2.2. Deterministic and Extended Path Simulations

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.

Reminder of the problem

- We have a DSGE model in the form

$$\mathbb{E}_t f\left(x_{t+1}, x_t, x_{t-1}, u_t\right) = 0 \tag{1}$$

  which includes a non-differentiable function representing inequality constraints
- $f(\cdot)$ is a *known* function and the solution implies the unknown policy:

$$x_t = g\left(x_{t-1}, u_t\right) \tag{2}$$

- Projection methods can be used if the model is small, but for larger models, perturbation (e.g. using dynare) will not capture the inequality constraints.
- One option to simplify the problem is to remove uncertainty, so the model becomes:

$$f\left(x_{t+1}, x_t, x_{t-1}, u_t\right) = 0 \tag{3}$$

Perfect Foresight

Perfect-foresight simulation

To solve, we can stack the model over T periods

$$F(X) = \left\{ \begin{array}{c} f(x_0, x_1, x_2, z_1) \\ f(x_1, x_2, x_3, z_2) \\ \vdots \\ f(x_{T-1}, x_T, x_{T+1}, z_T) \end{array} \right\} = 0 \tag{4}$$

Initial and end conditions given ($x_0 = x_{T+1} = \bar{x}$)

- For an N equation model over T periods, there will be $N \times T$ unknowns and equations
- Can use a root-finding algorithm to find x to satisfy (4).

Perfect-foresight IRF example

We can write our example model

$$f\left(b\right) = \min\left\{\mu(b_{t+1}, b_t, b_{t-1}, z_t), (\underline{b} - b_t)\right\} = 0 \tag{5}$$

where:

$$\mu(b_{t+1}, b_t, b_{t-1}, z_t) = \frac{1}{c(b_t, b_{t-1}, z_t)} - r\beta\left[\frac{1}{c(b_{t+1}, b_t, z_{t+1})}\right] + \delta b_t \tag{6}$$

$$c(b_t, b_{t-1}, z_t) = \frac{1}{1+\chi}\left[\exp\left(z_t\right) + rb_{t-1} - b_t\right] \tag{7}$$

- We can substitute out c_t, h_t and μ, so are left with a single equation model (+ shock process)

Perfect-foresight IRF example in Matlab

See code: **/borrowing_constraints/VFI/soe_irf_fsolve.m**

- Solving this example in Matlab is straightforward
- We can even just use fsolve, setting up a function to return $F(b)$:

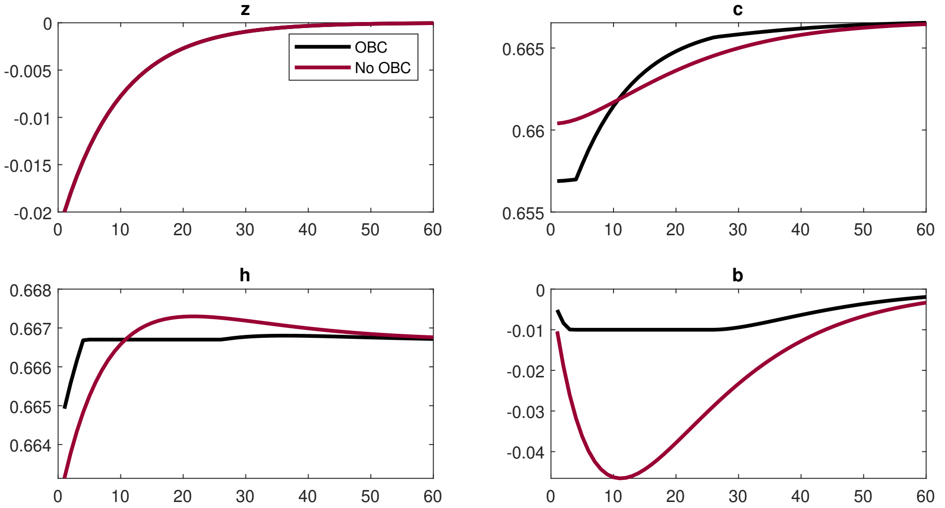
```
function F = model_f( b , z , ss , p )
    lag_b = [ ss.b ; b(1:end-1) ];
    c = ( exp( z ) + p.r .* lag_b - b ) ./ (1 + p.chi);
    lead_c = [ c(2:end) ; ss.c ];
    mu = (1./c) - p.betta * p.r * (1./lead_c) + p.delta .* b;
    F = min( mu , b-p.b_limit );
end
```

and

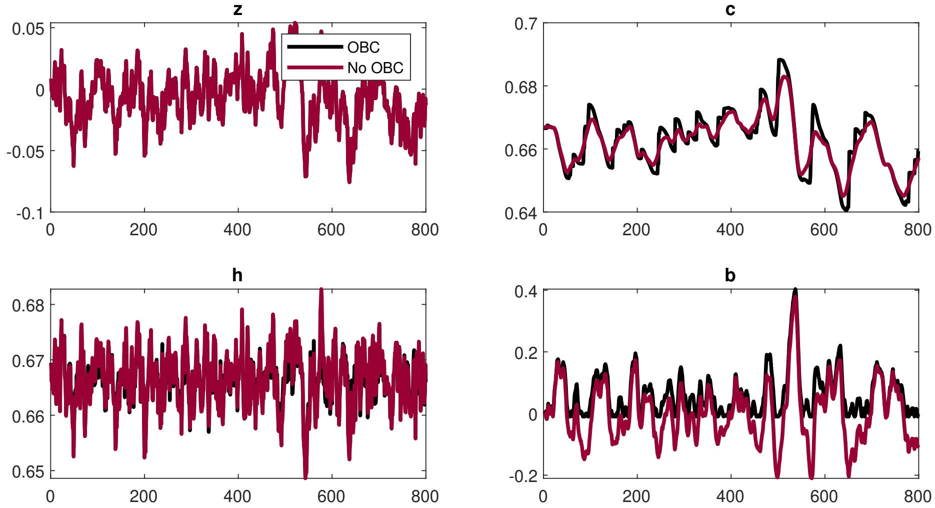
```
fun = @(b) model_f( b , z , ss , p );
b0 = ss.b * ones( T , 1 );
[b , ~ , flag , ~] = fsolve( fun , b0 , options );
```

- Note: the start and end values, $b_0 = b_{T+1} = \bar{b}$, the steady-state value.
- The same code can be used to compute simulated time-series (see **soe_ts_fsolve.m**)

Perfect-foresight IRF



Perfect-foresight Time-series



Perfect-foresight simulation in Dynare

See code: **/borrowing_constraints/VFI/soe_obc.mod**

- To use Dynare's perfect foresight solver, you must specify the shocks:

```
shocks;
    var epsz; periods 1;  values -2;
end;
```

- and replace the `stoch_simul` command with:

```
simul( periods=400 );
```

Perfect-foresight simulation in Dynare

See code: **/borrowing_constraints/VFI/soe_obc.mod**

- To use Dynare's perfect foresight solver, you must specify the shocks:

```
shocks;
    var epsz; periods 1;  values -2;
end;
```

- and replace the `stoch_simul` command with:

```
simul( periods=400 );
```

- Variety of algorithms to choose from - default is a Newton-type method.
  - This is an iterative procedure, which for a single variable would be written:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{8}$$

OBC considerations - LMMCP 1/2

- The presence of min/max operator can introduce a singularity into the Jacobian
  - We may fail to converge to a solution
- Dynare can improve on the default Newton method by treating it as a Levenberg-Marquardt mixed complementarity problem (LMMCP)
- This specifically deals with inequality constraints

OBC considerations - LMMCP 1/2

- The constraint must be given, replacing the min/max operator
  - i.e., the complimentary slackness condition

The equation in

```
model
    ...
    b = max( exp(z) + r*b - 1 / (betta*r*(lead_muc) + p.delta ) , b_limit);
    ...
end
```

is replaced with:

```
[mcp = 'b > -0.01'] b = exp(z) + r*b - 1 / (betta*r*(lead_muc) + p.delta );
```

- The LMMCP option is switched on in the `simul` command:

```
simul( periods=400 , lmmcp );
```

Perfect-foresight Dynare – comments

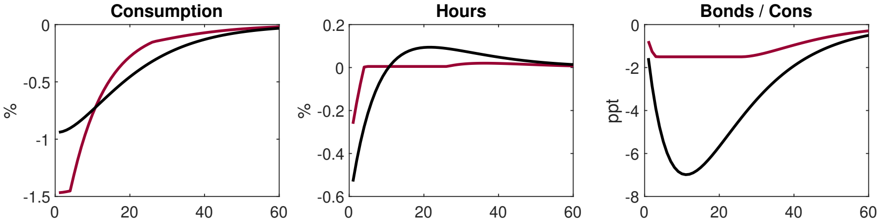
Performance:

- Robustness of Newton-type method improved recently via homotopy
  - Progressively increases shock scale, helping models which are difficult to solve
- The LMMCP can help with hard-to-solve problems
- Still the methods are slow and unreliable
- It is difficult to tell if non-convergence is to because there is no solution

Accuracy:

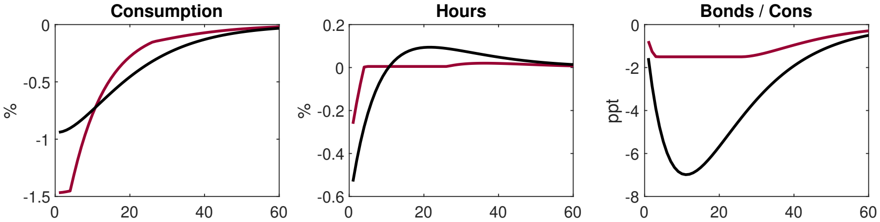
- Captures model non-linearities, including OBCs
- Perfect-foresight IRF has a natural interpretation
- Can also simulate response to *news* of a future shock, or transitions between long-run equilibria (absent uncertainty)
- Simulated time-series do not have a natural interpretation – perfect anticipation

Perfect foresight: IRF to large technology shock

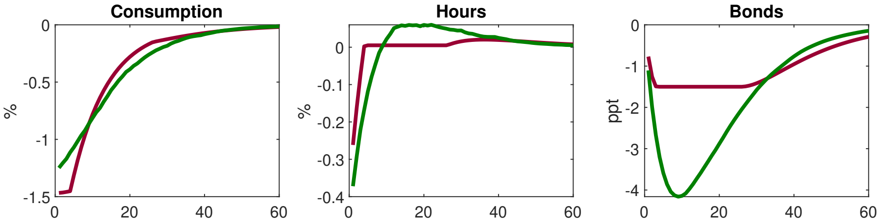


- No constraint, with OBC

Perfect foresight: IRF to large technology shock



- No constraint, with OBC



- Projection (devation from ergodic mean), perfect foresight (Newton method) (dev. from SS)

Extended Path

Extended path simulations

The previous approach assumes the perfect foresight of shocks

Alternatively we could assume firms and households are always surprised by shocks

- Every period they observe current and past shocks, and expect future shocks to equal zero

The underlying numerical problem is the same:

- Every period the perfect foresight simulation is solved to compute current decisions
- This is the Fair & Taylor (1983) extended-path method and is invoked in dynare using:

```
extended_path(order="0", periods="10000");
```

instead of `simul` or `stoch_simul`.

See code: **/extended-path/soe_obc.m**

Extended path simulations in Matlab

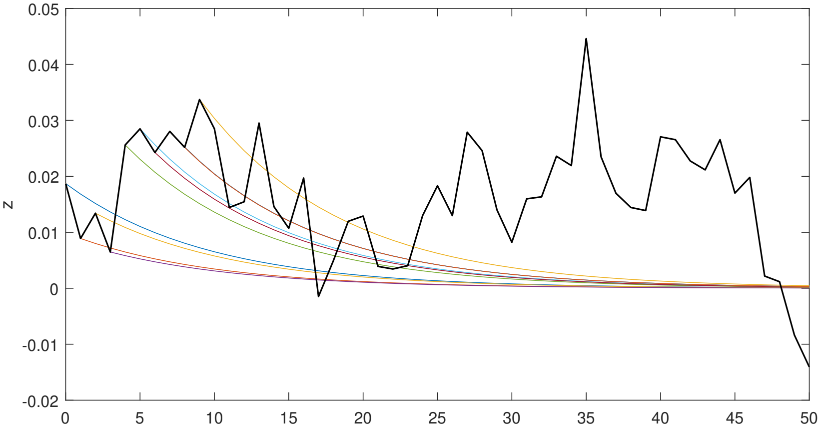
The extended path algorithm builds on the perfect-foresight solver.

- We can again easily demonstrate this in Matlab using fsolve
- see code: **/extended-path/soe_ts_ep.m**

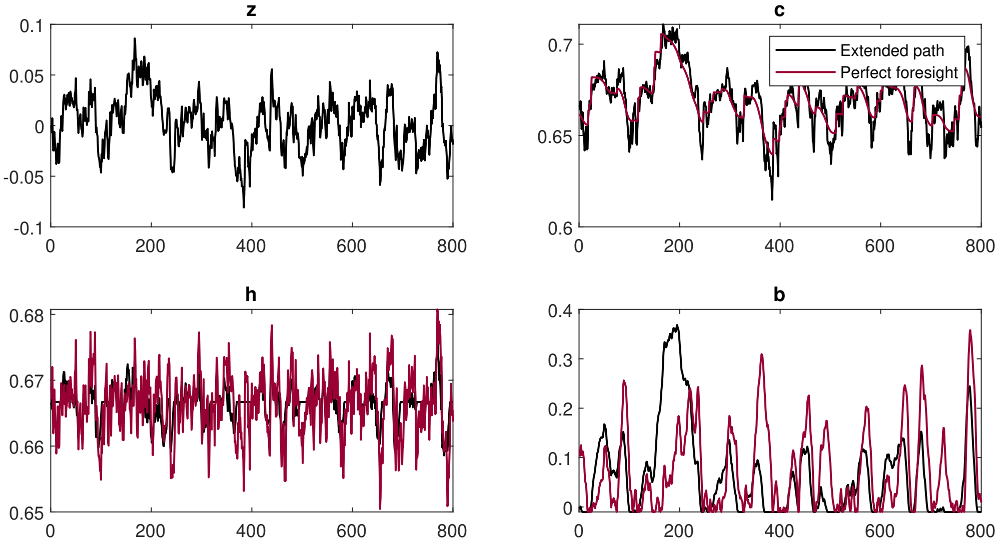
Extended path simulations in Matlab

The extended path algorithm builds on the perfect-foresight solver.

- We can again easily demonstrate this in Matlab using fsolve
- see code: **/extended-path/soe_ts_ep.m**
- Actual series for z (black), with expected path at different points in time.



Extended-path vs perfect foresight simulations



Stochastic Extended-Path Algorithm

Dynare has an adapted extended-path algorithm to incorporate the role of risk

The *stochastic* extended-path algorithm computes expectations every period up to a finite horizon

The **first** step is to draw a random path for all exogenous variables

Second, Dynare computes the corresponding path for the endogenous variables using the algorithm:

1. An integer k is chosen which is the number of periods for which expectations must be computed.
2. Beginning with expectations equal to the steady-state values $\mathbb{E}_s x_{s+r} = x$ for $r = 0, ..., k$, obtain a new set of expectations by solving the non-linear model dynamically. Every period the expectations are evaluated using a Gaussian quadrature.
3. Setting the new expectations as the starting values, this is repeated until convergence.
4. Repeat the above but increasing r by 1 and repeat this until convergence.

(Stochastic) Extended-Path in dynare

Shocks are specified as when using `stoch_simul`, but instead use:

```
extended_path(order="16", periods="10000");
```

(Stochastic) Extended-Path in dynare

Shocks are specified as when using `stoch_simul`, but instead use:

```
extended_path(order="16", periods="10000");
```

In practice, the extended-path method in dynare doesn't solve the full rational expectations model. The modeller specifies the maximum value of k for which to solve using the `order` option

- In this case, expectations are computed as if shocks could be non-zero for 16 more periods. After this horizon, the agents would believe that there would be no more future shocks.
- For accuracy, the order should be set to be the same magnitude as the decay of the model.
- dynare uses Gaussian quadrature to evaluate the expectations which scales exponentially in the number of shocks and the order (although not the number of states).
- In practice, solving with sufficient accuracy is infeasible even for very small models.
- The accuracy improvement seems small compared to the significant computational cost.

Simulated Moments

Projection results:

	Mean	**Standard deviation**	**Skewness**	
	Relative to no constraint	Relative to no constraint	Baseline	No constraint
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

All methods:

		Mean	**Standard deviation**	**Skewness**
		Relative to projection	Relative to projection	(projection)
VFI	Consumption	≈ 0	+0.02%	-0.23 (-0.22)
	Hours	≈ 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)

References I

Fair, R. C. & Taylor, J. B. (1983), 'Solution and Maximum Likelihood Estimation of Dynamic Nonlinear Rational Expectations Models', *Econometrica* **51**(4), 1169–1185.