3.1. News Shock Method

Occasionally Binding Constraints in DSGE Models

Jonathan Swarbrick[1]
Bank of Canada

Bank of Canada – CMFE-Carleton Virtual Series
Advanced Topics in Macroeconomic Modelling

January 2021

[1]The views expressed are those of the authors and should not be interpreted as reflecting the views of the Bank of Canada.

Concept

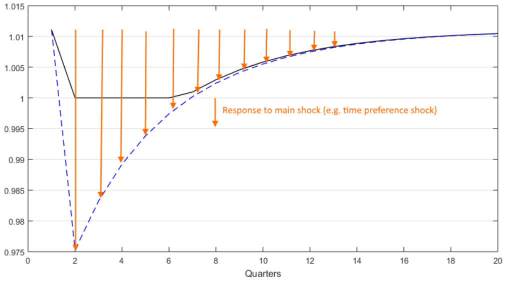
Idea: use add factors or news shocks to impose the bounds

- These are an endogenous (and therefore predictable) source of information about the constraint

Concept

Idea: use add factors or news shocks to impose the bounds

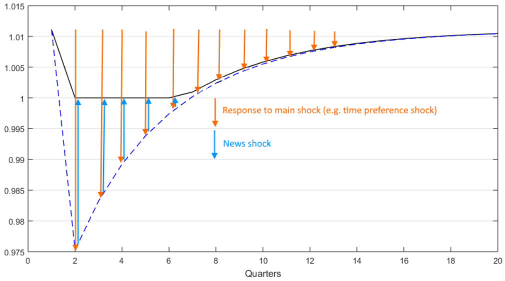
- These are an endogenous (and therefore predictable) source of information about the constraint
- If an OBC would otherwise be violated, anticipated add factors push the constrained variable(s) back to their bound.



Concept

Idea: use add factors or news shocks to impose the bounds

- These are an endogenous (and therefore predictable) source of information about the constraint
- If an OBC would otherwise be violated, anticipated add factors push the constrained variable(s) back to their bound.
- A shock causing an OBC to bind is then a combination of the original shock and a sequence of *news shocks*
- The new shocks state that the variable will be higher than it otherwise would for a while



Add factor

Recall that the sufficient condition for charactizing household borrowing is

$$b_t = \max\left\{\exp(z_t) + rb_{t-1} - \frac{1+\chi}{r\beta\mathbb{E}_t\left[\frac{1}{c_{t+1}}\right] - 2\delta b_t}, \underline{b}\right\}. \tag{1}$$

We can replace this condition with

$$b_t = \exp(z_t) + rb_{t-1} - \frac{1+\chi}{r\beta\mathbb{E}_t\left[\frac{1}{c_{t+1}}\right] - 2\delta b_t} + a_t \tag{2}$$

a_t is an 'add factor', solved to ensure $b_t \geq \underline{b}$.

Solving add factors

Suppose a shock pushes down borrowing to the constraint.

We can use an extended-path type approach

- This is the approach implemented in **IRIS**
- Similar to OccBin, but more general than OBCs
- Use linear policy function when away from the bound
- When the constraint binds, use the non-linear equation above, guess how long the constraint will bind and solve the sequence for a_t to satisfy the OBC

News shocks

Alternatively, we can use news shocks

$$a_t = \sum_{s=0}^{T} y_{t,t-s} \tag{3}$$

where $y_{t,t-s}$ is a news shock observed in period $t - s$, but hitting in period t.

- This is the approach at the heart of **dynareOBC**
- Notice that the add-factor is predictable from the state
- Include the news shocks in the model before it is linearized/solved
- Solve the sequence of news shocks to satisfy the OBC using the linear policy function

Naive implementation 1/2

See the code in **borrowing_constraints/news_shocks/soe_obc_simulate.m**

- Choose a horizon beyond which you expect to be away from the OBC, e.g., $T = 30$
- Add news shocks known this number of periods in advance to the model code:

```
varexo epsz news news_1 news_2 news_3 ... news_30

...

add_factor = news + news_1(-1) + news_2(-2) + ... + news_30(-30);
```

  Now the add factor is a function of news shocks
- Solve the model as usual, the news shocks will appear in the policy function, showing up as lots more state variables.
- Simulate the model period-by-period, each time checking if $b_t < \underline{b}$.
- If so, solve the 30 news shocks observed that period necessary to impose the OBC...

Naive implementation 2/2

- Notice that the period t policy function is

$$b_t = g\left(b_{t-1}, z_{t-1}, \epsilon_t, \left\{\left\{y_{t+k,t+k-s}\right\}_{s=k}^{T}\right\}_{k=0}^{T}\right) \tag{4}$$

- To solve a perfect-foresight IRF for a shock causing the OBC to bind in period $t = s$, we only need news shocks *observed* at $t + k - s$, and hitting the constrained equation in periods $t + k$
- The policy function is then:

$$b_t = g\left(b_{t-1}, z_{t-1}, y_{t,1}\right) \tag{5}$$

- We already know $z_t \forall t$ and $b_0 = \underline{b}$, so only want to solve $\{y_{t,1}\}_{t=1}^{1+T}$ satisfying:

$$\begin{aligned}
y_{t,s} &\geq 0 && \text{Shocks only push } b \textbf{ up} \text{ to bound} && (6)\\
y_{t,s}(b_t - \underline{b}) &= 0 && \text{Shocks only when } b \text{ constrained} && (7)\\
b_t &\geq \underline{b} && \text{Constraint satisfied} && (8)
\end{aligned}$$

- We could use `fmincon` to minimize the shocks, satisfying the constraints.

DynareOBC perfect foresight solver

- There is a much better way to characterize the problem
- Lets first compute the path of $\mathcal{B}_t \equiv b_t - \underline{b}$ for some arbitrary vector $y \in \mathbb{R}^T$.
  - Let $m_k \in \mathbb{R}^T$ be a column vector with impulse response of $\mathcal{B}_t$ to a news shock of size 1 at period k with $\mathcal{B}_0 = 0$, and let

$$M \equiv \begin{bmatrix} m_1 & m_2 & \cdots & m_T \end{bmatrix} \tag{9}$$

  - It follows that the path of $\mathcal{B}_t$ is given by

$$My \tag{10}$$

- The path of $\mathcal{B}_t$ given the original shock with the addition of arbitrary news shock vector y is

$$q + My \tag{11}$$

- q is a $T \times 1$ vector with IRF of $\mathcal{B}_t \equiv b_t - \underline{b}$ to the shock up to horizon T
- $y = \begin{bmatrix} y_{1,1} & y_{2,1} & \cdots & y_{T,1} \end{bmatrix}'$ is a vector of news shocks up to horizon T
- m_k is a $T \times 1$ vector with IRF of $\mathcal{B}_t$ to news shock at period k
- $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_T \end{bmatrix}$ is a $T \times T$ matrix

Solving news shocks: LCP

- We want to compute the vector of shocks $y \in \mathbb{R}^T$ to impose the bound:

$$\mathcal{B} = q + My \geq 0, \tag{12}$$

- q is a $T \times 1$ vector with IRF of b_t to the shock up to horizon T
- $\eta = \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & \cdots & \eta_{1,T} \end{bmatrix}'$ is a vector of news shocks up to horizon T
- m_k is a $T \times 1$ vector with IRF of b_t to news shock at period k
- $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_T \end{bmatrix}$ is a $T \times T$ matrix

Solving news shocks: LCP

- We want to compute the vector of shocks $y \in \mathbb{R}^T$ to impose the bound:

$$\mathcal{B} = q + My \geq 0, \tag{12}$$

- so that the news shocks are only used to impose the bound:

$$y\,(q + My) = 0 \tag{13}$$
$$y \geq 0. \tag{14}$$

- q is a $T \times 1$ vector with IRF of b_t to the shock up to horizon T
- $\eta = \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & \cdots & \eta_{1,T} \end{bmatrix}'$ is a vector of news shocks up to horizon T
- m_k is a $T \times 1$ vector with IRF of b_t to news shock at period k
- $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_T \end{bmatrix}$ is a $T \times T$ matrix

Solving news shocks: LCP

- We want to compute the vector of shocks $y \in \mathbb{R}^T$ to impose the bound:

$$\mathcal{B} = q + My \geq 0, \tag{12}$$

- so that the news shocks are only used to impose the bound:

$$y\,(q + My) = 0 \tag{13}$$
$$y \geq 0. \tag{14}$$

- For a given q and M, the news shock problem is characterised as a linear complementarity problem, LCP(q, M), to find y subject to equations (12), (13) and (14).

- q is a $T \times 1$ vector with IRF of b_t to the shock up to horizon T
- $\eta = \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & \cdots & \eta_{1,T} \end{bmatrix}'$ is a vector of news shocks up to horizon T
- m_k is a $T \times 1$ vector with IRF of b_t to news shock at period k
- $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_T \end{bmatrix}$ is a $T \times T$ matrix

Further comments

Some comments:

- dynareOBC represents LCP(q, M) as a mixed integer linear programming problem
  - Very well studied class of problem, with many, well optimized, fully global solvers.
- The terminal condition implies agents believe the economy will return to a given (locally determinate) steady-state.
  - in ELB model, equivalent to assuming long-run inflation target is credible
  - in the example borrowing constraints model, there is only one steady state
- The MILP formulation allows careful consideration of multiple short-run equilibria is they exist
  - Unique in that guaranteed to find a solution if one exists
  - dynareOBC will report whether a solution exists and if there are multiple
- Extends to multiple OBCs and higher-order pruned perturbations

Obtaining DynareOBC

- You can download the latest DynareOBC release from this page:

  https://github.com/tholden/dynareOBC/releases

  - DynareOBC should auto-update itself the first time it runs.

- Or, you can click "clone or download" then "download ZIP" on the DynareOBC homepage.
  - This does not give all dependencies, but DynareOBC should automatically download these dependencies the first time it runs.

- DynareOBC works best if you have admin rights on your own machine, so it may install dependencies automatically.
  - If you don't have admin rights, the ReadMe file contains details of what your IT administrator must install for you.

DynareOBC installation

- To install, just add the root DynareOBC folder (i.e. the one containing **dynareOBC.m**) to your MATLAB path.
- If you have installed a MILP solver (recommended, not compulsory) that also has to be added to your path.
  - We will be using Gurobi.
- Once you have done this, test DynareOBC by running (in MATLAB):

```
dynareOBC testsolvers
```

  This should produce output roughly as shown on the next slide.

- After running **TestSolvers**, it is a good idea to clean up the path using:

```
dynareOBC rmpath
```

Sample DynareOBC TestSolvers Output

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|                   Test|    Solution|                                 Solver message|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   Core functionalities|         N/A|                   Successfully solved (YALMIP)|
|                     LP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                     LP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                     QP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                     QP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                   SOCP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                   SOCP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                   SOCP|     Correct|             Successfully solved (GUROBI-GUROBI)|
|                    SDP|     Correct|                 Successfully solved (DSDP-OPTI)|
|                    SDP|     Correct|                 Successfully solved (DSDP-OPTI)|
|                    SDP|     Correct|                 Successfully solved (DSDP-OPTI)|
|                    SDP|     Correct|                 Successfully solved (DSDP-OPTI)|
|                 MAXDET|   Incorrect|   Maximum iterations or time limit exceeded (DSDP-OPTI)|
|                 MAXDET|   Incorrect|                 Successfully solved (DSDP-OPTI)|
|          Infeasible LP|         N/A|        Either infeasible or unbounded (GUROBI-GUROBI)|
|          Infeasible QP|         N/A|        Either infeasible or unbounded (GUROBI-GUROBI)|
|         Infeasible SDP|         N/A|                  Infeasible problem (DSDP-OPTI)|
|      Moment relaxation|     Correct|                 Successfully solved (DSDP-OPTI)|
|         Sum-of-squares|   Incorrect|   Maximum iterations or time limit exceeded (DSDP-OPTI)|
|           Bilinear SDP|         N/A|                             No suitable solver|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Checking OPTI Toolbox Installation:
Checking Paths...                  Ok
Checking   LP Solver Results...    Ok
Checking  MILP Solver Results...   Ok
Checking    QP Solver Results...   Ok
Checking  MIQP Solver Results...   Ok
Checking   SDP Solver Results...   Ok
Checking   NLS Solver Results...   Ok
```

Basic Use – Generating IRFs

- Open a MOD file for a model with an OBC, e.g. an NK model:

  BBW-NK-ZLB/news-shocks-irf/NK_ZLB.mod

- Make sure your `stoch_simul` command contains `periods=0` and `irf=40` (or some other positive integer).

- You enter the equation with the max / min operator in the model block

- Then, from the folder containing the MOD file, run:

```
dynareOBC mod_file_name.mod
```

  E.g.:

```
dynareOBC NK_ZLB.mod
```

DynareOBC's Behaviour 1/2

- Watching the output, you will see three separate Dynare calls.
- You will also notice some printed diagnostics about existence and uniqueness.
  - These will be discussed next week.
- You will then see some IRFs.
- In these IRFs:
  - The solid line is the impulse response imposing the constraint(s).
  - The dotted line is the path in the absence of constraints.
- IRFs are stored in **oo_.irfs**, as in standard Dynare.

DynareOBC's Behaviour 2/2

- Using the BKW model, you might find that the ZLB has not been hit in any of the IRFs.
- To fix this, we need to increase the scale of the impulse used to generate the IRFs. This may be done by running e.g.:

```
dynareOBC NK_ZLB shockscale=3
```

- The number 3 after `ShockScale` specifies that we want the IRF to magnitude 3 shocks.
- `ShockScale` does not change the standard deviation of the model's shocks, it only scales the initial impulse.
  - This difference will be important when we look at rational expectations rather than perfect foresight solutions.
- Another option is `bypass` which ignores any non-differentiable functions, i.e. treats constraints as always/never binding.

DynareOBC Options

- `bypass` and `shockscale=3` are some of DynareOBC's many options.
- Options are **not** case sensitive, so you can as well enter `ShockScale`.
- When an option takes a parameter (such as `shockscale`), there must not be spaces between the option name, the equals sign and the value.
- To see all of DynareOBC's options, just enter `dynareOBC` without arguments in MATLAB.
- Alternatively, view the DynareOBC home-page: https://github.com/tholden/dynareOBC
  Or look at **README.md**, **ReadMe.txt** or **ReadMe.pdf** in the DynareOBC root directory.

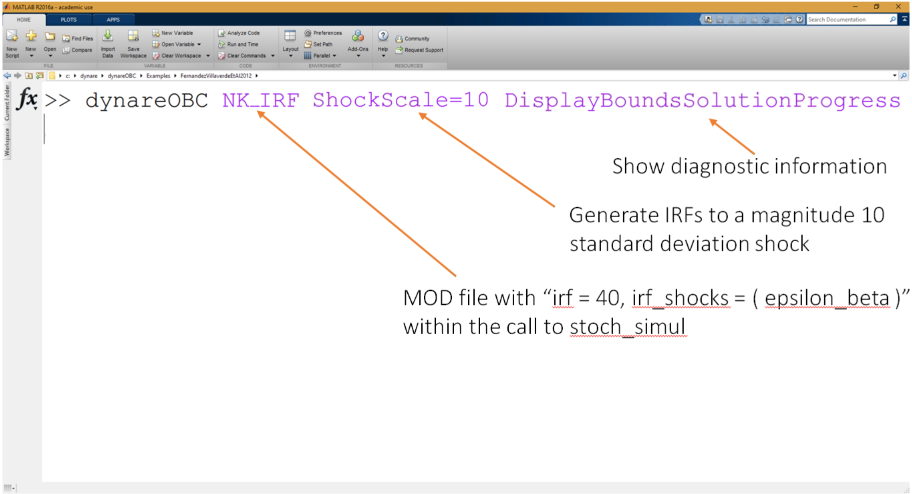
Useful references for other MOD files for experimentation

- DynareOBC contains a number of examples within its **Examples** sub-folder.
  - As does Dynare, again within its examples sub-folder.
- Johannes Pfeifer maintains a large collection of MOD files for popular models here:

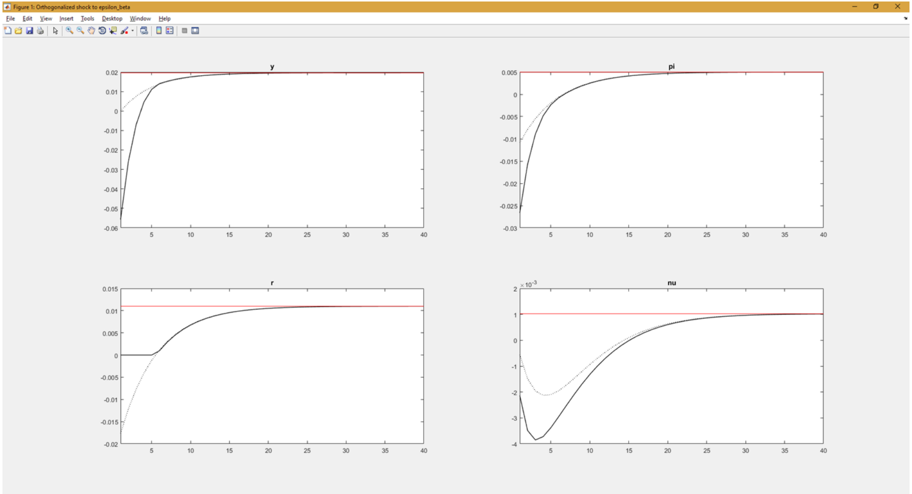
  https://github.com/JohannesPfeifer/DSGE_mod
- The Wieland et al. model database contains further MOD files for standard models here:

  http://www.macromodelbase.com/download/
- You no doubt have many models of your own floating around your hard drive.

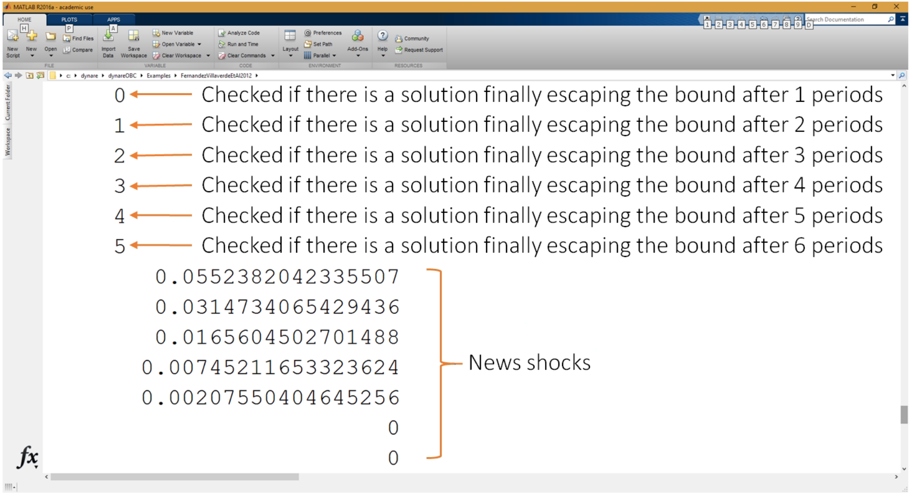
Generating a perfect foresight IRF with diagnostic information



DynareOBC outputs IRF plots as usual ...



… and some diagnostics



Relating LCP problem with IRFs

- Recall the IRF of bounded variable x_1 above, was given as:

$$x_1 = q + My \tag{15}$$

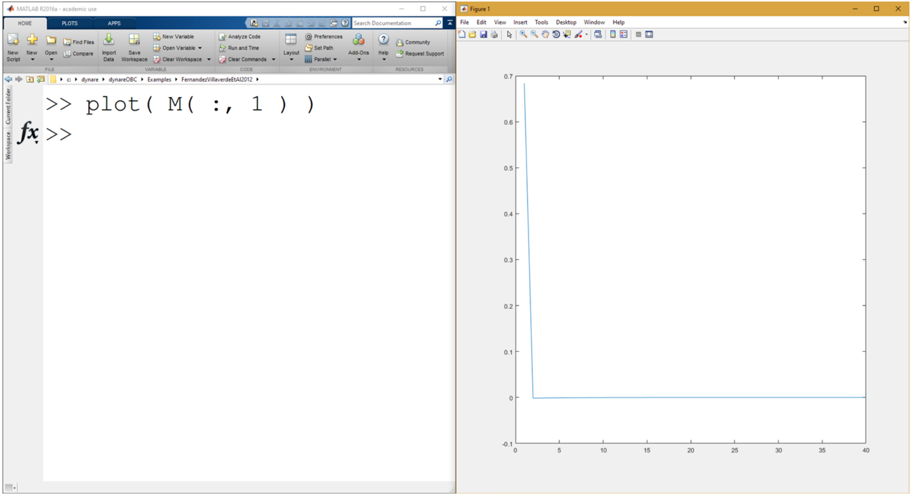
  where q is the impulse response to the original shock, y a vector of news shocks, and M the matrix with the impulse responses to the news shocks

- We can plot the IRF of the interest rate using:

```
r_steady = log( 1.005 / 0.994 );
q = dynareOBC_.IRFsWithoutBounds.r_epsilon_b' + r_steady;
M = dynareOBC_.MMatrix( 1:40, 1:5 );
y =  [  0.05523820423354 9
        0.03147340654294 3
        0.01656045027014 8
        0.00745211653323 6
        0.00207550404645 2 ];
plot( 1:40, q, ':k', 1:40, q + M * y, '-k' );
```

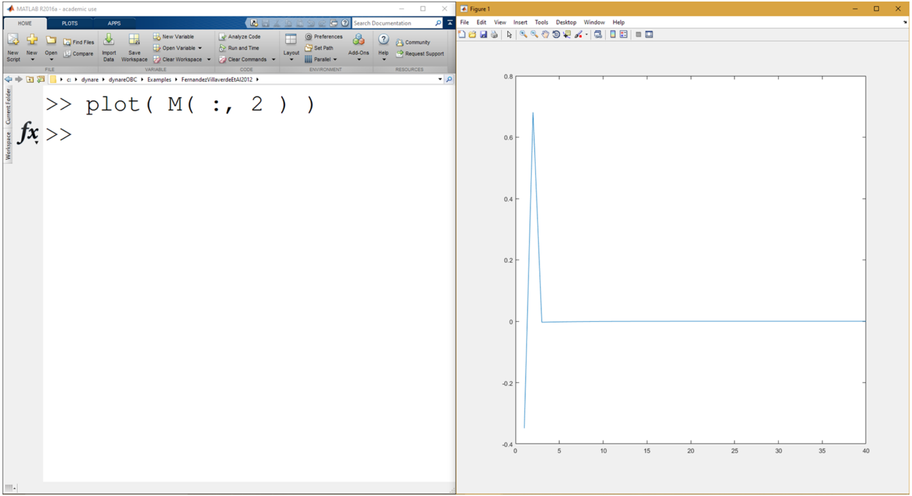
The M Matrix

The first column of M is an IRF to a monetary policy shock:



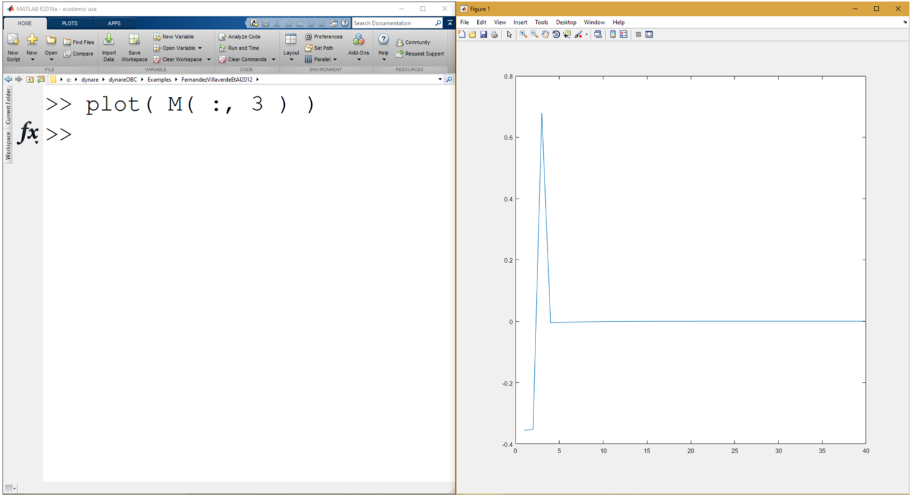
The M Matrix

The 2nd is an IRF to news of a monetary policy shock hitting 1 period ahead:



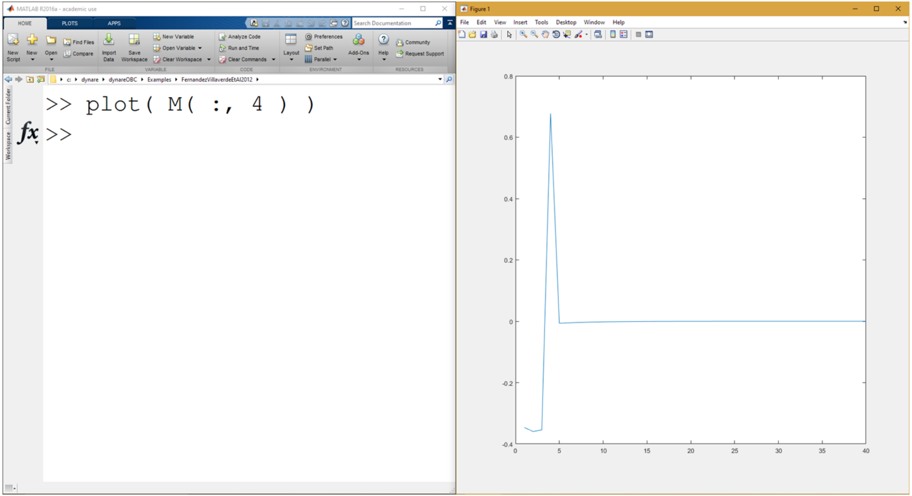
The M Matrix

The 3rd is an IRF to news of a monetary policy shock hitting 2 periods ahead:



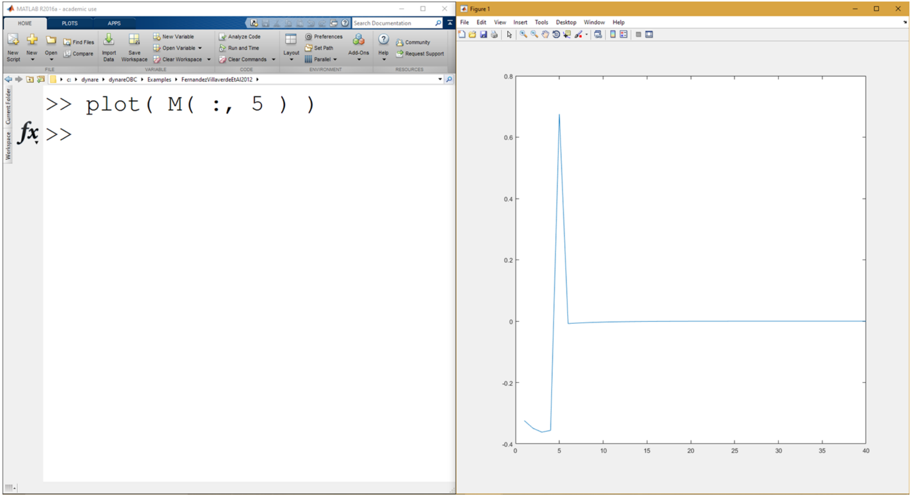
The M Matrix

The 4th is an IRF to news of a monetary policy shock hitting 3 periods ahead:



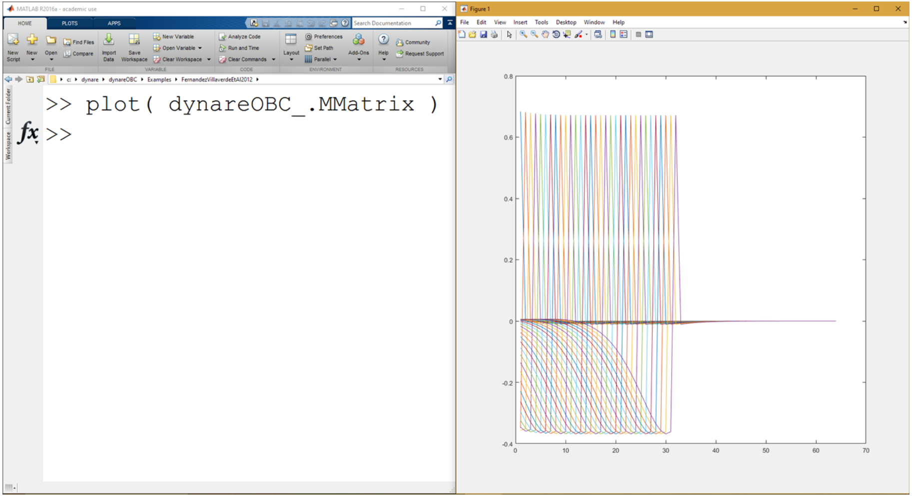
The M Matrix

The 5th is an IRF to news of a monetary policy shock hitting 4 periods ahead:

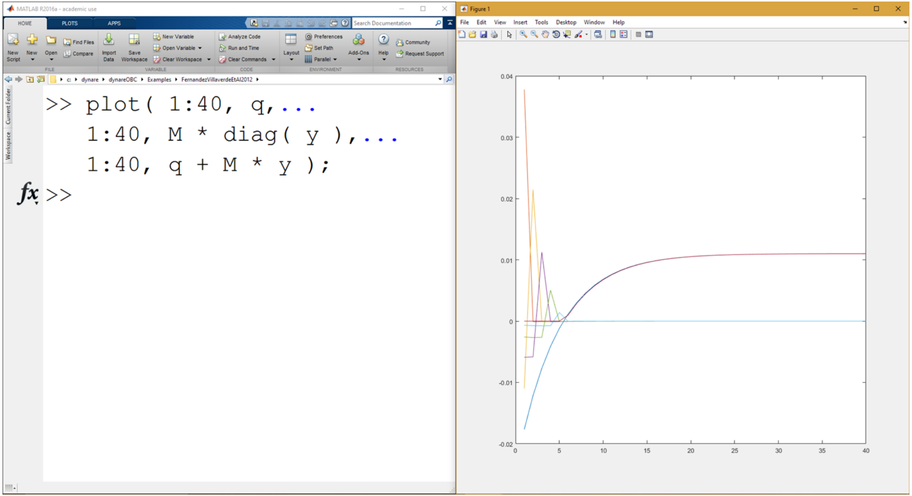


The M Matrix

and so on:



The solution



Simulation of Models with OBCs with DynareOBC

- The algorithm extends to an extended-path type simulation.
  - Every period, we solve for the expected future path of the economy ignoring the constraint. This takes the place of q.
  - Under perfect-foresight, we need to find y to impose the bound.
  - As with the extended-path algorithm, this is done period-by-period to compute the news shocks up to a horizon T

Simulation of Models with OBCs with DynareOBC

- The algorithm extends to an extended-path type simulation.
  - Every period, we solve for the expected future path of the economy ignoring the constraint. This takes the place of q.
  - Under perfect-foresight, we need to find y to impose the bound.
  - As with the extended-path algorithm, this is done period-by-period to compute the news shocks up to a horizon T
- Try using the code: **BBW-NK-ZLB/news-shocks-extended-path/do.m**
  which solves the model using:

```
dynareOBC NK_ZLB
```

Simulation of Models with OBCs with DynareOBC

- The algorithm extends to an extended-path type simulation.
  - Every period, we solve for the expected future path of the economy ignoring the constraint. This takes the place of q.
  - Under perfect-foresight, we need to find y to impose the bound.
  - As with the extended-path algorithm, this is done period-by-period to compute the news shocks up to a horizon T
- Try using the code: **BBW-NK-ZLB/news-shocks-extended-path/do.m** which solves the model using:

```
dynareOBC NK_ZLB
```

- You may notice that DynareOBC now spends some time obtaining parameteric solutions.
  - These speed up solving bound problems in which the bound only binds for a few periods.

DynareOBC simulation problems

- Following the previous slide, you will receive the message:
  `Impossible problem encountered. Try increasing TimeToEscapeBounds, or reducing the magnitude of shocks.`

DynareOBC simulation problems

- Following the previous slide, you will receive the message:
  `Impossible problem encountered. Try increasing TimeToEscapeBounds, or reducing the magnitude of shocks.`
- This means that there is definitely no solution to the particular LCP that the algorithm was attempting to solve.
  - This contrasts with the other solution methods we've discussed, in which a failure to converge to a solution may just reflect a failure of the algorithm.

DynareOBC simulation problems

- Following the previous slide, you will receive the message:
  `Impossible problem encountered. Try increasing TimeToEscapeBounds, or reducing the magnitude of shocks.`
- This means that there is definitely no solution to the particular LCP that the algorithm was attempting to solve.
  - This contrasts with the other solution methods we've discussed, in which a failure to converge to a solution may just reflect a failure of the algorithm.
- However, in this case the problem is fixable by increasing `TimeToEscapeBounds`, which is the maximum T the algorithm considers. Its default value is 32, so we might try running, e.g.:

```
dynareOBC NK_ZLB.mod timetoescapebounds=64
```

DynareOBC Simulation Output

Like standard Dynare, DynareOBC will print assorted moments of the simulated variables.

- Note that the presence of occasionally binding constraints often leads to substantial skewness.
- As in standard Dynare, DynareOBC discards the initial simulation periods. The precise number of periods discarded is controlled by the drop option to `stoch_simul`.
- Note that DynareOBC starts simulations from a draw from the stationary distribution in the absence of OBCs, so large values for drop should not be needed.

As in standard Dynare, the simulated variables (with the bounds imposed) are stored in `oo_.endo_variables`, which is ordered in declaration order.

- `dynareOBC_.SimulationsWithoutBounds` contains the simulation without imposing the bound, for comparison.

Some generalizations

- The approach generalizes to multiple bounds:
  - DynareOBC stacks IRF of bounded variables ignoring bounds into q
  - and stacks vectors of news shocks into y
  - M is then a block matrix with each bounded variable's response to each bounded variable's news shocks
  - The stacked solution for paths is $q + My$ and everything goes through as above.

Some generalizations

- The approach generalizes to multiple bounds:
  - DynareOBC stacks IRF of bounded variables ignoring bounds into q
  - and stacks vectors of news shocks into y
  - M is then a block matrix with each bounded variable's response to each bounded variable's news shocks
  - The stacked solution for paths is $q + My$ and everything goes through as above.

- Dynare converts any bounded equation to the form $x_t = \max\{0, x_t^*\}$
  - e.g.,if $x_t = \min\{0, x_t^*\}$, then dynare converts it to $-x_t = \max\{0, -x_t^*\}$
  - if $x_t = \max\{z_t, y_t\}$, then dynare converts it to $x_t - z_t = \max\{0, y_t - z_t\}$

Some generalizations

- The approach generalizes to multiple bounds:
  - DynareOBC stacks IRF of bounded variables ignoring bounds into q
  - and stacks vectors of news shocks into y
  - M is then a block matrix with each bounded variable's response to each bounded variable's news shocks
  - The stacked solution for paths is $q + My$ and everything goes through as above.
- Dynare converts any bounded equation to the form $x_t = \max\{0, x_t^*\}$
  - e.g.,if $x_t = \min\{0, x_t^*\}$, then dynare converts it to $-x_t = \max\{0, -x_t^*\}$
  - if $x_t = \max\{z_t, y_t\}$, then dynare converts it to $x_t - z_t = \max\{0, y_t - z_t\}$
- The approach generalizes to higher order pruned perturbation approximation using approach in Lan & Meyer-gohde (2013)
  - A d-order pruned perturbation approximation is linear in shocks to power of d, so additive effects of shocks maintained with y^d.
  - (Also implies closed form covariance matrix which will be useful for integration discussed next).

Precautionary Effects

Risk of constraints binding

- When away from constraints, the risk of their future binding should affect agent behaviour
- Under regime switching, this could be captured with transition probabilities
- Similarly, news shocks should be partly predictable from the state
- We can follow the stochastic extended-path method and integrate over future uncertainty period-by-period

Integrating over future uncertainty 1/2

To evaluate uncertainty, we follow the stochastic extended-path method and integrate over S periods to determine expected path of the news shocks, y, and bounded variables

- Due to nice properties of pruned perturbation solutions obtain Gaussian approximation without Monte-Carlo simulation
- this is because the covariance matrix ignoring the bounds is available in closed form
- We can integrate over S periods of future uncertainty as a S dimension problem: much quicker than existing S.E.P. in Dynare (exponential in S).
- Using sparse cubature rules can make cubature degree dynamic so simulation very quick away from the bound.
- The underlying perfect foresight solver is considerably faster than dynare's

Integrating over future uncertainty 2/2

To avoid a possible ‘step’ at horizon S, use cosine windowing function to *phase out* the shock variance. If the ‘true’ shock variance is Σ, we use

$$\hat{\Sigma}_k = \frac{1}{2}\left[1 + \cos\left(\pi \frac{\min\{k-1, S\}}{S}\right)\right]\Sigma \tag{16}$$

Integrating over future uncertainty 2/2

To avoid a possible 'step' at horizon S, use cosine windowing function to *phase out* the shock variance. If the 'true' shock variance is Σ, we use

$$\hat{\Sigma}_k = \frac{1}{2}\left[1 + \cos\left(\pi \frac{\min\{k-1, S\}}{S}\right)\right]\Sigma \tag{16}$$

Integrating over future OBC risk in DynareOBC

DynareOBC uses quasi-Monte Carlo with $2^{1+l} - 1$ Sobol points, $l \in \mathbb{N}$.

- This is invoked with the option `Cubature`, or setting any of the cubature options, such as:
  - `QuasiMonteCarloLevel=INTEGER` to set the number of samples, l
- To control the horizon to which to integrate, set `PeriodsOfUncertainty=INTEGER`
  - The default is 16 but because of the cosine window is used, in practice, it is roughly half this number.
- To speed up integration, we can use one of the following options:
  - `CubaturePruningCutOff=FLOAT` (default: 0.01) which 'prunes' eigenvalues of the covariance matrix less than FLOAT.
  - `CubatureRelWeightCutOff=FLOAT` (default: 0.0001) which 'prunes' integration weights less than FLOAT.
  - `MaxCubatureDimension=INTEGER` (default: 128). If the algorithm needs to integrate over a larger space, it will 'prune' all but the INTEGER largest eigenvalues of the covariance matrix to zero.
- There are also lots of options to slow things down (and increase accuracy!)

Impulse Response Functions with DynareOBC

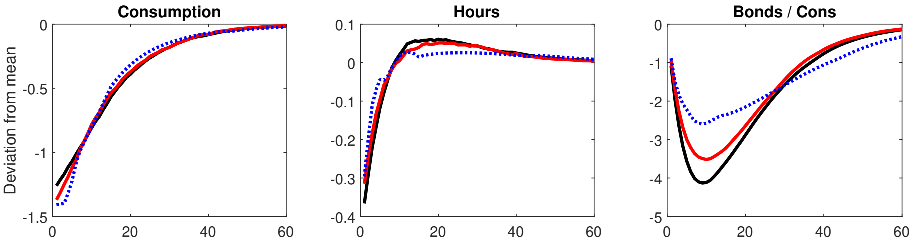
IRFs at first order:

- Without cubature, it is just a perfect-foresight IRF.

Higher orders:

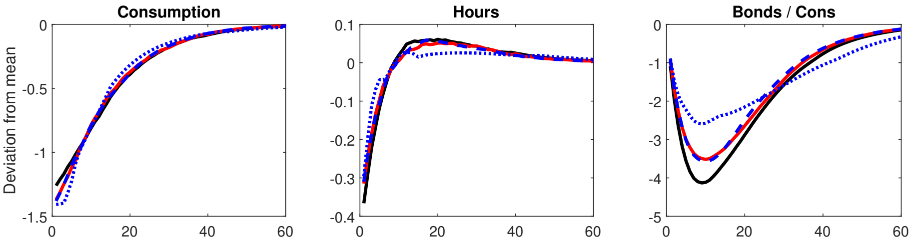
- With OBCs but without cubature, news shock problem solved, but future expectations of news shocks do not effect agent decisions.
- With cubature, we integrate out the effects of future uncertainty on the expectation of future news shocks, but not current uncertainty.
- So not generalized IRF (unconditional IRF)
- As before, we can do so via Monte-Carlo simulation.
  - Invoke using option `slowIRFs`
  - In addition to the mean IRF, you can compute median IRFs via Monte Carlo using `medianIRFs`

Impulse response functions to technology shock



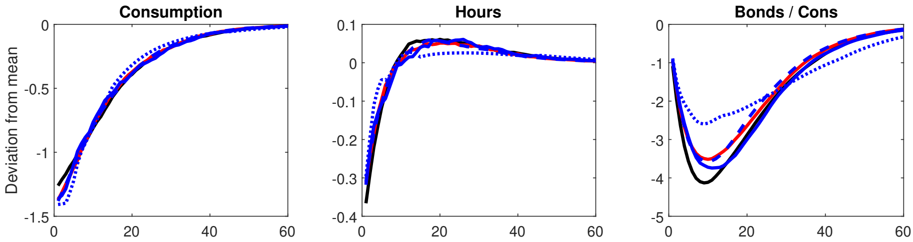
- Black = projection, red = OccBin (GIRF)
- blue dot = dynareOBC (cubature)

Impulse response functions to technology shock



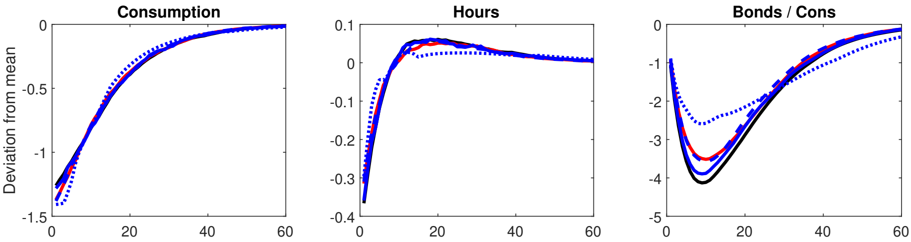
- Black = projection, red = OccBin (GIRF)
- blue dot = dynareOBC (cubature)
- blue dash = dynareOBC (GIRF)

Impulse response functions to technology shock



- Black = projection, red = OccBin (GIRF)
- blue dot = dynareOBC (cubature)
- blue dash = dynareOBC (GIRF)
- blue solid = dynareOBC (GIRF + cubature)

Impulse response functions to technology shock



- Black = projection, red = OccBin (GIRF)
- blue dot = dynareOBC (cubature)
- blue dash = dynareOBC (GIRF)
- blue solid = dynareOBC (GIRF + cubature) [2nd order]

Simulated Moments

		Mean	Standard deviation	Skewness	
		Relative to no constraint	Relative to no constraint	Baseline	No constraint
Projection methods	Consumption	+0.03%	+15%	-0.22	0.09
	Hours	-0.01%	-43%	-0.09	-0.04
	Bonds / $\bar{c}$	0.3% $\longrightarrow$ 5%	-59%	1.18	0.007

		Mean	Standard deviation	**Skewness** (projection)
		Relative to projection	Relative to projection	
VFI	Consumption	$\approx$ 0	+0.02%	-0.23 (-0.22)
	Hours	$\approx$ 0	+0.6%	-0.09 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.9%	<1%	1.18 (1.18)
Extended-path	Consumption	-0.03%	+0.5%	-0.29 (-0.22)
	Hours	+0.003%	+1%	-0.1 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.0%	-1.4%	1.3 (1.18)
Penalty function	Consumption	-0.02%	+25%	0.01 (-0.22)
	Hours	-0.002%	-99%	-0.1 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.4%	$\approx$-100%	-0.19 (1.18)
Piecewise linear	Consumption	-0.006%	+0.8%	-0.29 (-0.22)
	Hours	+0.003%	+5%	-0.4 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.0%	+0.9%	1.3 (1.18)
News shocks (order 1, no cubature)	Consumption	-0.06%	+0.7%	-0.36 (-0.22)
	Hours	+0.006%	+2%	-0.04 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.0%	-0.8%	1.3 (1.18)
News shocks (order 1, with cubature)	Consumption	-0.004%	-0.3%	-0.38 (-0.22)
	Hours	+0.005%	+2.5%	-0.09 (-0.09)
	Bonds / $\bar{c}$	5.0% $\longrightarrow$ 4.6%	-1.6%	1.3 (1.18)

Existence and Uniqueness

Existence and Uniqueness

- DynareOBC can check for the existence and uniqueness for the solution to models with OBCs
  - Depends on characteristics of the M-matrix.
- You will have likely noticed details in the output already
- For example, using Smets & Wouters (2003, 2007), we can find combinations of predicted future shocks for which:
  - there are multiple solutions (including conbinations with one solution in which r does not hit the ZLB)
  - There are zero solutions
  - Note: both models are determinate under price level targeting
- Holden (2019) sets out necessary and sufficient conditions for stability/uniqueness of linear models with an OBC

Some intuition for multiplicity

- Suppose we believe the bound will be escaped after 2 periods. The LCP problem is to find a solution $y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}'$ satisfying:

$$y_1, y_2 \geq 0 \tag{17}$$

$$\left(q_1 + M_{11} y_1 + M_{12} y_2\right), \left(q_2 + M_{21} y_1 + M_{22} y_2\right) \geq 0 \tag{18}$$

$$y_1 \left(q_1 + M_{11} y_1 + M_{12} y_2\right) = y_2 \left(q_2 + M_{21} y_1 + M_{22} y_2\right) = 0 \tag{19}$$

Some intuition for multiplicity

- Suppose we believe the bound will be escaped after 2 periods. The LCP problem is to find a solution $y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}'$ satisfying:

$$y_1, y_2 \geq 0 \tag{17}$$

$$\left(q_1 + M_{11} y_1 + M_{12} y_2\right), \left(q_2 + M_{21} y_1 + M_{22} y_2\right) \geq 0 \tag{18}$$

$$y_1 \left(q_1 + M_{11} y_1 + M_{12} y_2\right) = y_2 \left(q_2 + M_{21} y_1 + M_{22} y_2\right) = 0 \tag{19}$$

- There are two quadratics so up to 4 solutions, given by:
  1. $y_1 = y_2 = 0$. Exists if $q_1 \geq 0$ and $q_2 \geq 0$
  2. $y_1 = -\frac{q_1}{M_{11}}, y_2 = 0$. Exists if $\frac{q_1}{M_{11}} \leq 0$ and $M_{11} q_2 \geq M_{21} q_1$
  3. $y_1 = 0, y_2 = -\frac{q_2}{M_{22}}$. Exists if $\frac{q_2}{M_{22}} \leq 0$ and $M_{22} q_1 \geq M_{12} q_2$
  4. $y_1 = \frac{M_{12} q_2 - M_{22} q_1}{M_{11} M_{22} - M_{12} M_{21}}, y_2 = \frac{M_{21} q_1 - M_{11} q_2}{M_{11} M_{22} - M_{12} M_{21}}$. Exists if $M_{11} M_{22} - M_{12} M_{21} \leq 0$

Some intuition for multiplicity

- Suppose we believe the bound will be escaped after 2 periods. The LCP problem is to find a solution $y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}'$ satisfying:

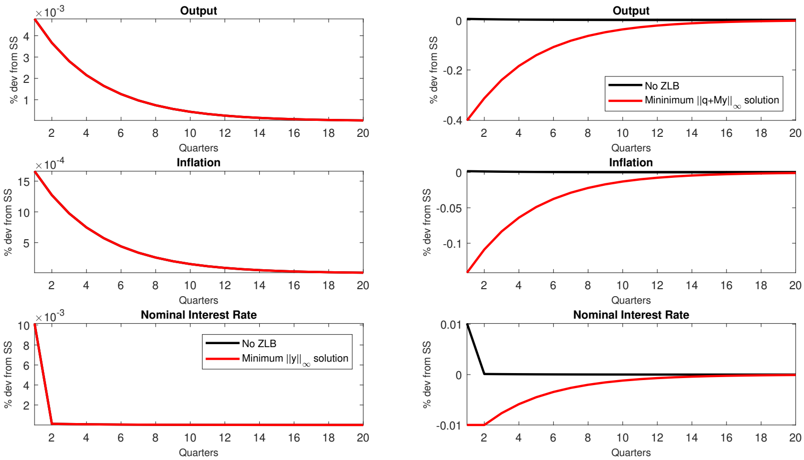
$$y_1, y_2 \geq 0 \tag{17}$$

$$(q_1 + M_{11}y_1 + M_{12}y_2), (q_2 + M_{21}y_1 + M_{22}y_2) \geq 0 \tag{18}$$

$$y_1 (q_1 + M_{11}y_1 + M_{12}y_2) = y_2 (q_2 + M_{21}y_1 + M_{22}y_2) = 0 \tag{19}$$

- There are two quadratics so up to 4 solutions, given by:
  1. $y_1 = y_2 = 0$. Exists if $q_1 \geq 0$ and $q_2 \geq 0$
  2. $y_1 = -\frac{q_1}{M_{11}}, y_2 = 0$. Exists if $\frac{q_1}{M_{11}} \leq 0$ and $M_{11}q_2 \geq M_{21}q_1$
  3. $y_1 = 0, y_2 = -\frac{q_2}{M_{22}}$. Exists if $\frac{q_2}{M_{22}} \leq 0$ and $M_{22}q_1 \geq M_{12}q_2$
  4. $y_1 = \frac{M_{12}q_2 - M_{22}q_1}{M_{11}M_{22} - M_{12}M_{21}}, y_2 = \frac{M_{21}q_1 - M_{11}q_2}{M_{11}M_{22} - M_{12}M_{21}}$. Exists if $M_{11}M_{22} - M_{12}M_{21} \leq 0$
- So multiple solutions for at least some combination of $q_1, q_2 \geq 0$ if
  1. $M_{11}, M_{21} \leq 0$, or,
  2. $M_{22}, M_{12} \leq 0$, or,
  3. $M_{11}M_{22} - M_{12}M_{21} \leq 0$

Example: Brendon, Paustian and Yates (2013) model



- Shock is to the Euler equation (BPY 3 equation NK model).

Computational approach and choice of equilibria

- DynareOBC expresses the LCP problem is a mixed integer linear programming problem (MILP)
  - this is a well studied problem with many good solution algorithms
- Given $\tilde{\omega} > 0$, $q \in \mathbb{R}^T$ and $M \in \mathbb{R}^{T \times T}$, we find $\alpha \in \mathbb{R}$, $\hat{y} \in \mathbb{R}^T$ and $z \in \{0, 1\}^T$ that satisfy:

$$0 \leq \alpha \tag{20}$$

$$0 \leq \hat{y} \leq z \tag{21}$$

$$0 \leq \alpha q + M\hat{y} \leq \tilde{\omega}\left(1_{T \times 1} - z\right) \tag{22}$$

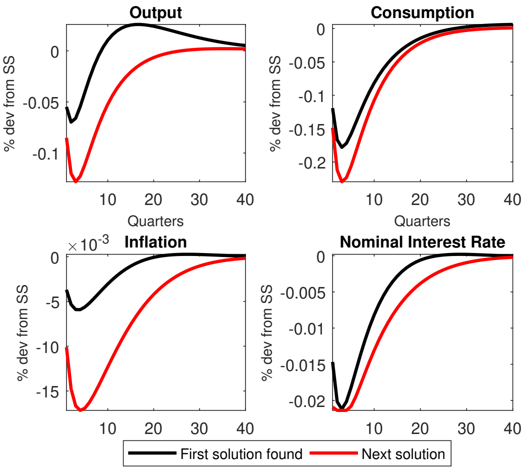
- If it exists, the solution will be $y = \frac{\hat{y}}{\alpha}$
- As $\tilde{\omega} \to 0$, the solution given will be one that minimises $\|q + My\|_\infty$
- As $\tilde{\omega} \to \infty$, the solution given will be one that minimises $\|y\|_\infty$
- DynareOBC sets $\tilde{\omega} = \omega \|q\|_\infty$, where ω is set using option `omega=0.0001` for example.
  - Setting, e.g., `omega=10000` would choose a solution minimising time at the bound.

More options related to inner solution procedure

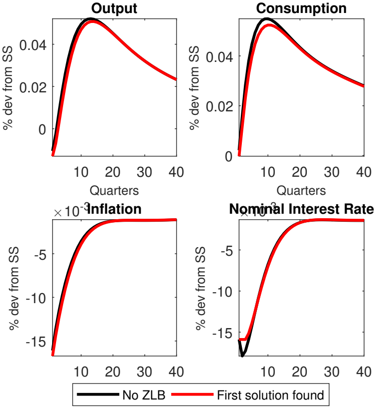
As well as choosing `omega`, there are other options that can help select between equilibria:

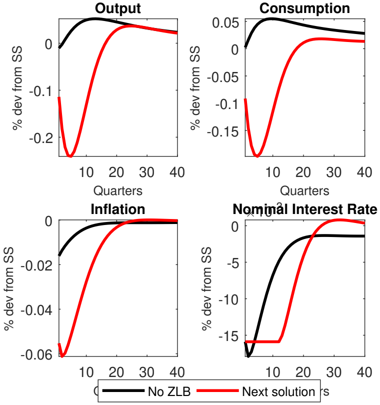
- `ReverseSearch`: by default DynareOBC searches for a solution that minimises time at the bound. This reverses and maximises time at the bound (subject to being less than `TimeToEscapeBounds`)
- `FullHorizon`: using this just solves the bounds problem at the longest horizon
- `SkipFirstSolutions=INTEGER`: when this is set > 0, dynareOBC will ignore the first `INTEGER` solutions found. If there were no other solutions, the last one will be used.
    - so if `ReverseSearch` is **not** used, dynareOBC will find a solution at the bound for longer. The opposite is true if `ReverseSearch` is used.

Example: Smets & Wouters (2003) - shock to β



Example: Smets & Wouters (2007) - shock combination





- Combination of shocks to all 7 stochastic processes - see code for detail.

References I

Holden, T. (2019), Existence, uniqueness and computation of solutions to dynamic models with occasionally binding constraints.

Lan, H. & Meyer-gohde, A. (2013), Pruning in Perturbation DSGE Models - Guidance from Nonlinear Moving Average Approximations Pruning in Perturbation DSGE Models.

Smets, F. & Wouters, R. (2003), 'An estimated dynamic stochastic general equilibrium model', *Journal of the European Economic Association* **1**(5), 1123–1175.

Smets, F. & Wouters, R. (2007), 'Shocks and frictions in us business cycles: A bayesian dsge approach', *American Economic Review* **97**(3), 586–606.
URL: *http://www.aeaweb.org/articles?id=10.1257/aer.97.3.586*