Discussion Paper Series – CRC TR 224

Discussion Paper No. 253
Project C 01

Efficient Solution and Computation of Models With Occasionally Binding Constraints

Gregor Boehl [1]

January 2021

[1] University of Bonn

Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through CRC TR 224 is gratefully acknowledged.

Efficient Solution and Computation of Models with Occasionally Binding Constraints

Latest version: http://gregorboehl.com/live/obc_boehl.pdf

Gregor Boehl

University of Bonn

January 11, 2021

Abstract

Structural macroeconometric analysis and new HANK-type models with extremely high dimensionality require fast and robust methods to efficiently deal with occasionally binding constraints (OBCs), especially since major developed economies have again hit the zero lower bound on nominal interest rates. This paper shows that a linear dynamic rational expectations system with OBCs, depending on the expected duration of the constraint, can be represented in closed form. Combined with a set of simple equilibrium conditions, this can be exploited to avoid matrix inversions and simulations at runtime for significant gains in computational speed. An efficient implementation is provided in Python programming language. Benchmarking results show that for medium-scale models with an OBC, more than 150,000 state vectors can be evaluated per second. This is an improvement of more than three orders of magnitude over existing alternatives. Even state evaluations of large HANK-type models with almost 1000 endogenous variables require only 0.1 ms.

Keywords: Occasionally Binding Constraints, Effective Lower Bound, Computational Methods

1 Introduction

Occasionally binding constraints have become an important part of economic modelling, especially since western central banks see themselves (again) constraint by the so-called zero lower bound (ZLB) of the nominal interest rate. A binding ZLB constraint

* *Email:* gboehl@uni-bonn.de. *Address:* University of Bonn, Adenauerallee 24–42, 53113 Bonn, Germany. I am grateful to Hans-Martin von Gaudecker, Alexander Meyer-Gohde, Kenneth Judd, Alexander Richter, Felix Strobel and participants of the 2018 Stanford MMCI Conference, the 2018 EEA Annual Congress and the 2018 VfS Jahrestagung for discussions and helpful comments on the contents of this paper. Part of the research leading to the results in this paper has received financial support from the Alfred P. Sloan Foundation under the grant agreement G-2016-7176 for the Macroeconomic Model Comparison Initiative (MMCI) at the Institute for Monetary and Financial Stability. I gratefully acknowledge financial support by the DFG under CRC-TR 224 (project C01) and under project number 441540692.

poses a major problem for a quantitative-structural analysis: Linear solution methods do no work in the presence of a nonlinearity such as the ZLB and existing alternatives tend to be computationally demanding. The urge to study macroeconomic questions related to the Great Recession and the current Covid-19 crisis in a quantitative-structural framework requires algorithms that are not only accurate, but that are also robust, fast, and computationally efficient.

A particularly important application where efficient and fast methods for occasionally binding constraints (OBCs) are needed is the Bayesian estimation of macroeconomic models. In the US, for example, the ZLB episodes from 2008 to 2015 and again from 2020 onwards will be part of the time series used in structural econometrics for the next decades to come. This calls for methods that are fast enough to explicitly account for the endogenously binding ZLB during the estimation procedure. Further, an active literature develops large-scale New Keynesian models that feature heterogeneous agents and a larger number of idiosyncratic states (HANK models, see e.g. Reiter, 2009; Kaplan et al., 2018; Bayer et al., 2020). It is straightforward that researchers wish consider OBCs such as the ZLB in this framework. However, the fact that these models tend to have up to 1000 dimensions (and more) simply overburdens the currently available methods.

This paper aims to fill this gap, and significantly expands the set of possible applications of models with OBCs. Given any dynamic general equilibrium model with one or several OBCs, I develop a closed-form state-space representation for the complete expected trajectory of the endogenous variables as a function of the expected duration at the ZLB (the "ZLB spell duration"). Furthermore, I provide the necessary conditions of a rational expectations equilibrium for a set of ZLB spell durations given the state of the economy. Given these two ingredients, the expected ZLB spell durations can be found via a simple iterative scheme. Using the closed-form solution together with the equilibrium conditions allows to check for a model equilibrium instantaneously instead of simulating a complete anticipated equilibrium path for a given ZLB spell. This increases the computational speed of the algorithm substantially.

There exist several solution concepts for DSGE models with OBCs.[1] The currently most frequently used algorithm is OccBin, which was introduced in Guerrieri and Iacoviello (2015). The authors propose a recursive representation of the solution given the state of the economy and a set of spell durations. They propose a Newton-like method to iteratively find the set of spell durations. The relatively high computational costs of their approach stems from the fact that for each guess of the spell durations, the Newton-like method requires the complete simulation of the anticipated trajectory. This requires repeated matrix inversions at runtime, which are computationally expensive. While my method shares some features with their algorithm (and, given uniqueness, will return an identical solution), it has a considerable advantage in terms of computation speed and, hence, is also well suited for parameter inference as well as for very high dimensional models.

The method presented in Holden (2016, 2017) is robust and accurate, especially with regard to proper equilibrium selection. It is however not optimized with regard to computational speed as each guess of the spell durations requires at least one matrix inversion. As the outcomes of the method presented in this paper is, given uniqueness, identical

[1] Early work with nonlinear models includes e.g. Coenen and Wieland (2003, 2004); Coenen et al. (2004).

to those of the work cited above, I refer to the papers cited above for comparisons with other nonlinear methods such as policy function iteration.

I provide speed benchmarks of the suggested method for two applications that both incorporate an endogenously binding ZLB: I first benchmark against a standard medium-scale New-Keynesian model in the style of Smets and Wouters (2007) and then against the HANK model of Bayer et al. (2020). For the medium scale model, my method performs one evaluation of a state vector in less than seven microseconds ($\frac{7}{1000}$ ms) or more than 150,000 evaluations per second. For the HANK-type model with almost 1000 endogenous variables these are still about 0.1 ms per evaluation or 8219 evaluations per second. In comparison, OccBin takes in average 0.01 seconds per evaluation, corresponding to 95.7 evaluations per second. This implies a speed advantage of the method presented in this work by a factor of 1500.

A reference implementation of the proposed solution method, together with a parser and related econometric tools, is implemented in the `pydsge` package.[2] Computational advantage also depends on efficient implementation. The implementation is written in the freely available language Python.[3] The rest of this paper is structured as follows. Section 2 develops the solution method and discusses details regarding the implementation. In Section 3 I provide speed benchmarks. Section 4 concludes.

2 Method

Assume a linear rational expectations model that is subject to n_c OBCs.[4] Inspired by Uhlig et al. (1995); Binder and Pesaran (1995); Villemot et al. (2011), the linearized first order conditions of this system can be represented by

$$E_t\left[\mathfrak{A}z_{t+1} + \mathfrak{B}z_t + \mathfrak{C}z_{t-1} + \mathfrak{D}\epsilon_t + \sum_j^{n_c} \mathfrak{h}_j \max\left\{\mathfrak{a}_j z_{t+1} + \mathfrak{b}_j z_t + \mathfrak{c}_j z_{t-1} + \mathfrak{d}_j \epsilon_t, \bar{r}_j\right\}\right] = 0, \tag{1}$$

where y_t is the n-dimensional vector of all model variables and ϵ_t the n_ϵ-dimensional vector of iid. exogenous shocks. $\mathfrak{A}$, $\mathfrak{B}$ and $\mathfrak{C}$ are generic $n \times n$ system matrices whereas, $\mathfrak{D}$ is a $n \times n_\epsilon$ matrix. Equation (1) can elegantly be reduced to

$$E_t\left[Ay_{t+1} + By_t + Cy_{t-1} + \sum_j^{n_c} h_j \max\left\{a_j y_{t+1} + b_j y_t + c_j y_{t-1}, \bar{r}_j\right\}\right] = 0, \tag{2}$$

with $y_t = (z_t, \epsilon_{t+1})$, $A = \begin{vmatrix}\mathfrak{A} & 0\\ 0 & 0\end{vmatrix}$, $B = \begin{vmatrix}\mathfrak{B} & 0\\ 0 & I\end{vmatrix}$, $C = \begin{vmatrix}\mathfrak{C} & \mathfrak{D}\\ 0 & 0\end{vmatrix}$ and $a_j = (\mathfrak{a}_j, 0)$, $b_j = (\mathfrak{b}_j, 0)$, $c_j = (\mathfrak{c}_j, \mathfrak{d}_j)$, $h_j = (\mathfrak{h}_j, 0)$ for each constraint j. Each variable $r_{j,t}$ $\forall j \in 1, 2, \cdots, n_c$ is

[2] The package is available at `https://github.com/gboehl/pydsge`.

[3] Python, combined with the just-in-time compiler `Numba`, can provide speed benchmarks that are en-par with compiled languages such as C or Fortran while comprising the advantages of a high-level programming language.

[4] See e.g. Schmitt-Grohé and Uribe (2004) for pertubation methods to obtain a linear representation from the nonlinear model.

subject to an occasionally binding constraint represented by a set of vectors $\{a_j, b_j, c_j\}$ and a minimum value of $r_{j,t}$ denoted $\bar{r}_j$. $h_j \in \mathbb{R}^n$ contains the coefficients of $r_{j,t}$ in each equation of the linear system.

Also define the system in which all constraints are slack (the *unconstrained system*) as

$$\hat{A}y_{t+1} + \hat{B}y_t + \hat{C}y_{t-1} = 0, \tag{3}$$

with

$$\hat{A} = A + \sum_j^{n_c} h_j \otimes a_j \tag{4}$$

$$\hat{B} = B + \sum_j^{n_c} h_j \otimes b_j, \tag{5}$$

$$\hat{C} = C + \sum_j^{n_c} h_j \otimes c_j. \tag{6}$$

Let me borrow the Assumptions 1 and 2 from Holden (2016) (or, alternatively, from Rendahl (2017)) and restate them here:

Assumption 1. *For any given $y_0 \in \mathbb{R}^n$, Equation* (3) *has a unique solution, which takes the form $y_t = Fy_{t-1}$ for $t \in \mathbb{N}^+$, where $F = -(\hat{B} + \hat{A}F)^{-1}\hat{C}$, and where all the eigenvalues of F are weakly inside the unit circle.*

Assume further, to imply that all the eigenvalues of F are strictly inside the unit circle, that:

Assumption 2.

$$\det\left(\hat{A} + \hat{B} + \hat{C}\right) \neq 0, \tag{7}$$

Equation (2) can be cast in the form (see e.g. Klein (2000) or Villemot et al. (2011)):

$$PE_t x_{t+1} = Mx_t + \sum_j^{n_c} h_j \max\left\{p_j E_t x_{t+1} + m_j x_t, \bar{r}_j\right\}, \tag{8}$$

with $x_t = \begin{vmatrix} s_{t-1} \\ c_t \end{vmatrix}$, where c_t are the forward looking variables (controls) and s_{t-1} are the states updated by the time-t shocks as above.

Denote the system in which *all* constraints are slack (the *unconstrained system*) as

$$\hat{P}E_t x_{t+1} = \hat{M}x_t, \tag{9}$$

with

$$\hat{P} = \left(P + \sum_j^{n_c} h_j \otimes p_j\right) \qquad \text{and} \qquad \hat{M} = \left(M + \sum_j^{n_c} h_j \otimes m_j\right). \tag{10}$$

For the larger part of this section I will assume that P and $\hat{P}$ are invertible as this simplifies display. I will first outline the solution method for only one constraint, taking the durations for which the constraint holds as given. Then I present a simple iteration scheme to find the expected durations. The generalization to many occasionally binding constraints is then straightforward and left to the reader.

2.1 A representation in closed form

To fix ideas, let us focus on the case with only one constraint, which is given by (h_0, p_0, m_0). Under the assumption that P and $\hat{P}$ are invertible System (9) can be rewritten as

$$E_t x_{t+1} = \begin{cases} \hat{N} x_t & \forall\, p_0 E_t x_{t+1} + m_0 x_t - \bar{r} \geq 0 \\ \\ N x_t + q_0 \bar{r} & \forall\, p_0 E_t x_{t+1} + m_0 x_t - \bar{r} < 0, \end{cases} \tag{11}$$

with $\hat{N} = \hat{P}^{-1}\hat{M}$, $N = P^{-1}N$ and $q_0 = P^{-1}h_0$.

We first assume that if the constraint binds, it always binds already in the current period t (hence there is *no transition* to the constraint). Let k_t be the expected duration of the spell to the constraint in period t. Denote a rational expectations solution to (9) given k_t and the state variables s_{t-1} as the function f such that

$$c_t = f(k_t, s_{t-1}). \tag{12}$$

In slight abuse of notation, I will occasionally use k and $f(k)$ as shorthand where the states s_{t-1} and the time-t subscripts are understood. Also, denote as $x_t|k$ the variables vector conditional on expecting the constraint to hold for k periods, which is trivial to find once f is known.

For the unconstrained system $\hat{N}$, c_t can be found using familiar methods like the QZ-decomposition as suggested by Klein (2000). Denote this (linear) solution for c_t by the matrix Ω:

$$c_t = \Omega s_{t-1} \quad \forall\, p_0 E_t x_{t+1} + m_0 x_t > \bar{r} \tag{13}$$

For $\Psi = \begin{vmatrix} -\Omega & I \end{vmatrix}$, Equation (13) implies that

$$E_t \left\{ \Psi \begin{vmatrix} s_{t+k} \\ c_{t+k+1} \end{vmatrix} \right\} = 0 \quad \forall\, p_0 E_t x_{t+k+1} + m_0 x_{t+k} \geq \bar{r}, \tag{14}$$

i.e. for every future period $t+k$ in which the system is expected to be unconstrained.

Now assume that the constraint binds at time t and will continue to do so until period $t+k$. Iterating System (11) forward yields

$$E_t \left\{ \begin{vmatrix} s_{t+k-1} \\ c_{t+k} \end{vmatrix} \right\} = N^k x_t + (I-N)^{-1}(I-N^k) q_0 \bar{r}, \tag{15}$$

where $(I-N)^{-1}(I-N^k) = \sum_{i=0}^{k-1} N^i$ is the transformation for a geometric series of matrices. Finally, we can combine Equations (14) and (15) to find f, i.e. a solution of the controls c_t in terms of the state variables s_{t-1} given k:

$$f(k, s_{t-1}) = \left(c_t : \Psi N^k \begin{vmatrix} s_{t-1} \\ c_t \end{vmatrix} = -\Psi (I-N)^{-1}(I-N^k) q_0 \bar{r} \right). \tag{16}$$

Since q_0 is a vector of constants, the whole RHS of Equation (16) is known and solving for c_t is simple.

Let us now relax the assumption that a shock triggers the constraint to hold immediately in time t. This case is in particular relevant for models with persistent endogenous state variables. Take Equation (16) as the starting point and allow for a number of periods l in the unconstrained system $\hat{N}$ *until* the system is at the constraint:

$$f(k, l, s_{t-1}) = \left(c_t : \Psi N^k \hat{N}^l \begin{vmatrix} s_{t-1} \\ c_t \end{vmatrix} = -\Psi (I - N)^{-1} (I - N^k) q_0 \bar{r} \right). \tag{17}$$

Using Equation (17) and Equation (15) augmented by $\hat{N}^l$, we can express the expectations on the variable vector conditional on (l_t, k_t) of the economy in period j, $E_t x_j|(l_t, k_t)$, as a function F with

$$\begin{aligned} E_t x_j|(l_t, k_t) = F_j(l_t, k_t, s_{t-1}) =& N^{\max\{s-l,0\}} \hat{N}^{\min\{l,s\}} \begin{vmatrix} f(l_t, k_t, s_{t-1}) \\ s_{t-1} \end{vmatrix} \\ &+ (I - N)^{-1} (I - N^{\max\{j-l,0\}}) q_0 \bar{r}. \end{aligned} \tag{18}$$

Note that $F_1(0, 0, j_{t-1})$ is the generic solution to the unconstrained system.

2.2 Solving for the spell durations (l, k)

Again, let us first consider the simpler case in which we assume that any shock that causes the constraint to bind, it will cause it to bind immediately in time t (the *non-transitory* case). The following proposition summarizes the conditions for (x_t, s_{t-1}, k) to be a rational expectations equilibrium:

Proposition 1 (non-transitory equilibrium)**.** *Assuming non-transition, a number of expected periods k^* at the constraint is part of a rational expectations equilibrium iff*

$$p_0 E_t[x_{t+k+1}|k^*] + m_0 x_{t+k}|k^* \geq \bar{r} > p_0 E_t[x_{t+k+1}|k] + m_0 x_{t+k}|k \tag{19}$$

for all $k^* > k \geq 0$, hence if in expectations the system is constrained for exactly k^* periods.

Let us proceed to the practically relevant case where agents may expect the unconstrained system to prevail for some transition time before the constraint binds for k periods. Using Equation (18), Proposition 2 summarizes the respective equilibrium conditions.

Proposition 2 (transitory equilibrium)**.** *A pair (l^*, k^*) is part of a rational expectations equilibrium iff*

$$p_0 E_t[x_{t+j+1}|(l^*, k^*)] + m_0 x_{t+j}|(l^*, k^*) \geq \bar{r} \quad \forall j < l^* \wedge j \geq k^* + l^* \tag{20}$$

and

$$p_0 E_t[x_{t+j+1}|(l^*, k^*)] + m_0 x_{t+j}|(l^*, k^*) < \bar{r} \quad \forall l^* \leq j < k^* + l^*. \tag{21}$$

In other words, (l, k) are part of an equilibrium, if in expectations, the constraint starts binding exactly in period $t + l$ and ends to bind exactly in period $t + l + k$.

Unfortunately there is no closed form solution for (l^*, k^*) given s_{t-1}. A set of (l^*, k^*) that satisfies Proposition 2 must be found using an iterative scheme. As this problem requires an iterative scheme on an integer domain, a theoretical assessment is at least difficult because most theoretical work on similar algorithms deals with real valued functions. Given those limits, some insights regarding the existence and uniqueness of such solutions are provided by Holden (2017). An integer based Newton-like method as employed by Guerrieri and Iacoviello (2015) is not efficient because it requires the evaluation of the complete anticipated trajectory of x_t and furthermore can not draw on the nice convergence properties of the standard real-valued Newton method. This makes their method diffucult to use if OBC problems are not well behaved. Given s_{t-1}, an equilibrium satisfying conditions (20) and (21) will generally be the same as in Holden (2017). The crucial advantage of the formulation provided here is the closed form expression of $E_t[x_{t+j}|(l, k)]$.

An optimal iterative scheme can be hand-tailored to the problem. For the purpose of the estimation of large-scale DSGE models, in which the constraint is the zero lower bound on nominal interest rates, Boehl and Strobel (2020), Boehl et al. (2020) and Boehl and Lieberknecht (2021) use the following iterative scheme:

```
l, k = 0, 0
for l in range(l_max):
    if b F(l, 0, l, v) - r_bar < 0:
        # constraint binds: interrupt loop
        break
    if l is l_max - 1:
        # return that l=k=0 is an equilibrium
        return 0, 0
...
```

Hence, if the constraint is not reached within `l_max` periods ahead in the future, exit. Otherwise assume $k > 0$ and iterate over l and k until the equilibrium conditions in (19), (20) and (21) are satisfied:

```
...
for l in range(l_max):
    for k in range(1, k_max):
        if l:
            if b F(l, k, 0, v) - r_bar < 0:
                # skip inner loop to next k
                continue
            if b F(l, k, l-1, v) - r_bar < 0:
                # skip inner loop ...
                continue
        if b F(l, k, k+l, v) - r_bar < 0:
            continue
        if b F(l, k, l, v) - r_bar > 0:
            continue
        if b F(l, k, k+l-1, v) - r_bar > 0:
            continue
        # if we made it here, this must be an equilibrium
```

```
        return l, k
# if the loop went though without finding an equilibrium, throw a warning or set
    error flag
flag = True
warn('No equilibrium exists!11')
```

This scheme is very efficient for the specific problem because for more than 50% of the data points used the ZLB is not binding and the method will already exit in the first loop.[5] If it does not exit, then for post-2008 data points it is predominantly the case that the ZLB already is binding. In this case $l^* = 0$ and only k^* is to be determined. As, according to the Primary Dealer Survey, most market participants expected the ZLB to be binding for about eight quarters, the procedure will on average need 8 guesses until an equilibrium is found. If at all, l^* will only be positive in 2008Q3 when the economy is not yet at the ZLB, but the shocks originating from the Subprime Mortgage Crisis have triggered the ZLB to be expected to bind in the very near future.

While the above procedure is tailored to work most efficiently in the context of estimating DSGE models with the ZLB, it is generic and applicable to any sort of constraint. The resulting transition function is linear for the region where the constraint does not bind and (increasingly) nonlinear when it binds. Note that this algorithm includes an active assumption on equilibrium selection: if for a s_{t-1} several sets of (l^*, k^*) exist that satisfy Proposition 2, the set with the lowest l^* is chosen.

2.3 Preprocessing and the case of singular P or $\hat{P}$

Let us now turn to the practically more relevant case where we relax the assumption that P and $\hat{P}$ are invertible. Use the QL decomposition on $M = QL$ and on $\hat{M} = \hat{Q}\hat{L}$. Let n_c be the number of control variables. Premultiplication of (9) by Q ($\hat{Q}$, respectively), and premultiplication of the n_c lower rows by the inverse of the lower-right $n_c \times n_c$ submatrix of L ($\hat{L}$) leads to:[6]

$$\begin{vmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{21} & \hat{P}_{22} \end{vmatrix} \begin{vmatrix} s_t \\ E_t c_{t+1} \end{vmatrix} = \begin{vmatrix} \hat{M}_{11} & 0 \\ \hat{M}_{21} & I \end{vmatrix} \begin{vmatrix} s_{t-1} \\ c_t \end{vmatrix} \qquad \forall\ p_0 E_t x_{t+1} + m_0 x_t - \bar{r} \geq 0, \tag{22}$$

$$\begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix} \begin{vmatrix} s_t \\ E_t c_{t+1} \end{vmatrix} = \begin{vmatrix} M_{11} & 0 \\ M_{21} & I \end{vmatrix} \begin{vmatrix} s_{t-1} \\ c_t \end{vmatrix} + \begin{vmatrix} h_{0,s} \\ h_{0,c} \end{vmatrix} \bar{r} \qquad \forall\ p_0 E_t x_{t+1} + m_0 x_t - \bar{r} < 0. \tag{23}$$

Additionally to the function f introduced in Equation (12), define g as

$$s_t = g(l, k, s_{t-1}), \tag{24}$$

and note that

$$F_0(l, k, s_{t-1}) = \begin{vmatrix} g(l, k, s_{t-1}) \\ f(l, k, s_{t-1}) \end{vmatrix}. \tag{25}$$

[5]For example, Boehl and Strobel (2020) use US data from 1998 to 2020, which contains the ZLB period from 2008Q4 to 2015Q4.

[6]The fact that the lower-right submatrices of L and $\hat{L}$ are nonsingular follows simply from the fact that controls are defined on their future values.

Again, both functions are linear in s_{t-1} given (l,k) and take the form

$$f(l,k,s_{t-1}) = \tilde{f}(l,k)s_{t-1} + \bar{f}(l,k)\bar{r}, \tag{26}$$
$$g(l,k,s_{t-1}) = \tilde{g}(l,k)s_{t-1} + \bar{g}(l,k)\bar{r}, \tag{27}$$

where $\tilde{f}(0,0)$ and $\tilde{g}(0,0)$ are found using any solution routine for linear systems, and $\bar{f}(0,0) = \bar{g}(0,0) = \overrightarrow{0}$.

For $k > 0$ we can then express these functions recursively as

$$g(l,k,s_{t-1}) = \begin{cases} \left(P_{11} + P_{12}\tilde{f}(l-1,k)\right)^{-1}\left(N_{11}s_{t-1} - P_{12}\bar{f}(l-1,k)\right) & \text{if } l > 0, \\ \left(\hat{P}_{11} + \hat{P}_{12}\tilde{f}(0,k-1)\right)^{-1}\left(\hat{N}_{11}s_{t-1} + h_{0,s} - \hat{P}_{12}\bar{f}(0,k-1)\right) & \text{if } l = 0, \end{cases} \tag{28}$$

$$f(l,k,s_{t-1}) = \begin{cases} \left(P_{21} + P_{22}\tilde{f}(l-1,k)\right)g(l,k,s_{t-1}) - M_{21}s_{t-1} + P_{22}\bar{f}(l-1,k) & \text{if } l > 0, \\ \left(\hat{P}_{21} + \hat{P}_{22}\tilde{f}(0,k-1)\right)g(l,k,s_{t-1}) - \hat{M}_{21}s_{t-1} + \hat{P}_{22}\bar{f}(0,k-1) - h_{0,c} & \text{if } l = 0. \end{cases} \tag{29}$$

As these expressions involve an inversion of a matrix of the same dimensionality of the state space, it is efficient to preprocess all functions within a reasonable range `l_max` and `k_max` and store the result for later use. In the same run, $p_0 E_t[x_{j+1}|(l,k)] + m_0 x_j|(l,k) = p_0 F_{j+1}(l,k,s_{t-1}) + m_0 F_{j+1}(l,k,s_{t-1})$ can be pre-processed and stored for efficient checking of the conditions in Proposition 2. This is a $(1 \times n)$ vector and a scalar for each combination of (l,k,s) under consideration. Checking each condition in Proposition 2 then only requires a dot-vector multiplication and a scalar addition.

3 Benchmarking Compuation Speed

Processing and calculation speed are key aspects of the design of the algorithm introduced in the previous section. This section presents benchmarks. For this purpose I consider two models with the ZLB on the nominal interest rate as an occasionally binding constraint. The first model is the medium-scale model of Smets and Wouters (2007), calibrated to the posterior mode from Boehl and Strobel (2020), where the model is estimated to US data from 1998 to 2020. The binding ZLB from 2008 to 2015 is explicitly modeled as an OBC and a nonlinear filter is used for likelihood inference. The inclusion of the ZLB in the estimation procedure, apart from its obvious appeal for economic analysis, has the further advantage that regions of the parameter space in which ZLB solutions are irregular are not included in the posterior distribution as they necessarily exhibit a lower likelihood. The second model is the Heterogeneous Agent New Keynesian (HANK) model of Bayer et al. (2020) which is estimated from 1954 to 2019, but without explicitly accounting for and endogenously binding ZLB during the estimation.[7] While the first example is chosen because it is the workhouse model of modern monetary economics, the second model is chosen because of its high dimensionality with 929 endogenous variables.

[7]The system matrices were kindly provided by the authors.

Additionally, I provide benchmarks for OccBin (c.f. Guerrieri and Iacoviello, 2015) which is used frequently in applied work and implemented in Dynare (see Adjemian et al., 2011). For the benchmarks provided below I use the implementation from Cuba-Borda et al. (2019) which goes beyond the standard implementation. In particular, it avoids solving the model and preprocessing the system matrices for every new state. These steps together are accountable for about 98% of OccBins computation time.[8] I do not benchmark the HANK model as the Dynare preprocessor seems to have problems with handling matrices of this size.

For each exercise I draw 1,000,000 state vectors from a multivariate normal distribution with zero mean and covariance $\Sigma = 10\mathbf{I}_n$, where n is the number of states. Each sample is passed trough the nonlinear transition function and grouped according to its calculated expected ZLB duration. I set `l_max`=3 and `k_max`=30 to cover most cases. If within this range no ZLB equilibrium is found, the sample counts as "No ZLB solution". Note that there are many other reasons why an equilibrium can not be found and a sample may count in this category, e.g. the reversals documented by Carlstrom et al. (2015). For OccBin, `nperiods` is also set to 30. Although this causes a small number of samples to not converge, it decreases computation time in favor of Occbin. For the method from Section 2, the overall computation times are 155675 draws per second (6.426 seconds in total) for the model of Smets and Wouters (2007) and 8219 draws per second (121.67 seconds in total) for the HANK model. OccBin performed 95.7 draws per second for the medium-scale model, whereas total computation took 174.13 minutes.

	mean	std	% of samples
$k^* = 0$	5.435e-06	1.262e-05	48.66%
$k^* \in (1, 5)$	6.850e-06	4.675e-06	5.79%
$k^* \in (6, 10)$	9.354e-06	6.214e-06	4.34%
$k^* \in (11, 15)$	8.323e-06	4.058e-06	8.33%
$k^* \in (16, 20)$	6.975e-06	1.386e-06	19.99%
$k^* > 20$	6.894e-06	6.471e-07	12.89%
$l^* > 0 \mid k^* > 0$	1.410e-05	5.777e-06	5.69%
No ZLB solution	3.784e-05	2.307e-06	0.00%
Total	6.424e-06	9.137e-06	100.00%

Table 1: Speed benchmark of the method from Section 2 for the medium-scale model of (Boehl and Strobel, 2020) with 57 variables.
Note: The table shows the execution times per state vector in seconds. "No ZLB solution" collects draws for which no solution was found within a maximum of `l_max` and `k_max` periods ahead (two draws in total).

Tables 1 and 2 present the results of the benchmarking exercise for the method from Section 2. For the medium scale model, in about 50% of the samples the ZLB is not binding. For these cases the calculation takes the least time because the algorithm only has to confirm that the ZLB is not binding within the first `l_max` periods. Calculation

[8]When benchmarking against the original implementation of Guerrieri and Iacoviello (2015), the method presented in Section 2 performs more than 10,000 times faster. For the benchmarks of OccBinn, Matlab version R2019a is used with Dynare 4.6.1.

time increases with expected ZLB durations, which is an anticipated result given that more guesses are needed. The incremental increase in computation time for higher k^* is, however, rather marginal. Samples for which no solution could be found take almost four times longer than samples in which the ZLB does not bind, which is due to the fact that for these samples all possible combinations of (l^*, k^*) have to be ruled out. While for the medium-scale model these are only two samples in total, the HANK model exhibits a more complicated structure which reflects in a higher rate of samples with no solution of about 11%.

	mean	std	% of samples
$k^* = 0$	8.878e-05	9.486e-06	42.95%
$k^* \in (1, 5)$	1.470e-04	1.991e-05	10.04%
$k^* \in (6, 10)$	1.493e-04	1.770e-05	11.42%
$k^* \in (11, 15)$	1.508e-04	1.550e-05	8.52%
$k^* \in (16, 20)$	1.544e-04	3.577e-05	4.95%
$k^* > 20$	1.549e-04	1.429e-05	10.49%
$l^* > 0 \vert k^* > 0$	1.765e-04	2.104e-05	1.78%
No ZLB solution	1.290e-04	2.874e-05	11.61%
Total	1.217e-04	3.433e-05	100.00%

Table 2: Speed benchmark of the method from Section 2 for the HANK model of (Bayer et al., 2020) with 929 variables.
Note: The table shows the execution times per state vector in seconds. "No ZLB solution" collects draws for which no solution was found for a maximum of `l_max` and `k_max` periods ahead.

The comparison of the two tables also documents that the increase in the number of variables reflects less than one-to-one in calculation times. Recall that the first phase of the algorithm – finding (l^*, k^*) – only requires dot multiplications. In vectorized code such calculations generally scale disproportionate relative to the size of the vectors due to the relative reduction of computational fixed costs. The actual execution of $f(l^*, k^*)$ and $g(l^*, k^*)$, ignoring the additive component, has a maximal complexity of $O(n^3)$ and is likely to be faster.

Finally, 3 shows speed benchmarks for OccBin. The exercise reveals that the method from Section 2 performs more than 1500 times faster than OccBin. Overall, the percentages of draws in each bin are very similar to the percentages in Table 1. This confirms that indeed both methods find the same solution, if it is unique. Further, while for the method presented here a draw with an higher k^* does not seem to bear higher computational costs, computation times for OccBin increase with k^*. However, the Newton-like method of Occbin seems to require relatively less time to find solutions for draws with $l > 0$, while such draws are relatively more expensive for the method presented here. Lastly, for Occbin 0.12% of all draws no solution is found. While this number is already very low, it can be squeezed down to (almost) zero by setting `nperiods` to 100.

	mean	std	% of samples
$k^* = 0$	0.004080	0.000398	50.39%
$k^* \in (1, 5)$	0.008001	0.002042	4.59%
$k^* \in (6, 10)$	0.012571	0.002395	4.33%
$k^* \in (11, 15)$	0.015785	0.002302	8.56%
$k^* \in (16, 20)$	0.018874	0.002732	20.77%
$k^* > 20$	0.019556	0.003229	11.25%
$l^* > 0 \vert k^* > 0$	0.014800	0.003345	5.96%
No ZLB solution	0.009100	0.005291	0.12%
Total	0.010449	0.007152	100.00%

Table 3: Speed benchmark of OccBin19 (Guerrieri and Iacoviello, 2015; Cuba-Borda et al., 2019) for the medium-scale model of (Boehl and Strobel, 2020).
Note: The table shows the execution times per state vector in seconds. "No ZLB solution" collects all draws for which the iterative method either did not converge, or where a higher number of $l + k$ than `nperiods` was called during the iterative procedure.

4 Conclusion

This paper presents a fast and robust solution method for macroeconomic models with occasionally binding constraints. I transform the linearized equilibrium conditions into an extended reduced-form system that depends only on the initial states and the expected number of periods at the constraint. This allows for a very efficient computation of the solution. Speed benchmarks confirm very fast computation times even for large-scale models with heterogeneous agents and almost 1000 variables. The method performs more than 1500 times faster than OccBin (Guerrieri and Iacoviello, 2015). The work in Boehl et al. (2020); Boehl and Strobel (2020) confirms that the proposed method, when combined with a nonlinear filter for likelihood inference, enables the estimation of large-scale dynamic models while fully accounting for an endogenously binding zero lower bound on nominal interest rates during the estimation procedure.

References

Adjemian, S., Bastani, H., Juillard, M., Karamé, F., Maih, J., Mihoubi, F., Perendia, G., Pfeifer, J., Ratto, M., Villemot, S., 2011. Dynare: Reference Manual Version 4. Dynare Working Papers 1. CEPREMAP.

Bayer, C., Born, B., Luetticke, R., 2020. Shocks, Frictions, and Inequality in US Business Cycles. CEPR Discussion Papers 14364.

Binder, M., Pesaran, M.H., 1995. Multivariate rational expectations models and macroeconometric modelling: A review and some new results. Handbook of applied econometrics 1, 139–187.

Boehl, G., Goy, G., Strobel, F., 2020. A structural investigation of quantitative easing. IMFS Working Paper Series 142. Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Boehl, G., Lieberknecht, P., 2021. The Hockey Stick Phillips Curve and the Zero Lower Bound. Technical Report. URL: `https://gregorboehl.com`.

Boehl, G., Strobel, F., 2020. US business cycle dynamics at the zero lower bound. IMFS Working Paper Series 143. Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Carlstrom, C.T., Fuerst, T.S., Paustian, M., 2015. Inflation and output in new keynesian models with a transient interest rate peg. Journal of Monetary Economics 76, 230–243.

Coenen, G., Orphanides, A., Wieland, V., 2004. Price stability and monetary policy effectiveness when nominal interest rates are bounded at zero. Advances in Macroeconomics 4.

Coenen, G., Wieland, V., 2003. The zero-interest-rate bound and the role of the exchange rate for monetary policy in japan. Journal of Monetary Economics 50, 1071–1101.

Coenen, G., Wieland, V.W., 2004. Exchange-rate policy and the zero bound on nominal interest rates. American Economic Review 94, 80–84.

Cuba-Borda, P., Guerrieri, L., Iacoviello, M., Zhong, M., 2019. Likelihood evaluation of models with occasionally binding constraints. Journal of Applied Econometrics 34, 1073–1085.

Guerrieri, L., Iacoviello, M., 2015. Occbin: A toolkit for solving dynamic models with occasionally binding constraints easily. Journal of Monetary Economics 70, 22–38.

Holden, T.D., 2016. Computation of solutions to dynamic models with occasionally binding constraints. Technical Report.

Holden, T.D., 2017. Existence and uniqueness of solutions to dynamic models with occasionally binding constraints. Technical Report.

Kaplan, G., Moll, B., Violante, G.L., 2018. Monetary policy according to HANK. Technical Report 3.

Klein, P., 2000. Using the generalized schur form to solve a multivariate linear rational expectations model. Journal of economic dynamics and control 24, 1405–1423.

Reiter, M., 2009. Solving heterogeneous-agent models by projection and perturbation. Journal of Economic Dynamics and Control 33, 649–665.

Rendahl, P., 2017. Linear time iteration. Technical Report. IHS Economics Series.

Schmitt-Grohé, S., Uribe, M., 2004. Solving dynamic general equilibrium models using a second-order approximation to the policy function. Journal of economic dynamics and control 28, 755–775.

Smets, F., Wouters, R., 2007. Shocks and frictions in us business cycles: A bayesian dsge approach. American Economic Review 97, 586–606.

Uhlig, H., et al., 1995. A toolkit for analyzing nonlinear dynamic stochastic models easily .

Villemot, S., et al., 2011. Solving rational expectations models at first order: what Dynare does. Technical Report. Citeseer.