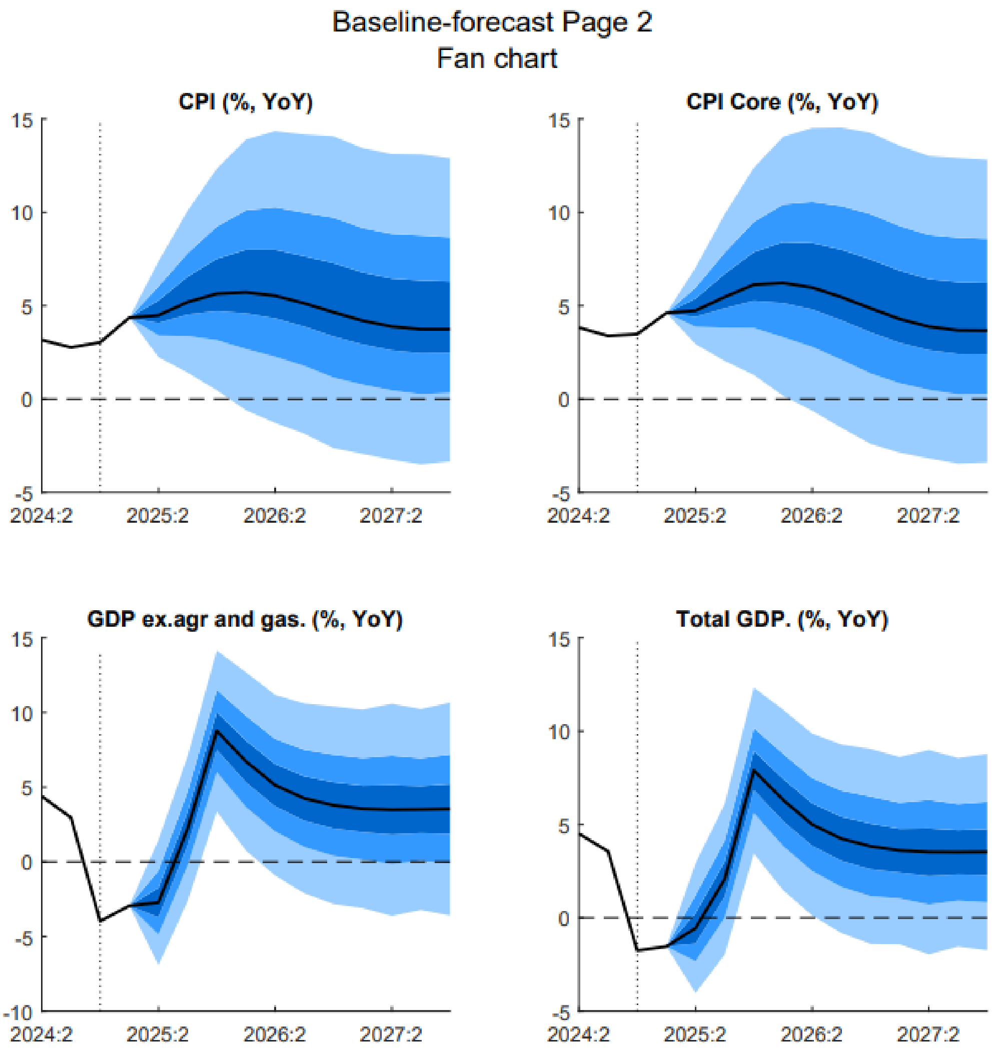
Annex 2: The scenario report highlights the difference between different model runs. This report illustrates how different assumptions implemented in different scenarios influence the outlook and potential policy implications. The scenario report includes additional figures, similar to the figures in the main report, which have been omitted for space considerations. The scenario report in the annex is an illustration of how scenario and policy analysis can be performed in the core-QPM framework and help policymakers take well-informed decisions about the policy stance and potential implications. The specific scenarios included are two examples where the exchange rate depreciates "Maligno" scenario, while the currency appreciates slightly in the "Benigno" scenario.

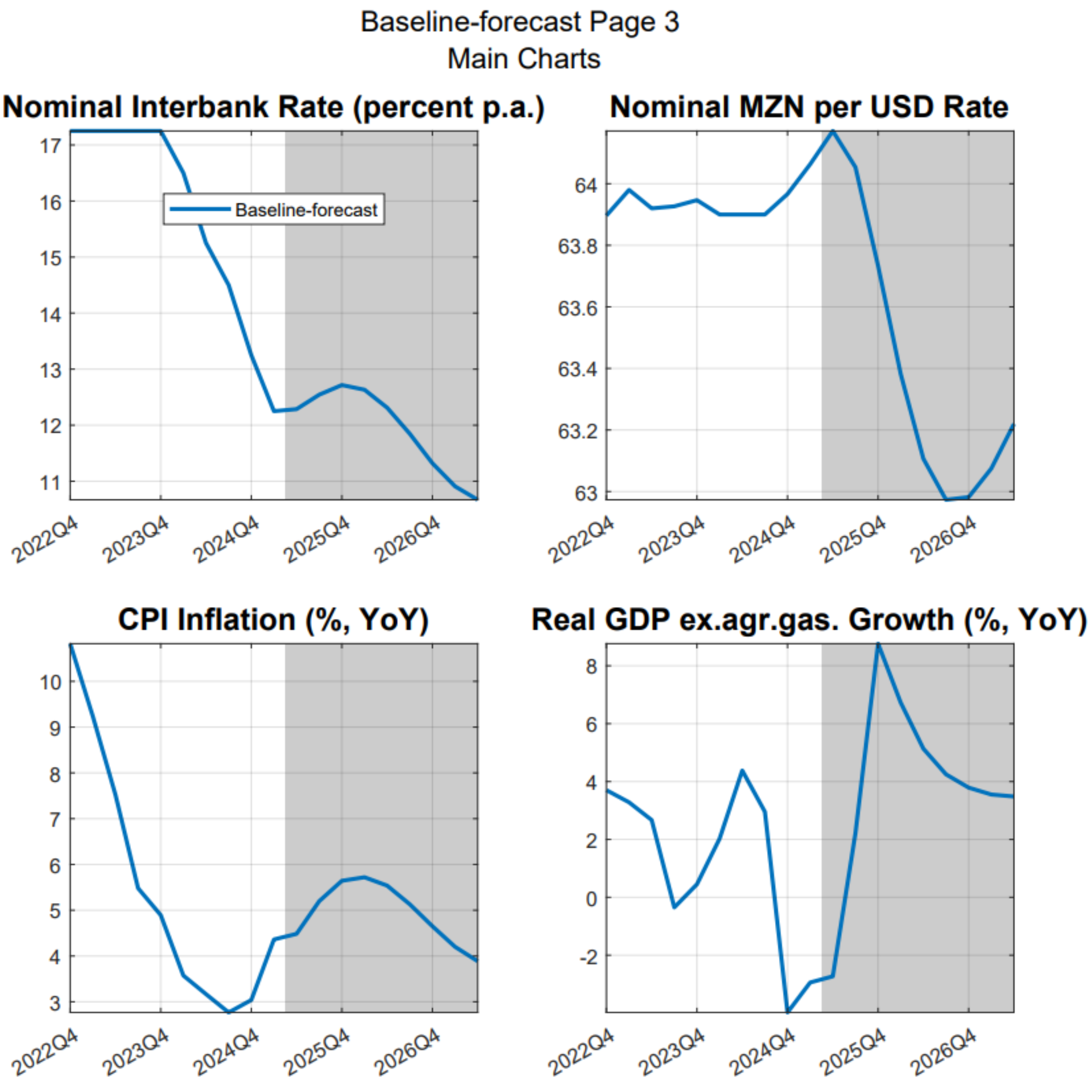
Annex 3: The filter report includes all variables which are included in the model and is more technical than the main and scenario report. The report is predominantly meant for DEE staff to understand the model and important assumptions. The filter report includes additional figures which have been omitted for space considerations.

Annex I. Main output from core-QPM model

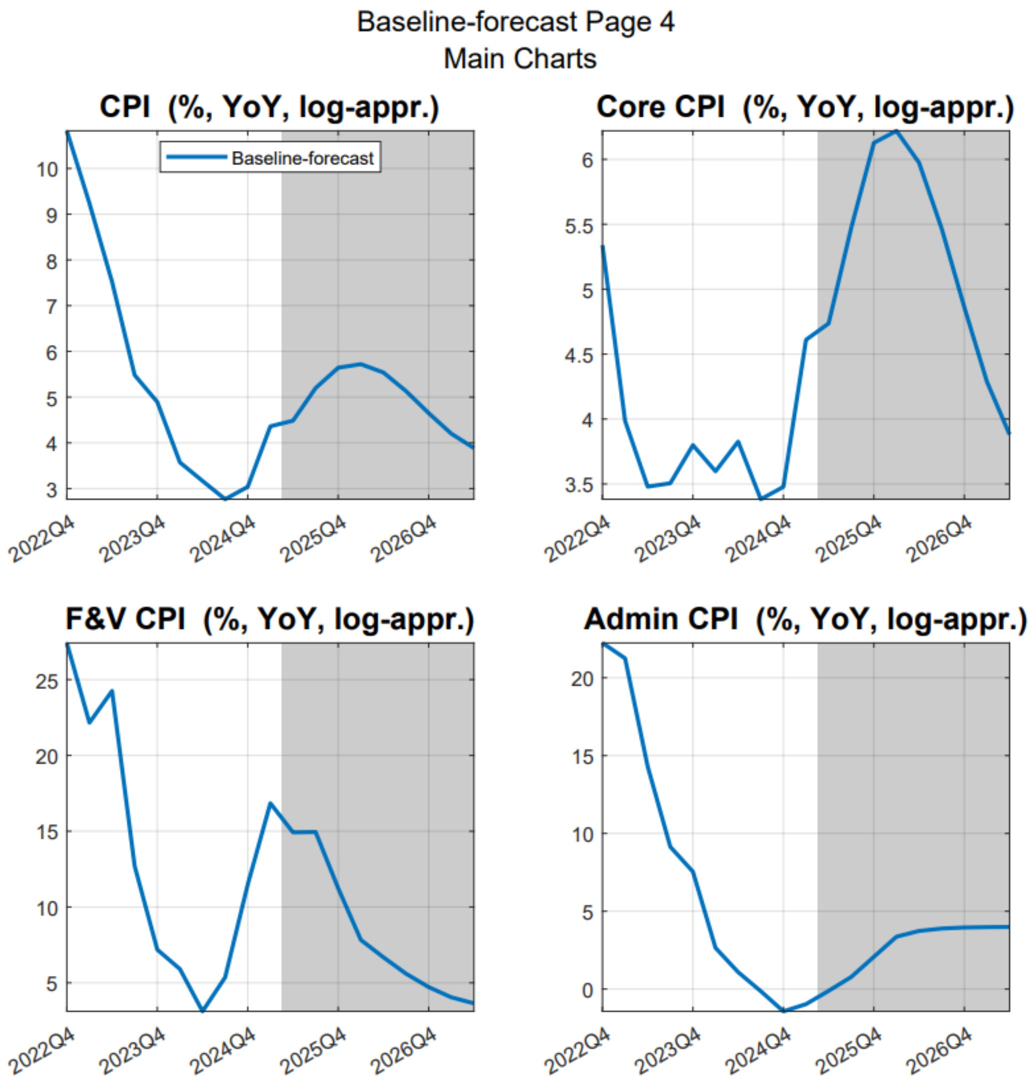
Report package for core-QPM model run: Baseline-forecast

April 10, 2025 13:58

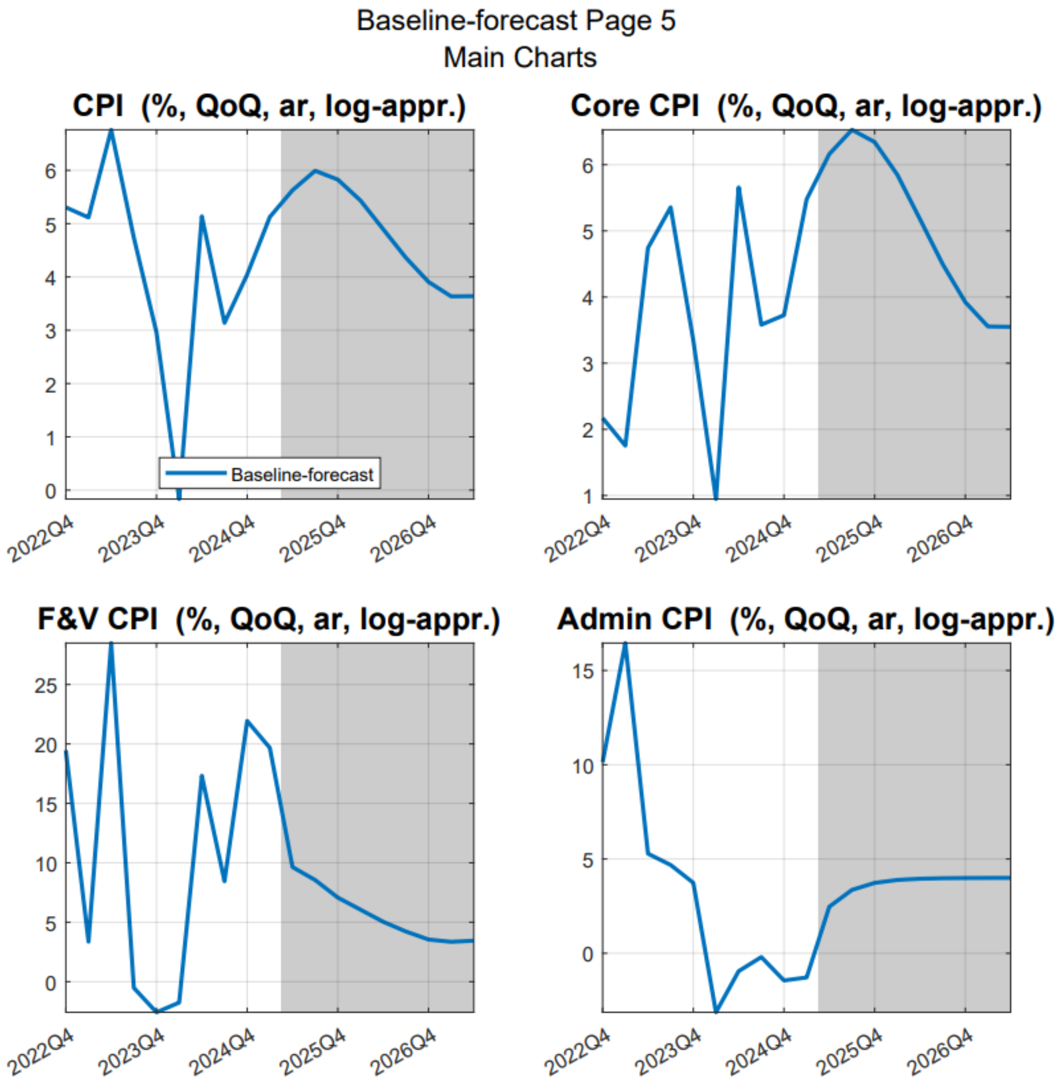

Baseline-forecast Page 2
Fan chart
CPI (%, YoY)
CPI Core (%, YoY)
GDP ex.agr and gas. (%, YoY)
Total GDP. (%, YoY)
15
10
5
0
-5
-10
2024:2
2025:2
2026:2
2027:2

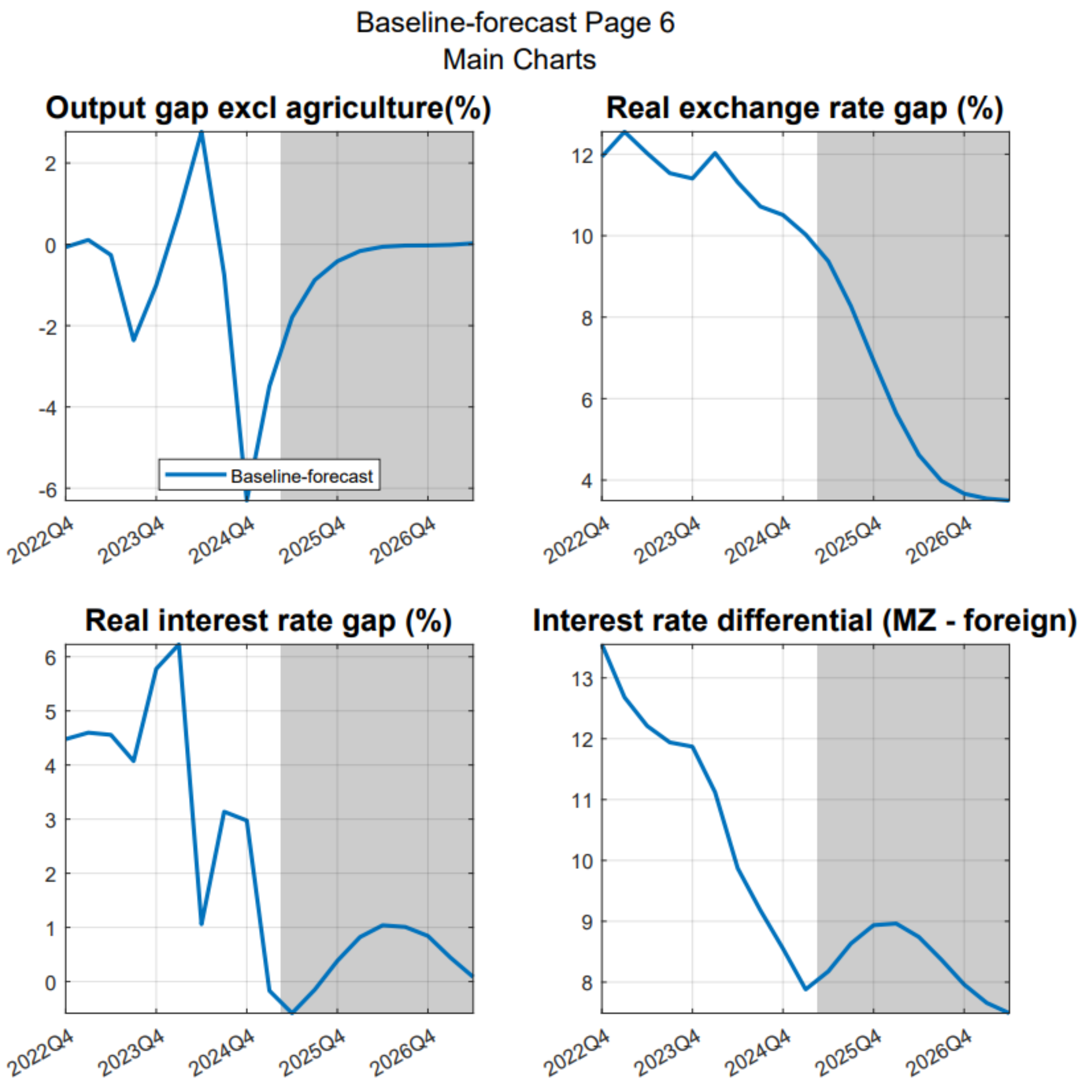


Baseline-forecast Page 3
Main Charts
Nominal Interbank Rate (percent p.a.)
Baseline-forecast
17
16
15
14
13
12
11
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4
Nominal MZN per USD Rate
64
63.8
63.6
63.4
63.2
63
CPI Inflation (%, YoY)
10
9
8
7
6
5
4
3
Real GDP ex.agr.gas. Growth (%, YoY)
8
6
4
2
0
-2



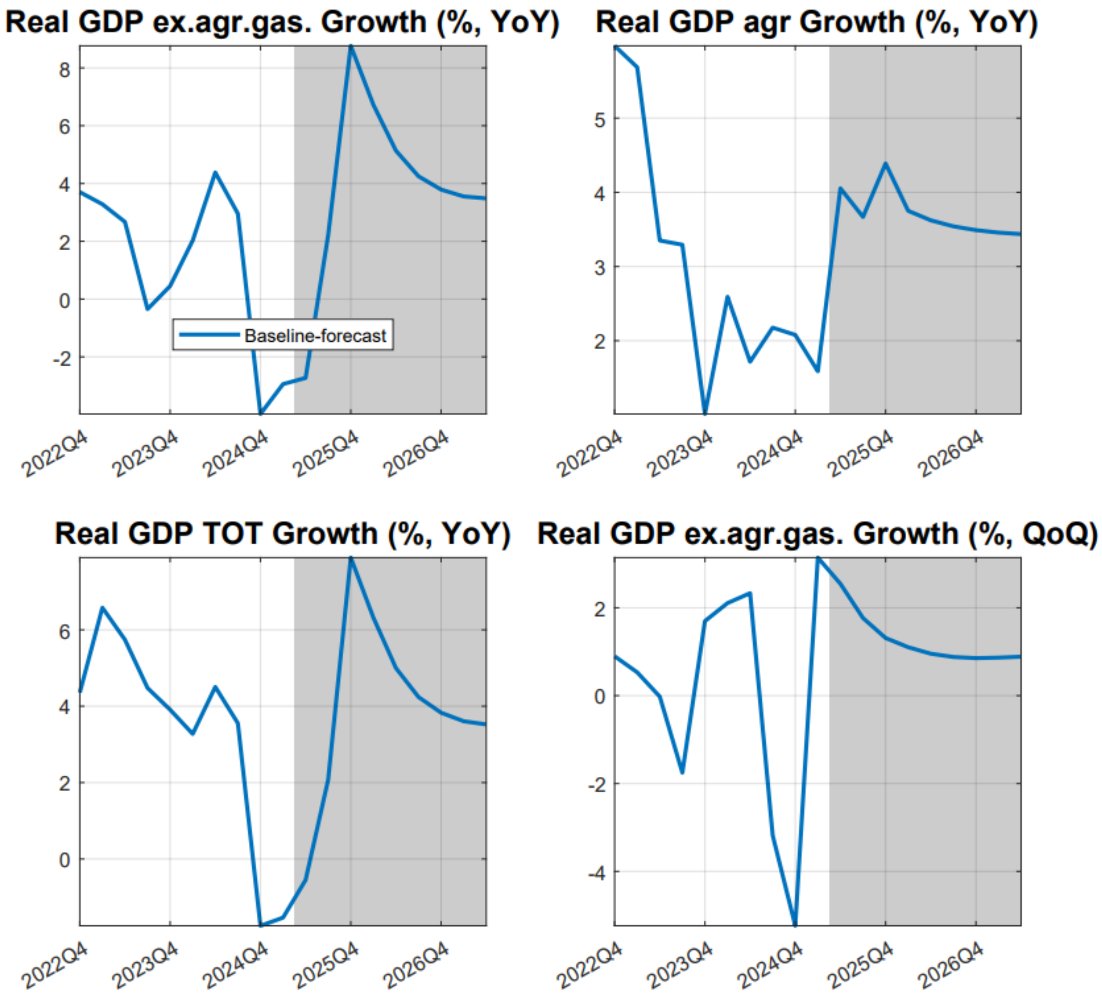
Baseline-forecast Page 4
Main Charts
CPI (%, YoY, log-appr.)
Baseline-forecast
10
9
8
7
6
5
4
3
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4
Core CPI (%, YoY, log-appr.)
6
5.5
5
4.5
4
3.5
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4
F&V CPI (%, YoY, log-appr.)
25
20
15
10
5
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4
Admin CPI (%, YoY, log-appr.)
20
15
10
5
0
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4


Baseline-forecast Page 5
Main Charts
CPI (%, QoQ, ar, log-appr.)
Core CPI (%, QoQ, ar, log-appr.)
F&V CPI (%, QoQ, ar, log-appr.)
Admin CPI (%, QoQ, ar, log-appr.)
Baseline-forecast
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4



Baseline-forecast Page 6
Main Charts
Output gap excl agriculture(%)
Real exchange rate gap (%)
Real interest rate gap (%)
Interest rate differential (MZ - foreign)
Baseline-forecast
2022Q4
2023Q4
2024Q4
2025Q4
2026Q4

Baseline-forecast Page 7
Main Charts



Model: **Baseline-forecast**
Date: 2025-04-10

	24Q2	24Q3	24Q4	25Q1	25Q2	25Q3	25Q4	26Q1	26Q2	26Q3	26Q4	27Q1
PRICES												
CPI Total	3.17	2.77	3.04	4.36	4.48	5.20	5.64	5.72	5.54	5.13	4.65	4.20
– CPI core	3.83	3.38	3.48	4.61	4.74	5.47	6.13	6.22	5.97	5.47	4.86	4.29
– CPI FV	3.12	5.36	11.49	16.84	14.92	14.95	11.23	7.83	6.68	5.60	4.72	4.04
– CPI admin	1.09	-0.14	-1.43	-0.97	-0.11	0.78	2.07	3.36	3.73	3.89	3.95	3.98
Policy Rate (percent)	15.25	14.50	13.25	12.25	12.29	12.54	12.72	12.63	12.31	11.85	11.32	10.91
MZN per USD	63.90	63.90	63.97	64.06	64.17	64.05	63.73	63.38	63.11	62.97	62.98	63.07
GDP												
GDP Total	4.51	3.55	-1.75	-1.54	-0.55	2.08	7.89	6.32	5.00	4.24	3.83	3.61
– GDP excl.gas.agr.	4.38	2.96	-3.98	-2.94	-2.73	2.22	8.76	6.73	5.13	4.25	3.79	3.55
– GDP agr	1.72	2.18	2.08	1.59	4.05	3.67	4.39	3.75	3.62	3.54	3.49	3.46
– GDP gas	26.11	20.21	13.37	3.44	9.30	-8.37	14.22	14.60	10.52	7.93	6.28	5.23
Policy rate (foreign)	5.38	5.32	4.70	4.37	4.11	3.91	3.78	3.67	3.57	3.48	3.36	3.25
CPI (foreign)	3.87	4.20	3.84	3.06	2.09	1.45	1.70	2.51	2.95	3.10	3.05	2.92
Real interest rate (gap)	1.06	3.14	2.97	-0.17	-0.58	-0.15	0.38	0.82	1.04	1.01	0.84	0.44

Model: **Baseline-forecast**

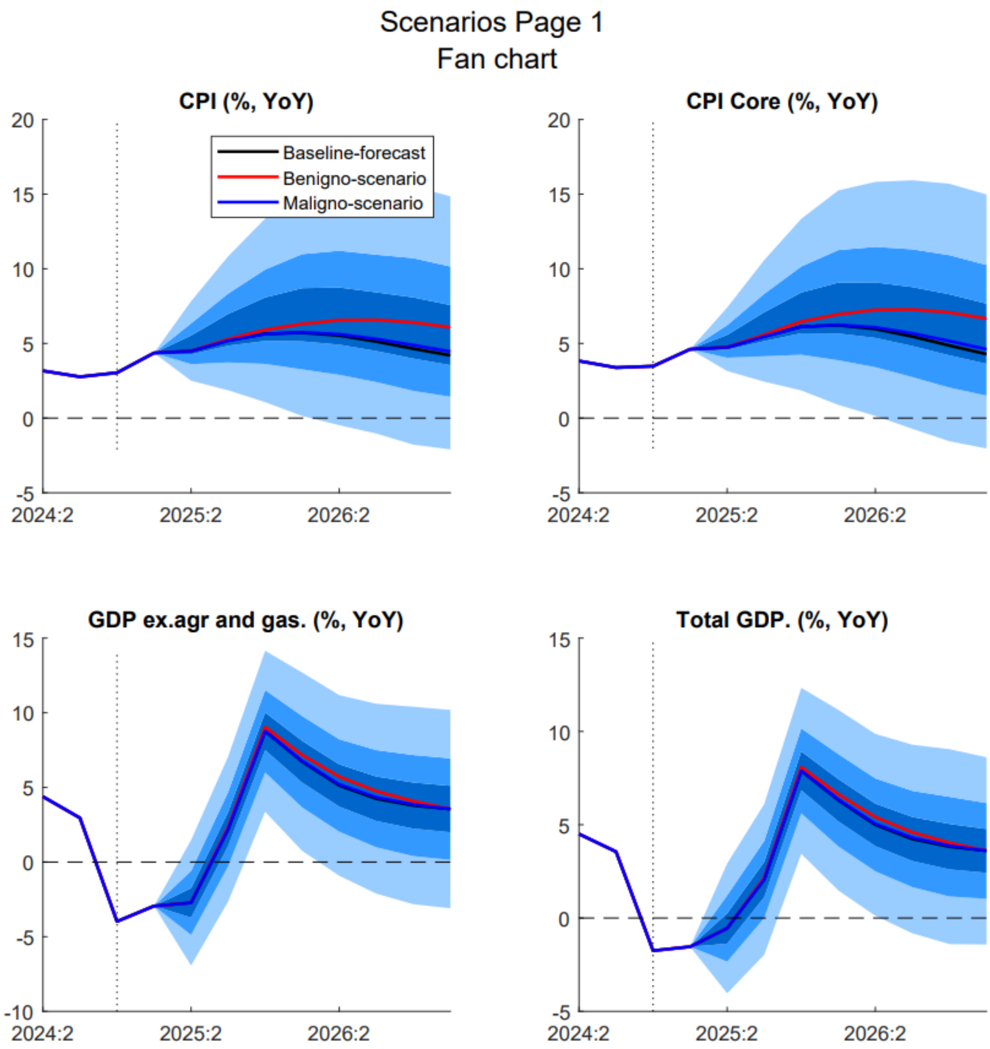
Date: 2025-04-10

	2024	2025	2026	2027
PRICES				
CPI Total	3.14	4.92	5.26	3.89
– CPI core	3.57	5.24	5.63	3.88
– CPI FV	6.47	14.49	6.21	3.69
– CPI admin	0.54	0.44	3.74	3.99
Policy Rate (percent)	14.88	12.45	12.03	10.70
MZN per USD	63.92	64.01	63.11	63.27
GDP				
GDP Total	2.40	1.97	4.85	3.54
– GDP excl.gas.agr.	1.35	1.33	4.97	3.52
– GDP agr	2.14	3.42	3.60	3.43
– GDP gas	23.39	4.65	9.83	4.45
Policy rate (foreign)	5.20	4.04	3.52	3.16
CPI (foreign)	3.47	2.08	2.91	2.71
Real interest rate (gap)	3.35	-0.13	0.93	0.05

Annex II. Scenario output report from core-QPM model (selected figures)

Report package for core-QPM model run: Scenarios

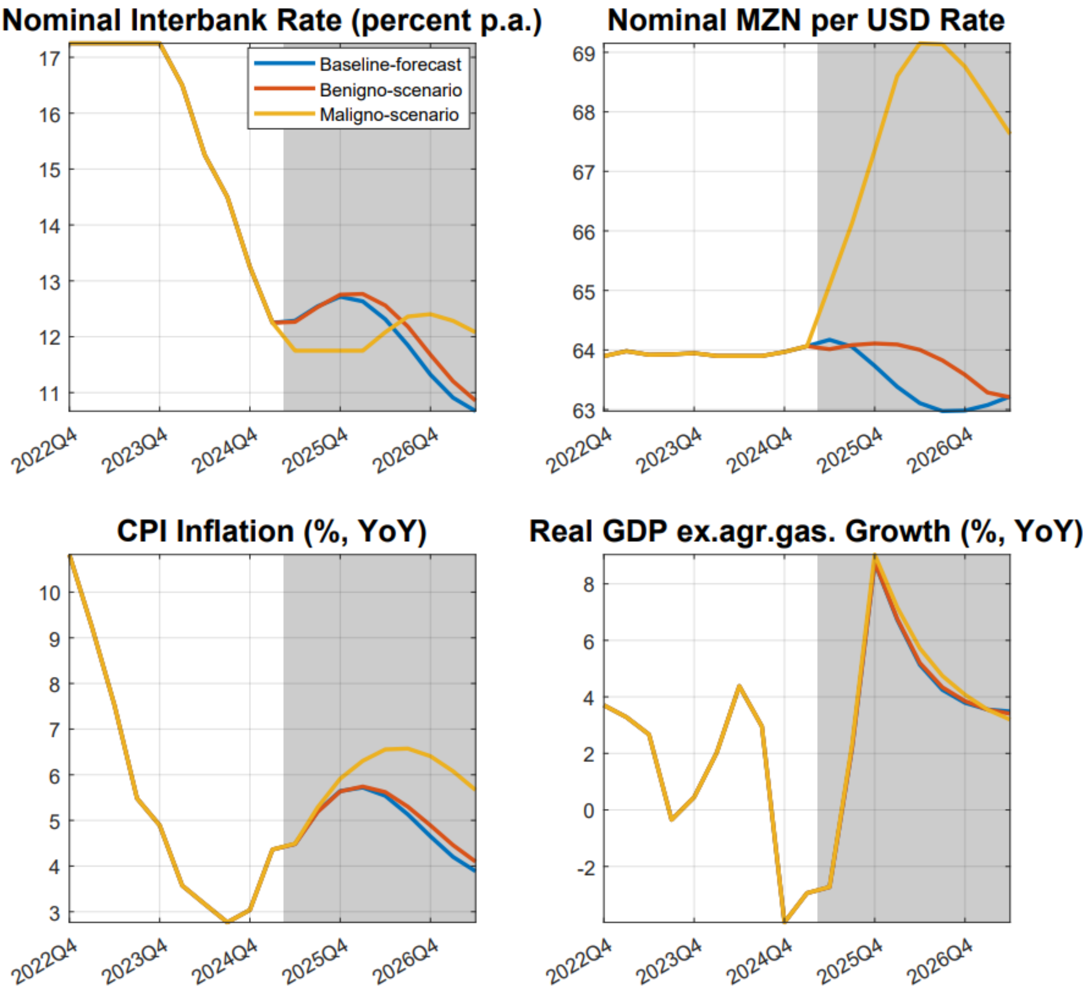
April 10, 2025 13:59



Scenarios Page 1
Fan chart
CPI (%, YoY)
Baseline-forecast
Benigno-scenario
Maligno-scenario
CPI Core (%, YoY)
GDP ex.agr and gas. (%, YoY)
Total GDP. (%, YoY)
2024:2
2025:2
2026:2

Scenarios Page 2

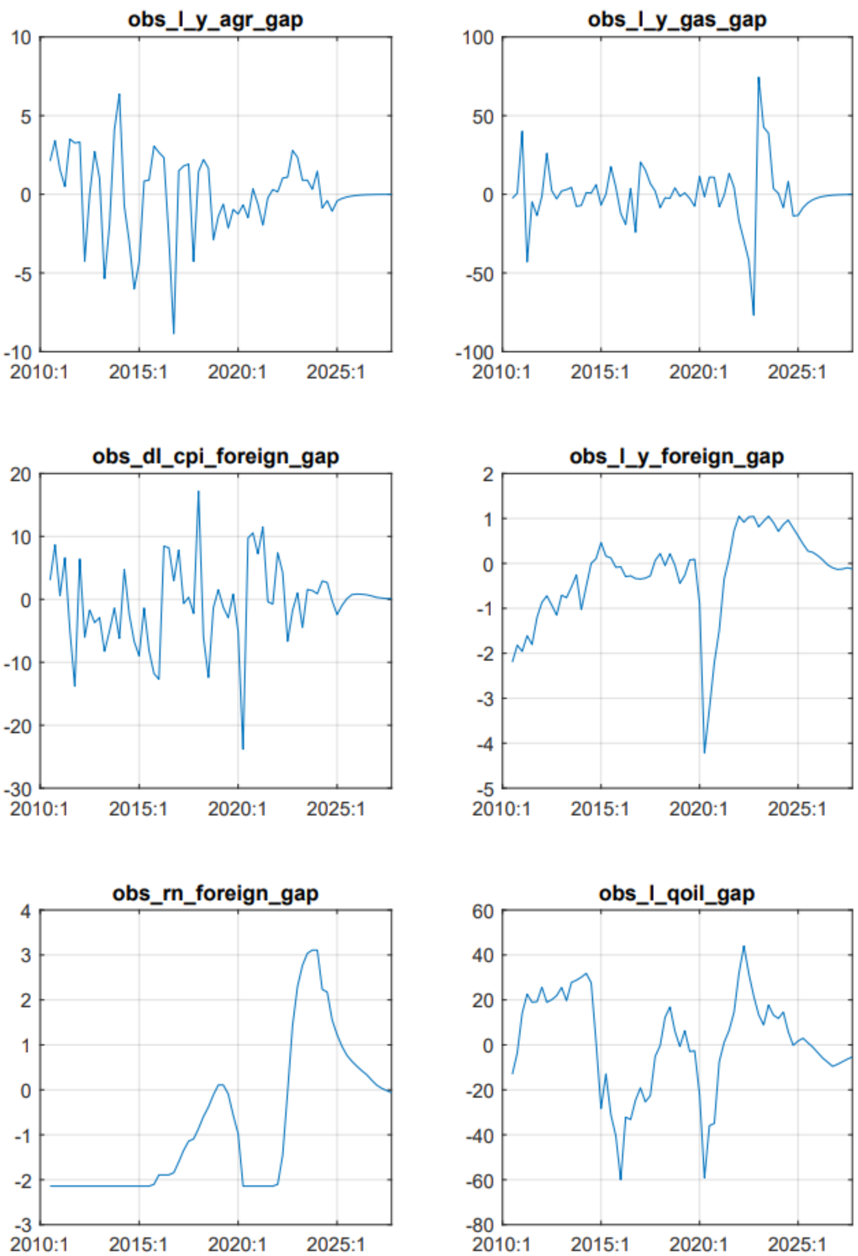
Main charts

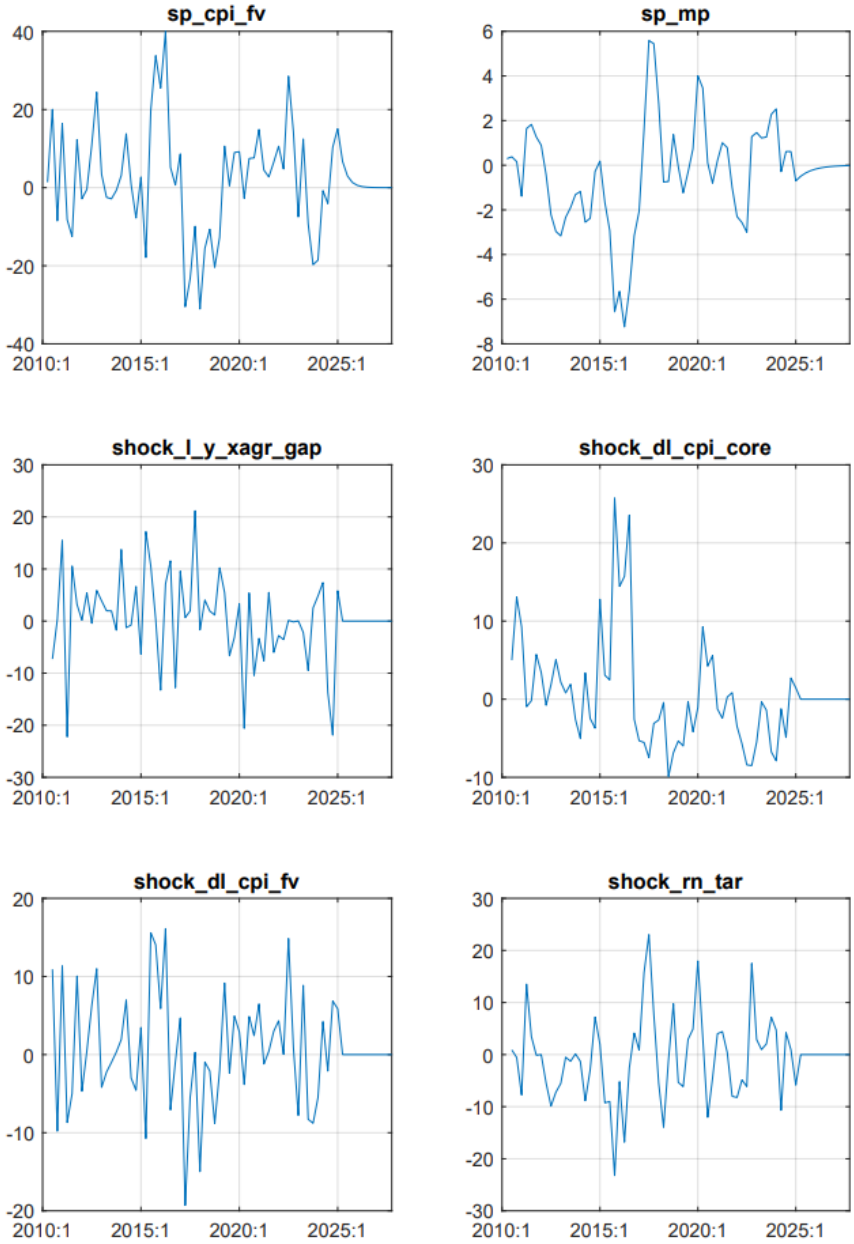


Annex III. Filter report from core-QPM model (selected figures)

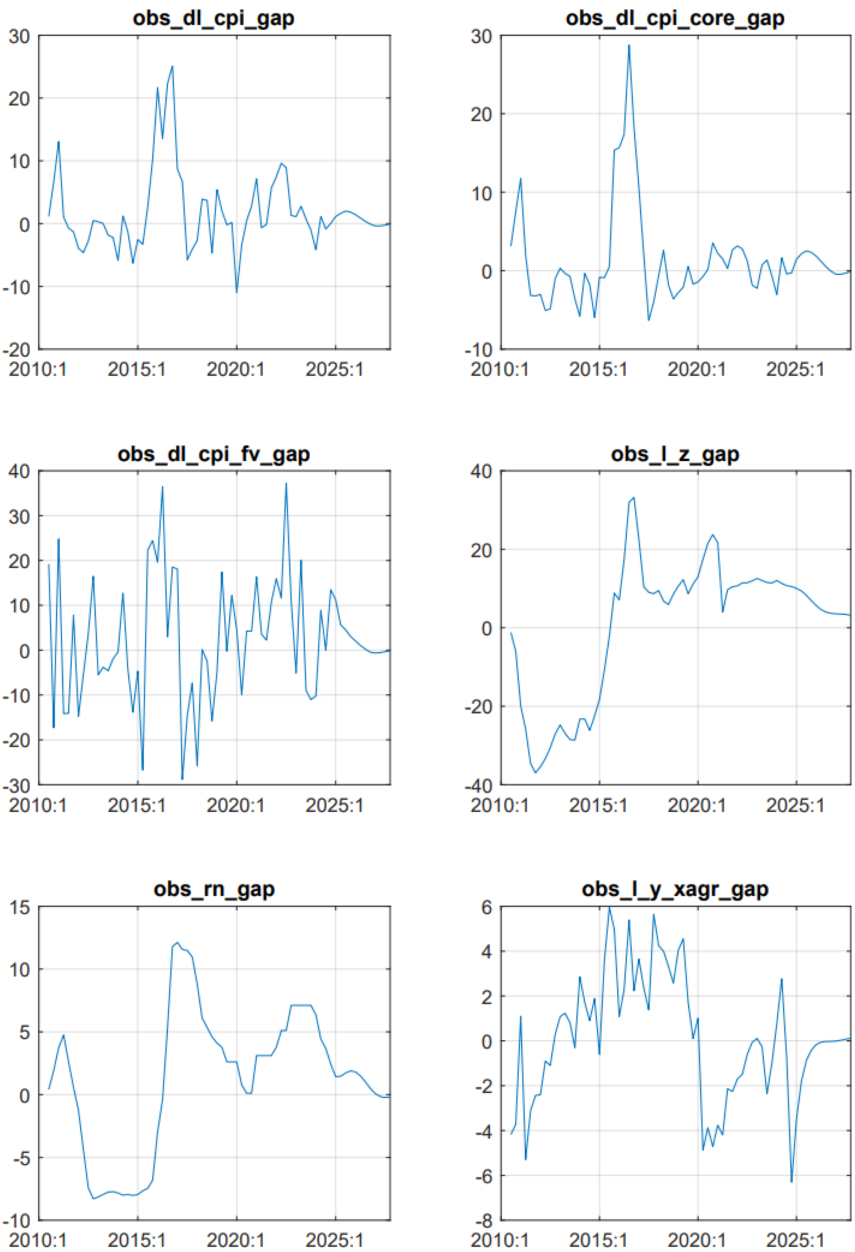
Filter Report core QPM

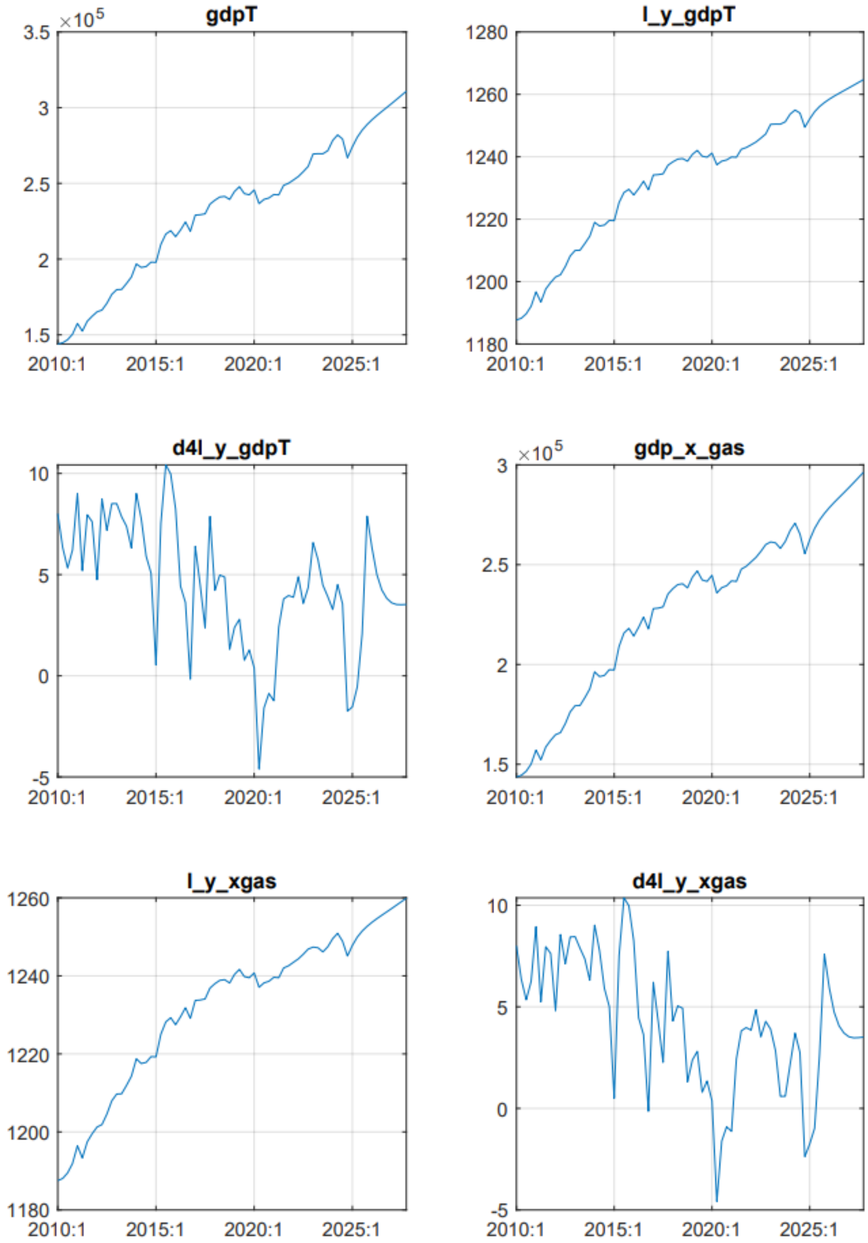
April 13, 2025

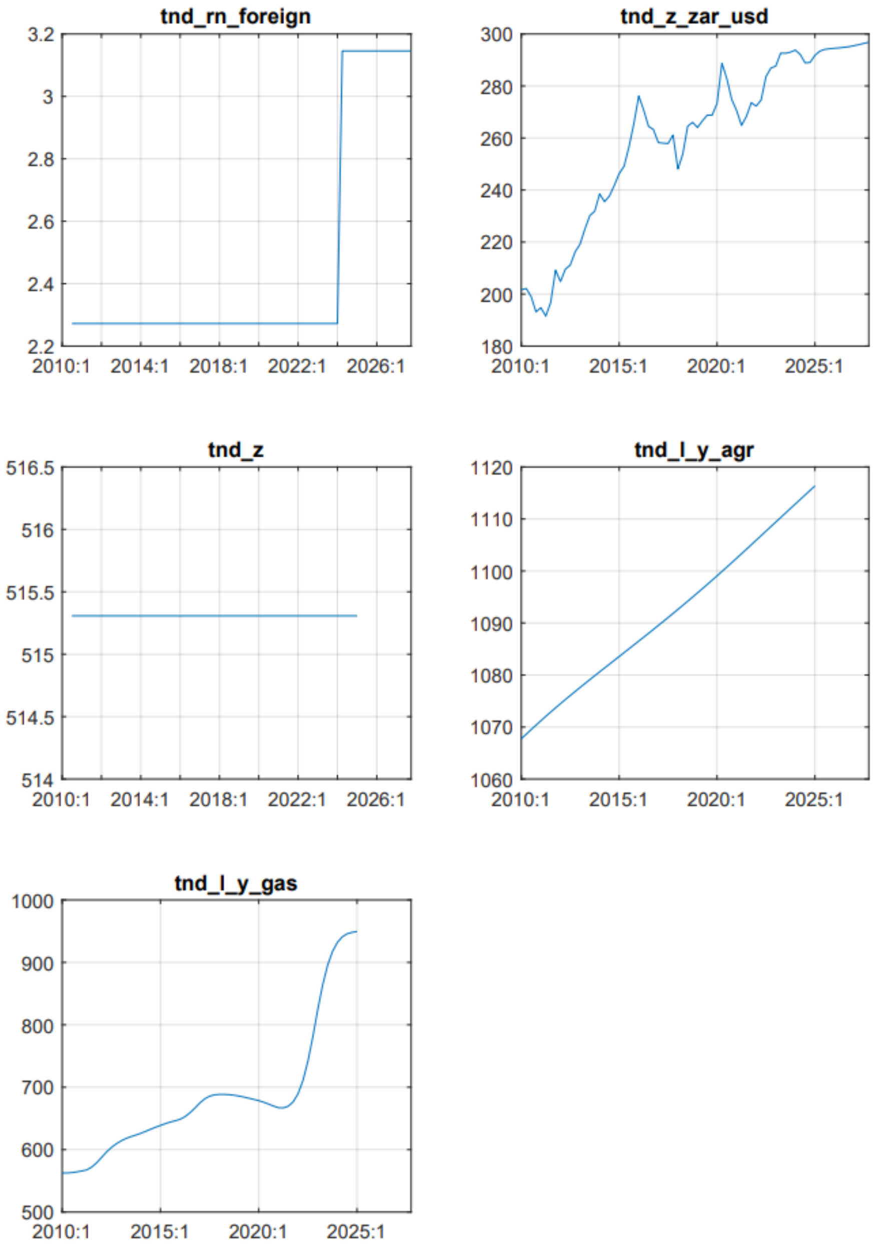

obs_l_y_agr_gap
obs_l_y_gas_gap
obs_dl_cpi_foreign_gap
obs_l_y_foreign_gap
obs_rn_foreign_gap
obs_l_qoil_gap
2010:1
2015:1
2020:1
2025:1



sp_cpi_fv
sp_mp
shock_l_y_xagr_gap
shock_dl_cpi_core
shock_dl_cpi_fv
shock_rn_tar
2010:1
2015:1
2020:1
2025:1


obs_dl_cpi_gap
obs_dl_cpi_core_gap
obs_dl_cpi_fv_gap
obs_l_z_gap
obs_rn_gap
obs_l_y_xagr_gap
2010:1
2015:1
2020:1
2025:1


gdpT
×10^5
3.5
3
2.5
2
1.5
2010:1
2015:1
2020:1
2025:1
l_y_gdpT
1280
1260
1240
1220
1200
1180
2010:1
2015:1
2020:1
2025:1
d4l_y_gdpT
10
5
0
-5
2010:1
2015:1
2020:1
2025:1
gdp_x_gas
×10^5
3
2.5
2
1.5
2010:1
2015:1
2020:1
2025:1
l_y_xgas
1260
1240
1220
1200
1180
2010:1
2015:1
2020:1
2025:1
d4l_y_xgas
10
5
0
-5
2010:1
2015:1
2020:1
2025:1


tnd_rn_foreign
3.2
3
2.8
2.6
2.4
2.2
2010:1
2014:1
2018:1
2022:1
2026:1
tnd_z_zar_usd
300
280
260
240
220
200
180
2010:1
2015:1
2020:1
2025:1
tnd_z
516.5
516
515.5
515
514.5
514
2010:1
2014:1
2018:1
2022:1
2026:1
tnd_l_y_agr
1120
1110
1100
1090
1080
1070
1060
2010:1
2015:1
2020:1
2025:1
tnd_l_y_gas
1000
900
800
700
600
500
2010:1
2015:1
2020:1
2025:1

Annex IV. User manual ("Cook book") (from mission in February 2024)

Introduction

The purpose of this manual is to go through the various steps and tasks that are typically involved in a forecasting and policy analysis process once the necessary conditioning assumptions like updated data and NTF etc. are available. More specifically, this is a detailed exposition of the steps involved in running the script ***run_mini_qpm_moz.m*** which parses the model into a model object, reads the data and makes necessary transformations, runs the Kalman filter to create initial values for all model variables, makes unconditional and conditional forecasts and corresponding forecast decompositions, among other things. Output from the various steps is reported in Excel-files and PDF's.

Preliminaries

The run-file "***run_mini_qpm_moz.m"*** is located in the folder **miniQPM**, which includes all the necessary input data, scrips, functions and toolboxes. To get started, open a new Matlab session and make sure you are located in the folder ***"…\miniQPM"***. No additional search paths are required.

Before starting the script, you need to make sure that all the necessary Data files, which are located in: ***"…\miniQPM\Data"***, are updated (**'db_monthly.csv', 'db_quarterly.csv', 'db_yearly.csv' and 'db_quarterly_external.csv'**). This includes both historical data and any forecasts you want to be part of the conditioning information set, i.e. near-term forecasts and medium-term forecasts for external (exogenous) variables.

You are now ready to open the '***run_mini_qpm.m'*** script in the MATLAB editor by double-clicking the **'*run_mini_qpm.m'*** in the ***'Current folder'*** window.

The 'run_mini_qpm.m' script

Main structure

The run-script is divided into different sections or steps. These can be summarized as follows:

1. Setting general options linked to sample range, forecast range etc.
2. Parsing and solving model
3. Running the 'load_data_miniQPM.m' script, which loads and transforms data, and creates the observable variables that are specified in the .mod-file.
4. Running Kalman filter to estimate starting values for all variables, including unobservables
5. Making unconditional forecasts
6. Making conditional forecasts
7. Making forecasts charts both unconditional and conditional forecasts, making forecast comparisons
8. Saving forecasts to .csv and .mat files
9. Estimating Bayesian AR(1)-models used as starting point to construct asymmetric fan charts
10. Making asymmetric fan charts
11. Making historical and forecast decompositions.

Section1: Setting useful date objects

It is often useful to set general options that will not change throughout the script at the beginning of the script. Below we show the whole section, which defines various date objects. We start by defining the start and end date (‘startDate’ and ‘endDate’) for the domestic data. Here the end-date should refer to the last date that will be treated as history, including the near-term forecasts, typically the quarter you are in at the time of the forecasting process. After setting the forecast horizon (‘fcastHor’), you can calculate some further date objects that are used later in the script.

```
%% Set options

% Set filtration range
startDate   =  qq(2010,3); % Start date for filtration [YYYY,Q]
endDate     = qq(2024,1); % End date for filtration[YYYY,Q]

fcastHor    = 20; % Number of quarters to forecast

% Generate date objects (do not change)
rng_hist        = startDate:endDate;
rng_filt        = rng_hist(1):rng_hist(end)+fcastHor;
rng_fcst        = rng_hist(end):rng_hist(end)+fcastHor;
rng_all         = rng_hist(1):rng_hist(end)+fcastHor;

starthist   = rng_hist(1);
endhist     = rng_hist(end);
startfcast  = rng_hist(end)+1;
endfcast    = startfcast+fcastHor;
fcastrange  = startfcast:endfcast;
```

Section 2: Parsing and solving the model

In the next step, we parse the model file (with extension .mod) to Matlab using the ‘model()’ function. Before solving the model, it is useful to check that the model has a well-defined steady state Given that this is a gap-model, i.e. all variables are interpreted as deviations from some constant or trend, the steady state of the model should be zero for all model variables.

```
%% Solve model
m = model('miniQPM_est_test.mod');


m = sstate(m);
chk = chksstate(m);
m = solve(m);
```

Section 3: loading the data transformation script

This step runs the script ‘load_data_miniqpm.m’ which loads the data and make necessary transformations. Alternatively, we could have included the whole content in this section but keeping it in a separate script makes the run-file simpler to read.

```
%% Load data
load_data_miniqpm;
```

We now turn to the details of the ‘load_data_miniqpm.m’ file. The first section of the script loads the required cvs-files. These files should be updated with the latest available data and near-term forecasts. They should also include medium-term assumptions used as conditioning information in the conditional forecasting exercise. This first and foremost concerns the external data. In the last part of this section some renaming of the national accounts take place. These variables have the extension _sa, indicating that they are seasonally adjusted. However, closer inspection indicates that this is not the case. Hence, the variables are renamed accordingly.

```
%% load data from exel
% dm = databank.fromCSV('data/db_monthly.csv');
dm = dbload('data\db_monthly.csv'); % monthly data from csv file
dq = dbload('data\db_quarterly.csv');
dy = dbload('data\db_yearly.csv');
dq_ext = dbload('data\db_quarterly_external.csv');
% dq_ext_prev = dbload('data\db_quarterly_external_prev.csv');

dq = dbmerge(dq,dq_ext);
dq.gdp = dq.gdpT_sa;
dq.gdp_agr = dq.gdp_agr_sa;
dq.gdp_gas = dq.gdp_gas_sa;
dq.gdp_xagr = dq.gdp_xagrxgas_sa;
```

In the next section, ‘Data processing’, data are seasonally adjusted where appropriate, and monthly and yearly data are converted into quarterly data. Subsequently, trends and historical averages are calculated, which are finally used to construct the gap-variables consistent with the model variables.

Seasonal adjustments are made using the x12() function in IRIS, which draws on the X13-ARIMA-SEATS filter of the U.S. Census Bureau. Of the monthly data, only the CPI indices are seasonally adjusted.

```
%% data processing

% This is the code you should use, so need to comment this in.

dm.cpi              = x12(dm.cpi_su);
dm.cpi_core         = x12(dm.cpi_core_su);
dm.cpi_fv           = x12(dm.cpi_fv_su);
```

The next step is to make sure that all variables have the same frequency, which is quarterly. In IRIS, this can be done using the 'convert()' function. For monthly data, like the various CPI indices, interest rates and exchange rates, the quarterly series are constructed as 3-month averages within the respective quarters (default option going from high to low frequency). For annual series, a quadratic interpolation approach is used which preserves the low frequency mean (the option method=quadsum). To see the options available type: `help tseries/convert`

```
% Convert monthly and annual data series to quarterly using the 'convert'
% function in IRIS
db_model.rn = convert(dm.ir_ib,'q');
db_model.FI = convert(dy.budg_bal_prim/dy.gdp_nom,'q','method','quadsum')*-100;
dq.mzn_usd = convert(dm.mzn_usd,'q');
dq.l_mzn_usd = 100*log(convert(dm.mzn_usd_cb,'q'));
db_model.dl_cpi = diff(log(convert(dm.cpi,'q')))*400;
db_model.l_cpi = (log(convert(dm.cpi,'q')))*100;
db_model.dl_cpi_core = diff(log(convert(dm.cpi_core,'q')))*400;
db_model.dl_cpi_fv = diff(log(convert(dm.cpi_fv,'q')))*400;
db_model.l_s = 100*log(convert(dm.mzn_usd,'q'));
```

In order to be compatible with the model variables, all observables need to be detrended or demeaned. Here we deviate from the approach taken in QPM, where the detrending is an integral part of the model. In the miniQPM, the detrending is done outside the model.

Take GDP in agriculture as an example. In the QPM, a (partial) univariate filter is assumed for most non-stationary variables. For GDP in agricultural (y_t), which in principle is assumed to be an exogenous variable this would typically look something like this:

$$\Delta y_t = \Delta\hat{y}_t + \Delta\bar{y}_t$$

$$\Delta\bar{y}_t = \rho\Delta\bar{y}_{t-1} + \epsilon_t$$

$$\Delta\hat{y}_t = \delta\Delta\hat{y}_{t-1} + \varepsilon_t$$

For this to be operational, we would need to know the parameters $\rho, \delta, \sigma_\epsilon, \sigma_\varepsilon$, where the latter two refer to the standard deviations of the stochastic shocks ϵ and ε respectively. It is demanding to estimate these parameters based solely on observations on y_t. Both the gap $\hat{y}_t$ and the trend $\bar{y}_t$ are unobservable. In miniQPM, we instead calculate the gap, $\hat{y}_t$, directly as $\hat{y}_t = y_t - \bar{y}_t$ where the trend,

$\bar{y}_t$, is calculated using an hp-filter. Hence for non-stationary variables we detrend the variable using the hp-filter, whereas for stationary variables we simply subtract the mean in order to calculate the gap-variable. Note that for some variables we allow for a break in the mean over the sample period. This is true for CPI where we use the sample average up until 2018, whereas we assume an inflation target of 4 percent in the period thereafter (i.e. the mid-point of interval 2 to 6).

Examples of detrending for the various GDP variables:

```
%trend in GDP assumed to be hpfilter
db_model.l_gdp_xagrxgas    = 100*log(dq.gdp_xagr);
db_model.l_gdp_agr         = 100*log(dq.gdp_agr);
db_model.l_gdp_gas         = 100*log(dq.gdp_gas);

tnd_l_y_xagrxgas = hpf(db_model.l_gdp_xagrxgas,'lambda',10000);
tnd_l_y_agr = hpf(db_model.l_gdp_agr,'lambda',10000);
tnd_l_y_gas = hpf(db_model.l_gdp_gas,'lambda',10000);
tnd_l_y_gas(qq(2023,3):endDate) = tnd_l_y_gas(qq(2023,3):endDate) +45;

db_model.l_y_xagrxgas_gap = db_model.l_gdp_xagrxgas-tnd_l_y_xagrxgas;
db_model.l_y_agr_gap = db_model.l_gdp_agr-tnd_l_y_agr;
db_model.l_y_gas_gap = db_model.l_gdp_gas-tnd_l_y_gas;
```

And for the CPI series:

```
% trend in cpi assumed to be average
before = qq(2010,3):qq(2018,1);
new    = qq(2018,2):endDate;
tnd_cpi = tseries(startDate,ones(size(startDate:endDate,2),1));
tnd_cpi(before) = mean(db_model.dl_cpi(before));
tnd_cpi(new) = 4;
db_model.dl_cpi_gap    = db_model.dl_cpi - tnd_cpi;
```

Next, the various trends are saved to be used for forecasting:

```
% Save trends in tnd-object, used for forecasting
tnd.tnd_FI          = tnd_FI;
tnd.tnd_cpi         = tnd_cpi;
tnd.tnd_cpi_core    = tnd_cpi_core;
tnd.tnd_cpi_foreign = tnd_cpi_foreign;
tnd.tnd_cpi_fv      = tnd_cpi_fv;
tnd.tnd_rn          = tnd_rn;
tnd.tnd_rn_foreign  = tnd_rn_foreign;
tnd.tnd_z           = tnd_z;
tnd.tnd_l_y_agr     = tnd_l_y_agr;
tnd.tnd_l_y_gas     = tnd_l_y_gas;
% For GDP we assume a constant trend in the forecasts, for historical data
% the trend is based on a HP-filter
tnd.tnd_l_y_xagrxgas    = tnd_l_y_xagrxgas;


save('data_miniQPM')
```

The final step is to create the observables as they are defined in the model file based on the gap variables that are created earlier in the section (found in the db_model) and store them in the structure 'db_obs'. All observables are plotted in a separate figure, and we strongly recommend that you take a close look at the charts to see if the gap variables look reasonable.

Section 4: Run filter

In this step, we run the Kalman filter to produce starting values for all the unobserved variables in the model, which together with the obsvervables allow us to produce forecasts.

```
%% Run filter

[m_filt, db_filt] = filter(m,db_obs,rng_filt);
```

To run the Kalman filter, we use the filter() function where the input arguments are the model object (m), the observable data (which are in db_obs) and the forecasting range which is set at the beginning of the script. The object "db_filt" contains the filtered data (including the the initial values and smoothed shocks), which will be used to generate forecasts. The output also includes "m_filt", which is an updated version of the input model (it will only change if `Relative=` is true and/or updates if `OutOfLik=' is non-empty – which is not the case here). To learn more about the 'filter()' function, you can type 'doc model/filter' in the Matlab command window.

Section 5: Unconditional forecasts

To make forecasts we apply the function 'jforecast()'. This is a general function which allows you to do both unconditional and conditional forecasting. What we term 'unconditional forecasts' here, is not the pure unconditional model forecasts. We already condition on the short-term forecasts and medium-term forecasts for external variables.

```
%% Produce unconditional forecast

% Mean forecasts
f        = jforecast(m, db_filt, fcastrange,'anticipate=',false);
uncert = dbextend(db_filt.std, f.std);
db_fcast = dbextend(db_filt.mean, f.mean);

% Transform variables
[db_fcast] = reporting_miniQPM(db_fcast,db_model,m,tnd,uncert,fcastrange,endDate);
```

The input arguments used are the model object, the initial values and smoothed shocks needed to map the future external variables from 'db_filt' (on which we condition) and the forecast range. In addition, we use the option 'anticipate=' to instruct the function to use unanticipated shocks (i.e. 'anticipate=', false). The two next lines adds the forecasted standard deviations and mean projections to the historical smoothed values. The standard deviations included in the 'uncert' structure can be used to construct fan charts. The db_fcast structure includes the mean projections of all model variables.

The last line calls the reporting_miniQPM script, which transforms the model variables back to levels and growth rates as observed in the data. For each gap, we add the estimated hp-trend or sample average, both for historical data and the projections and calculate the desired reporting variables.

```
% GDP xagrxgas
tnd_gdp = 0.85;
db3.trnd_growth = tnd.tnd_l_y_xagrxgas.diff(-1); %history from hpfilter
db4.trnd_growth = Series(fcastrange,ones(size(fcastrange,2),1)*tnd_gdp);
db5 = dbextend(db3,db4);
db_fcast.dl_y_xagrxgas = db5.trnd_growth + db_fcast.l_y_xagrxgas_gap.diff(-1);
db_fcast.l_y_xagrxgas=cumsum(db_fcast.dl_y_xagrxgas);
db_fcast.d4l_y_xagrxgas = db_fcast.l_y_xagrxgas.diff(-4);
```

The example above, which is taken from 'report_miniQMP' script starts by defining the potential GDP growth over the projection horizon. Here it is simply set to a fixed growth rate and you need to take a stand on this number. In the next line, historical quarterly growth rates for the potential based on the hp-filter, are stored in the from the Kalman filter are stored in db3. Next, potential growth rates equal to 'tnd_gdp' for the whole forecast horizon is stacked in an data array db4. The historical and future potential growth rates are joined as one time series array (db5). Given the trend growth, overall growth rates can be calculated as the sum of trend growth and growth in the corresponding gaps.

The last section of the report_miniQPM script calculates confidence bands that can be used to plot fan charts.

Section 6: Conditional forecasts

As mentioned above, even what we have termed unconditional forecasts above are in fact conditional forecasts. The conditioning information is already included in the csv data files. This includes the near term forecasts for all variables and medium-term forecasts for external variables. Since the external variables are modelled as AR-processes, there is a unique mapping between the external variable and the corresponding shock. Concerning the conditioning on near-term forecasts (up to one quarter ahead), the conditioning information is mapped to all the active shocks in the model. If you want to construct a scenario with alternative NTFs this can be most easily done in the csv data file. However, in cases where you want to condition on endogenous variables in the short to medium term using a specific set of shocks (and not all active shocks), we recommend using the code given in the conditional forecasts section.

In the first example used, we construct an alternative scenario where the policy rate is assumed to be kept constant over the whole forecasts period. As we have understood, the MPC sometimes is interested in this kind of 'procastination' exercise. This should raise the question of why interest rates are assumed to remain constant. Is it because demand is higher or because of cost-push pressures or otherwise? Or is it simply an exercise in which the central bank deviates from its normal reaction pattern due to factors outside the model. In the latter case, we would argue that this is best represented as monetary policy shocks. Hence, it would be natural to keep all other implied shocks constant and attribute the constant interest rate to the monetary policy shock only.

```
rn_table = {'2024Q2',    5.9;...
            '2024Q3',    5.9;...
            '2024Q4',    5.9;...
            '2025Q1',    5.9;...
            '2025Q2',    5.9;...
            '2025Q3',    5.9;...
            '2025Q4',    5.9};
```

To achieve this, we start by setting up a table which will be of type 'cell', where we specify the dates and values that the interest rate is going to take. Given the data vintage at hand (as of January 2024), this would imply that the policy rate took on the current value given for the whole forecasting period. It is of course possible to impose any value of a given model variable for any future period or date and allow. The example in the run-script also includes a table which lists conditioning values the real exchange rate, similar to the example with the policy rate. The only thing you must make sure is that each date in the tables is corresponding to one number and vice versa.

Next, the conditioning values are taken from the table (cell) and put in an array. you then define the scenario dates (range), the variables you want to condition on (in this case real exchange rate and the policy rate) and the corresponding shocks that you want to explain the conditioning assumptions (in this case the (monetary policy shock and the risk premium shock in the UIP equation). Next step is to fetch the smoothed values for the model variables and shocks obtained from the Kalman filter and add the conditioning information defined in the table for the forecasting horizon.

```
% Get the dates from "rn_table"
scenariorange = qq(str2double(rn_table{1,1}(1:4)),str2double(rn_table{1,1}...
    (6))):qq(str2double(rn_table{end,1}(1:4)),str2double(rn_table{end,1}(6)));

% Specify the variable you want to condition on
cond_var = {'l_z_gap','rn_gap'};

% Specify the shock you want to use to explain this variable
cond_shock = {'shock_prem','shock_rn_tar'};

% Get the database from the filtration
d1 = db_filt.mean;

% Define the level of the interest rate gap in the database
d1.rn_gap(scenariorange) = rn_gap;
d1.l_z_gap(scenariorange) = l_z_gap;
```

Crudely speaking, conditioning on endogenous variables amounts to finding a combination of the exogenous 'structural' shocks in the model that will – given the model and initial values – reproduce the imposed conditioning values. Hence, in some sense we are flipping what is assumed to be endogenous and what is assumed to be exogenous. Fixing future values for one single exogenous variable, like for example the policy rate, accounted for by one single shock, for example the monetary policy shock, will give a unique set of future values for the monetary policy shock. Given these shocks and the rest of the smoothed variables, this will reproduce the conditioning assumption regarding the policy rate. Hence, we ask the question, which monetary policy shocks are needed to produce a constant interest rate going forward. Below, we show how this is implemented in the run-script.

```
% Create a simulation plan
sp = plan(m,fcastrange);

% Specify the plan for conditional forecasting (do not change)
sp                  = exogenise(sp,cond_var, scenariorange);
sp                  = endogenise (sp, cond_shock,scenariorange);

% Produce forecasts
f_cond         = jforecast(m, d1, fcastrange, 'plan', sp,'anticipate=',false);
uncert_cond = dbextend(db_filt.std, f_cond.std);
db_cond_fcast = dbextend(db_filt.mean, f_cond.mean);

% Reporting
[db_cond_fcast] = reporting_miniQPM(db_cond_fcast,db_model,m,tnd,...
    uncert_cond,fcastrange,endDate);
```

The first step is to define a simulation plan, which is done using the IRIS function 'plan()'. Next, you specify as part of the simulation plan (here: sp) the variables that will be endogenous and exogenous, respectively (found in cond_var and cond_shock). Once the simulation plan is defined and specified, you can run the forecast function jforecast(), where information on the simulation plan is included.

Mean forecasts and uncertainty are saved in the data objects db_cond_fcast and uncert_cond, respectively. Finally, model variables are transformed back into reporting variables.

Section 7: Forecast charts

In this section forecast charts are produced for both the unconditional and conditional forecasts. The first lines of the section just set some options, renames the data objects containing the unconditional and conditional forecasts and set the chart legends, which are all stored in a separate object 'rep'.

```
%% Section 7: Forecast charts
rep = moz_reportSettings(...
            'cap',       'Forecast charts',...
            'filename',  'results\reports\forecast_charts_uncond_vs_cond',...
            'publish',   false,...
            'visible',   true);

db_chart.db1 = db_fcast;
db_chart.db2 = db_cond_fcast;

db_legends = {'Unconditional forecasts','Scenario 1'};
```

Then cells are created for all subsets of variables that you want to plot. These cells contain the variable names and corresponding subplot heading for all four figures plotted in each chart. Each chart can contain a maximum of four subplots. All the relevant settings and options is collected in the settings object 'rep' and rep.produce(), which is a method under the settings object. This requires the 'publish' property to be set to 'true, if also the property 'visible' is set to 'true', the charts will be displayed. The report is produced by the IRIS report class.

```
% Chart 3                              Variable name         Description
charts.CPI_Inflation_QoQ = {           'dl_cpi',             'CPI Inflation (%, QoQ @ar, log-appr.)';...
                                       'dl_cpi_core',        'Core CPI Inflation (%, QoQ @ar, log-appr.)';...
                                       'dl_cpi_fv',          'F&V CPI Inflation (%, QoQ @ar, log-appr.)';...
                                       'dl_cpi_admin',       'Admin CPI Inflation (%, QoQ @ar, log-appr.)'};
```

```
rep = forecast_charts_qpm(db_chart,db_legends,charts,rep,startfcast,endfcast);

rep.produce();
```

Section 8: Saving the results

In this section, you can save all the results you need to be saved. This can of course also be done successively in the preceding sections.

```
dbsave(db_fcast,'results/db_fcast_mini_qpm.csv');
```

In this example, all variables included in the db_fcast object are saved in folder 'results' with the name 'db_fcast_mini_qpm.csv'.

Section 9: Save a forecast report

This section saves quarterly forecasts for selected variables as a table, both in an excel file and a pdf. Any variable in 'db_fcast' can be included in the table by including the variable name, description and transformation in the cell 'tableList'.

The table reports forecasts for the entire forecast period.

```
% File name (no extension)
tableFileName = 'Forecast_table';

% Table title
tableTitle = 'Overview';

% List of variables in the table
%                Variable name          Description                 Transformation
tableList = {   'd4l_cpi',              'Headline CPI',             'percent, YoY';...
                'd4l_cpi_core',         'Core CPI',                 'percent, YoY';...
                'd4l_cpi_admin',        'Admin CPI',                'percent, YoY';...
                'd4l_cpi_fv',           'F&V CPI',                  'percent, YoY';...
                'd4l_y_xagrxgas',       'GDP growth',               'percent, YoY';...
                's',                    'MZN per USD',              '';...
                'rn',                   'Nominal interest rate',    'percent'};


% Date range for table
tablerange = fcastrange(1)-1:fcastrange(end);
```

Section 10: Creating fan charts step 1 (estimate AR-models)

As alluded to above, symmetric fan charts can be constructed based on the last section of the 'reporting_miniQPM' script. Sometimes, however, you would like to communicate that the risks are unbalanced or asymmetric. One way interpreting this in the context of the model, would be to think of the exogenous shocks as having asymmetric distributions. Note however that the model is estimated under the assumption that the shocks are normal. Hence, assuming asymmetric shocks would be a conditional statement. To simulate future paths based on asymmetric shock distributions is technically feasible but this is not a standard feature of IRIS.

Here we provide an alternative approach that estimates fan charts based on simple Bayesian AR models for each variable of interest. Although not fully consistent with the estimated miniQPM model, these fan charts will conveniently reflect out-of sample forecast errors. As the forecast horizon increases the RMSE of the forecasts will converge to the unconditional standard deviation of the AR process.

```
%% Estimate Bayesian AR1 models


variables = {'d4l_cpi','d4l_y_xagr'};
db = db_fcast;


for ii = 1:length(variables)
    data = db.(variables{ii})(startDate:endDate).data;
    lags = 4;
    options.priors.name = 'Jeffreys'; % Prior
    options.fhor = 40; % Forecast horizon

    % Estimate a Bayesian AR model
    AR_mod = bvar_(data,lags,options);

    % Get the median forecasts
    medFC = median(AR_mod.forecasts.with_shocks,3);
    % Get the 16th percentile
    lowFC = quantile(AR_mod.forecasts.with_shocks,0.15866,3);

    % Compute the standard deviation, assuming a normal distribution
    sigma.(variables{ii}) = medFC - lowFC;

end

save('sd_ar1','sigma');
```

This section starts by listing the variables for which we would like to produce fan charts. Here you can include any variable that is contained in the 'db_fcast'.

Next we loop over the selected variables, and estimate a Bayesian AR model for each variable. Given the posterior, we can draw parameters and shocks to simulate forecasts, starting from the last observation. Here we produce forecasts from 1 to 40 periods ahead based on 5000 draws (stored in the struct 'AR_mod.forecasts.with_shocks') from the posterior. Assuming a normal distribution for the forecasts at all horizons, we can then calculate the median and the forecast value which is one standard deviation (the 15.866 percent quantile), using the fact that for any normal distribution: standard deviation=median-G(15.866), where G is the inverse function of the cumulative normal (F(X)). Again, the reason for going to all this trouble is that it is a more robust approach in situations with limited draws than calculating means and standard deviations directly based on the simulated forecasts.

Finally, we save the estimated standard deviations (of the forecasts) in the mat-file 'sd_ar'.

Section 11: Asymmetric fan charts step 2

In this section, we create the skewed fan charts. The purpose is in a simple way to generate asymmetric fan charts which serve as useful communication devices in situations where the risk outlook appears unbalanced. It is not, however, not based on any deep and fundamental conditional insights regarding the underlying exogenous shocks.

The approach is identical to the one used by Bank of England to communicate unbalanced risk in their inflation projections and is based on combining two normal distributions truncated at their original median (and mode and mean), i.e two piecewise truncated normal distributions, where the overall standard deviation for each forecast horizon is as estimated in section 9. The approach allows for varying the skewness of this combined distribution by changing the standard deviations in the two truncated normal distribution, leaving the overall standard deviation constant.

```
%% Plot asymmetric fan charts
% List variables you want to plot (max 4 in each plot)
% Chart 1                               Variable         skewness parameter
charts1.Main_Economic_Indicators = { 'd41_cpi',            0.2,
                                     'd41_y_xagr'        0.0,

% Define start and end date for the charts
startplot = qq(2012,1);
endplot = qq(2025,4);

rep = moz_reportSettings(...
            'cap',       'Fan charts',...
            'filename', 'results\reports\fan_charts_unconditional',...
            'publish',   true,...
            'visible',   true);
```

The first step is to create a cell including the variable names and the skewness parameter. The skewness parameter can take on any value between -1 and 1, where a value of 0 indicate that the distribution is symmetric. This list is stored in the ‘charts1’ struct. Next, we define start and end dates for the fan chart plots. Settings are collected and stored in the rep object.

```
% Define the width of the fan chart (in percent)
fanWidth = [30 60 90];

% Load standard deviations from AR1 models
load('sd_ar1');

% Length of forecast horizon
fchor = length(fcastrange);

% List of charts
chartList = fieldnames(charts1);
```

Next, we define the sizes of the different colored probability bands (here the 30%, 60% and 90% bands), load the estimated standard deviations of the autoregressive processes saved in section 9, set the forecast range and collect the variables list.

```
% Compute percentiles for fan chart
for jj = 1:length(chartList)
    variables = charts1.(chartList{jj})(:,1);
    skewness  = charts1.(chartList{jj})(:,2);

    for ii = 1:length(variables)
        sigma_ii = sigma.(variables{ii})(1:fchor);
        gamma_ii = skewness{ii};
        point_fcast = db_fcast.(variables{ii})(fcastrange);

        % Compute percentiles for fan charts
        conf.(variables{ii}) = splitNormal(point_fcast,sigma_ii,gamma_ii,fanWidth);

        % Transform the data into a matrix
        pNames = fieldnames(conf.(variables{ii}));
        conf_mat_temp = nan(fchor,length(pNames)+1);
        counter = 0;
        for kk = 1:length(pNames)+1
            if kk==(length(pNames)/2+1)
                conf_mat_temp(:,kk) = point_fcast;
            else
                counter = counter+1;
                conf_mat_temp(:,kk) = conf.(variables{ii}).(pNames{counter});
            end
        end
        temp_mat = ones(length(startplot:startfcast-1),length(pNames)+1).*...
            db_fcast.(variables{ii})(startplot:startfcast-1);
        fans.(variables{ii}) = [temp_mat; conf_mat_temp(1:length(startfcast:endplot),:)];
    end

end
```

For all variables, we start by collecting the standard deviation (sigma), the skewness (gamma), and the actual mode forecasts. The percentiles are calculated using the ‘splitNormal()’ function, which takes the overall uncertainty as given by ‘”sigma” in ‘sd_ar1.mat’. Given the skewness parameter, gamma, one can calculate the corresponding standard deviations for the two separate normal distributions, respectively (sigma1 and sigma2 in ‘splitNormal.m’). Given the mode, this allows us to construct the two-piece potentially skewed distribution for all the forecast horizons chosen.

Section 12: Historical (and forecast) decomposition

In DSGE models, the source of dynamics away from the steady state are the “structural” shocks. The model equations describe the intrinsic propagation of the shocks. In the absence extrinsic shocks, the model variables will eventually remain constant at their steady state values.

It is often useful and illuminating to explain movements in the endogenous variables in terms of its underlying exogenous drivers (the structural shocks), both the historical developments and projections. This is what is often referred to as shock decompositions.

In this section, shock decompositions for any model variable of interest can be calculated. In the example, we show decompositions for the policy rate (“rn_gap”).

```
%% Historical shock decompositions
figure_title = 'interest rate';

vars = {
'rn_gap'
};

m1   = m;
db1 = db_fcast;

mNames = {'benchmark'};

packages = {
% Name of group (char) ,    List of shock in group (char | cellstr)
% 'Initial'                 ,    'Initial'
'Demand'       ,    'shock_l_y_xagr_gap'
'Prices'       ,    {'shock_dl_cpi_core','shock_dl_cpi_fv','shock_dl_cpi_admin'}
% 'food'         ,
% 'admin',
'Monetary policy',  'shock_rn_tar'
'Oil price',        'shock_l_qoil_gap'
'foreign_cpi', 'shock_dl_cpi_us'
'foreign gdp', 'shock_l_y_us_gap'
'foreign interestrate', 'shock_rn_foreign'
'Exchange rate',    'shock_prem'
% 'Fiscal impulse',   'shock_fi'
};

% Do shock decomposition

    dec = moz_shockDecomp(m1,db1,vars,packages);
```

In the first part, you choose the model and data to be used. The data are taken from the 'db_fcast', which includes both the historical (filtered) variables and the forecasts. Hence, without restricting the data range further, decompositions will as default be calculated for both the history and projections.

Next, we chose which shock contributions we would like to see. In the example, we include shocks to the IS-cure (which we term 'Demand') and the inflation equations (the two Phillips curves and the process for admin prices). The shocks to the price equations are collected in a group called 'Prices'. Note however, that you can freely chose the grouping you find useful. In addition, we include the monetary policy shock ('shock_rn_tar), external shocks including the oil price shock and the risk premium shock in the UIP condition. Contributions from the remaining shocks in the model, which are not included in the list, will show up in the chart under the heading 'rest'.

Having specified the list of variables and shocks, we run 'moz_shockDecomp' to produce the marginal shock contributions. The remaining part of the section specifies chart settings and produces the plot using the 'plotter' function in nb_toolbox.