FEDERAL RESERVE BANK *of* ATLANTA WORKING PAPER SERIES

Perturbation Methods for Markov-Switching DSGE Models

Andrew Foerster, Juan Rubio-Ramírez, Daniel F. Waggoner, and Tao Zha

Working Paper 2013-1
March 2013

Abstract: This paper develops a general perturbation methodology for constructing high-order approximations to the solutions of Markov-switching DSGE models. We introduce an important and practical idea of partitioning the Markov-switching parameter space so that a steady state is well defined. With this definition, we show that the problem of finding an approximation of any order can be reduced to solving a system of quadratic equations. We propose using the theory of Grobner bases in searching all the solutions to the quadratic system. This approach allows us to obtain all the approximations and ascertain how many of them are stable. Our methodology is applied to three models to illustrate its feasibility and practicality.

JEL classification: C6, E1

Key words: Markov-switching parameters, partition, higher order approximations, no certainty equivalence, quadratic system, Grobner bases

---

The authors thank Leonardo Melosi, Seonghoon Cho, Rhys Bidder, seminar participants at Duke University, the Federal Reserve Bank of St. Louis, the 2010 Society of Economic Dynamics meetings, the 2011 Federal Reserve System Committee on Business and Financial Analysis Conference, the 2012 Annual Meeting of the American Economic Association, the 8th Dynare Conference, and the 2012 National Bureau of Economic Research (NBER) Workshop on Methods and Applications for DSGE Models for helpful comments. This research is supported in part by the National Science Foundation Grant Nos. SES-1127665 and SES-1227397. The views expressed here are the authors' and not necessarily those of the Federal Reserve Banks of Atlanta or Kansas City, the Federal Reserve System, or the NBER. Any remaining errors are the authors' responsibility.

Please address questions regarding content to Andrew Foerster, Federal Reserve Bank of Kansas City, 1 Memorial Drive, Kansas City, MO 64198, 816-881-2751, andrew.foerster@kc.frb.org; Juan Rubio-Ramírez, Duke University, Federal Reserve Bank of Atlanta, CEPR, FEDEA, and BBVA Research, P.O. Box 90097, Durham, NC 27708, 919-660-1865, juan.rubio-ramirez@duke.edu; Daniel F. Waggoner, Federal Reserve Bank of Atlanta, 1000 Peachtree Street, N.E., Atlanta, GA 30309-4470, dwaggoner@frbatlanta.org; or Tao Zha, Federal Reserve Bank of Atlanta, Emory University, and NBER, 1000 Peachtree Street, N.E., Atlanta, GA 30309-4470, zmail@tzha.net.

Federal Reserve Bank of Atlanta working papers, including revised versions, are available on the Atlanta Fed's website at frbatlanta.org/pubs/WP/. Use the WebScriber Service at frbatlanta.org to receive e-mail notifications about new papers.

Perturbation Methods for Markov-Switching DSGE Models

1 Introduction

In this paper we show how to use perturbation methods as described in Judd (1998) and Schmitt-Grohe and Uribe (2004) to solve Markov-switching dynamic stochastic general equilibrium (MSDSGE) models. Our contribution advances the current literature in two significant respects. First, we develop a general methodology for approximating the solution of a larger class of Markov-switching models than currently possible and, second, we show the feasibility and practicality of implementing our methodology when we consider high-order approximations to the model solution. Current methods only allow for first-order approximations.

The literature on Markov-switching linear rational expectations (MSLRE) models has been an active field in empirical macroeconomics (Leeper and Zha (2003), Blake and Zampolli (2006), Svensson and Williams (2007), Davig and Leeper (2007), and Farmer et al. (2009)). Building on standard linear rational expectations models, the MSLRE approach allows model parameters to change over time according to discrete Markov chain processes. This nonlinearity has proven to be important in explaining changes in monetary policy and macroeconomic time series (Schorfheide (2005), Davig and Doh (2008), Liu et al. (2011), and Bianchi (2010)) and in modeling the expected effects of future fiscal policy changes (Davig et al. (2010), Davig et al. (2011), Bi and Traum (2012)). In particular, Markov-switching models provide a tractable way to study how agents form expectations over possible discrete changes in the economy, such as those in technology and policy.

There are, however, two main shortcomings with the MSLRE approach. First, that approach begins with a system of standard linear rational expectations equations that have been obtained from linearizing equilibrium conditions as though the parameters were constant over time. Discrete Markov chain processes are then annexed to certain parameters. As a consequence, the resultant MSLRE model may be incompatible with the optimizing behavior of agents in an

original economic model with Markov-switching parameters. Second, because it builds on linear rational expectations models, the MSLRE approach does not take into account higher-order coefficients in the approximation. Higher-order approximations improve the approximation accuracy and may be potentially important for certain economic questions.

This paper develops a general perturbation methodology for constructing first-order and second-order approximations to the solutions of MSDSGE models in which certain parameters vary according to discrete Markov chain processes. Our method can be easily expanded to higher-order approximations. The key is to find the approximations using the equilibrium conditions implied by the original economic model when Markov-switching parameters are present. Thus, our method overcomes the aforementioned shortcomings. By working with the original MSDSGE model directly rather than taking a system of linear rational expectations equations with constant parameters as a short-cut, we maintain congruity between the original economic model with Markov-switching parameters and the resultant approximations to the model solution. Such congruity is necessary for researchers to derive both first-order and second-order approximations.

Unlike the case with a standard model with constant parameters, one conceptual difficulty while working with MSDSGE models is that certain Markov-switching parameters, such as the mean growth rate of technology, complicate the steady-state definition. The definition of the steady-state for MSDSGE models should be independent of the realization of the discrete Markov chain process for any changing parameter. To this end, we introduce a new concept of steady-state. We partition the parameter space such that the subset of Markov-switching parameters that would influence the steady-state in the constant parameter case is now a function of the perturbation parameter, while the complementary subset is not. This new concept renders a key to deriving first-order and second-order approximations.

Several important results are as follows. First, we find that first-order approximations to the solutions of MSDSGE models are, in general, not certainty equivalent when a subset of Markov-switching parameters needs to be perturbed. Second, we identify the task of finding all the solutions to a system of quadratic equations as the only bottleneck in obtaining first-order

and second-order approximations to the solutions of MSDSGE models.

Third, we propose to remove the bottleneck by applying Gröbner bases, a key insight of our approach. Gröbner bases have not only a sound mathematical theory but also many computational applications. For example, Gröbner bases have been successfully applied by Kubler and Schmedders (2010a) and Kubler and Schmedders (2010b) to find multiple equilibria in general equilibrium models; Datta (2010) uses such bases to find all the Nash equilibria. For our purpose, Gröbner bases provide a computationally feasible way to obtain all the solutions to a system of polynomial equations. Once the bottleneck of solving a system of quadratic equations is resolved, the remaining task to obtain first-order and second-order approximations involves solving only systems of linear equations even for higher-order approximations.

Fourth, one may use a numerical algorithm to search for a solution to the quadratic system, but there is no guarantee that such an algorithm is capable of finding all solutions. By employing Gröbner bases to solve the quadratic system, we can first obtain all its solutions and then determine how many approximations are stable, where the stability concept follows the earlier work of Costa et al. (2005), Farmer et al. (2009), Farmer et al. (2011), and Cho (2011). This procedure enables researchers to ascertain both the existence and the uniqueness of a stable approximation.

The rest of the paper is organized as follows. Section 2 presents a general class of MSDSGE models, outlines our methodology, and introduces a new concept of steady-state. Section 3 derives first-order approximations and discusses the feature of no certainty equivalence. Section 4 shows how to reduce the problem of finding first-order approximations to that of finding all the solutions to a system of quadratic equations. Gröbner bases are proposed to tackle this problem and the concept of stability is introduced in this section. Section 5 derives a second-order approximation and shows that it involves solving only linear systems once a first-order approximation is obtained. Section 6 illustrates how to apply our methodology to three different MSDSGE models. Concluding remarks are offered in Section 7.

2 The General Framework

Our general framework is laid out as follows. In Section 2.1 we discuss a general class of MSDSGE models with a simple example to guide the reader through our new notation. In Section 2.2 we introduce a new concept of steady-state and discuss a new idea of partitioning the Markov-switching parameter space. In Section 2.3 we present the general form of model solutions and state the goal of this paper.

2.1 The Model

We study a general class of MSDSGE models in which some of the parameters follow a discrete Markov chain process indexed by s_t with the transition matrix $P = (p_{s,s'})$. The element $p_{s,s'}$ represents the probability that $s_{t+1} = s'$ given $s_t = s$ for $s, s' \in \{1, \ldots, n_s\}$, where n_s is the number of regimes. When $s_t = s$, the model is said to be in regime s at time t. We denote the vector of all Markov-switching parameters by θ_t, which has dimension $n_\theta \times 1$.[1]

Given $(x_{t-1}, \varepsilon_t, \theta_t)$, the equilibrium conditions for MSDSGE models have the general form

$$\mathbb{E}_t f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right) = 0_{n_x+n_y}, \tag{1}$$

where $\mathbb{E}_t$ denotes the mathematical expectation operator conditional on information available at time t, $0_{n_x+n_y}$ is an $(n_x + n_y) \times 1$ vector of zeros, x_t is an $n_x \times 1$ vector of (endogenous and exogenous) predetermined variables, y_t is an $n_y \times 1$ vector of non-predetermined (control) variables, ε_t is an $n_\varepsilon \times 1$ vector of i.i.d. innovations to the exogenous predetermined variables with $\mathbb{E}_{t-1}\varepsilon_t = 0_{n_\varepsilon}$, and $\chi \in \mathbb{R}$ is the perturbation parameter. Since the total number of equations in (1) is $n_y + n_x$, the function f maps $\mathbb{R}^{2(n_y+n_x+n_\varepsilon+n_\theta)}$ into $\mathbb{R}^{n_y+n_x}$.

Finding first-order and second-order approximations to the solutions of MSDSGE models using perturbation methods requires us to introduce new notation and lengthy algebraic work. To aid the reader through our new notation and derivations in the paper, we use a simple real business cycle (RBC) model as a clear demonstration of how our proposed methodology works.

[1] Note that the vector θ_t does not include other parameters that are constant over time. We call those parameters "constant parameters."

The RBC model is an economy with the representative household whose preferences over a stochastic sequence of consumption goods, c_t, are represented by the utility function

$$\max \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \log c_t$$

where β denotes the discount factor. The resource constraint is

$$c_t + k_t = z_t k_{t-1}^{\alpha} + (1-\delta) k_{t-1},$$

where k_t is a stock of physical capital and z_t represents a technological change that is a random walk in log with a Markov-switching drift as

$$\log z_t = \mu_t + \log z_{t-1} + \sigma \varepsilon_t,$$

where the drift μ_t takes two discrete values dictated by the Markov chain process represented by $s_t \in \{1, 2\}$, and $\varepsilon_t \sim N(0, 1)$.

The three equations characterizing the equilibrium are

$$\frac{1}{c_t} = \beta \mathbb{E}_t \frac{1}{c_{t+1}} \left(\alpha z_{t+1} k_t^{\alpha-1} + 1 - \delta \right),$$

$$c_t + k_t = z_t k_{t-1}^{\alpha} + (1-\delta) k_{t-1},$$

$$\text{and } \log z_t = \mu_t + \log z_{t-1} + \sigma \varepsilon_t.$$

Because $\log z_t$ has a unit root, the economy is non-stationary. To derive a stationary equilibrium, define $\omega_t = z_{t-1}^{\frac{1}{1-\alpha}}$ and let $\tilde{c}_t = \frac{c_t}{\omega_t}$, $\tilde{k}_{t-1} = \frac{k_{t-1}}{\omega_t}$, $\tilde{z}_t = \frac{z_t}{z_{t-1}}$. The transformed (re-scaled) equilibrium conditions become

$$\frac{1}{\tilde{c}_t} = \beta \mathbb{E}_t \frac{\tilde{z}_t^{\frac{1}{\alpha-1}}}{\tilde{c}_{t+1}} \left(\alpha \tilde{z}_{t+1} \tilde{k}_t^{\alpha-1} + 1 - \delta \right),$$

$$\tilde{c}_t + \tilde{k}_t \tilde{z}_t^{\frac{1}{1-a}} = \tilde{z}_t \tilde{k}_{t-1}^{\alpha} + (1-\delta) \tilde{k}_{t-1},$$

$$\text{and } \log \tilde{z}_t = \mu_t + \sigma \varepsilon_t.$$

Substituting out $\tilde{z}_t$ leads to the following two equilibrium conditions

$$\frac{1}{\tilde{c}_t} = \beta \mathbb{E}_t \frac{1}{\tilde{c}_{t+1}} \exp\left(\frac{\mu_t + \sigma \varepsilon_t}{\alpha - 1} \right) \left(\alpha \exp\left(\mu_{t+1} + \sigma \varepsilon_{t+1} \right) \tilde{k}_t^{\alpha-1} + 1 - \delta \right),$$

$$\text{and } \tilde{c}_t + \tilde{k}_t \exp\left(\frac{\mu_t + \sigma\varepsilon_t}{1-\alpha}\right) = \exp\left(\mu_t + \sigma\varepsilon_t\right)\tilde{k}_{t-1}^{\alpha} + (1-\delta)\,\tilde{k}_{t-1}.$$

Using the notation in this section, we have $n_y = 1$, $n_x = 1$, $n_\varepsilon = 1$, $n_\theta = 1$, $y_t = \tilde{c}_t$, $x_{t-1} = \tilde{k}_{t-1}$, and $\theta_t = \mu_t$. The equilibrium condition (1) can be specifically expressed as

$$\mathbb{E}_t f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right) =$$
$$\mathbb{E}_t \begin{bmatrix} \frac{1}{\tilde{c}_t} - \beta\frac{1}{\tilde{c}_{t+1}}\exp\left(\frac{\mu_t+\sigma\varepsilon_t}{\alpha-1}\right)\left(\alpha\exp\left(\mu_{t+1}+\chi\sigma\varepsilon_{t+1}\right)\tilde{k}_t^{\alpha-1} + 1 - \delta\right) \\ \tilde{c}_t + \tilde{k}_t\exp\left(\frac{\mu_t+\sigma\varepsilon_t}{1-\alpha}\right) - \exp\left(\mu_t+\sigma_t\varepsilon_t\right)\tilde{k}_{t-1}^{\alpha} - (1-\delta)\,\tilde{k}_{t-1} \end{bmatrix}. \tag{2}$$

2.2 Steady-State and Partition of Markov-Switching Parameters

The definition of the steady-state for MSDSGE models is more complicated than that for standard DSGE models with constant parameters. To be consistent with the definition of the steady-state in standard DSGE models with constant parameters, the definition of the steady-state for MSDSGE models should be independent of the realization of the discrete Markov chain process.

To achieve this objective, our key idea is to partition the vector θ_t of Markov-switching parameters into two subvectors. The first part concerns a subvector of parameters that would influence the steady-state in the constant parameter case and thus is a function of the perturbation parameter χ. The second part contains the subvector of all remaining parameters that would not affect the steady-state in the constant parameter case. We denote the first subvector by θ_{1t} and the second subvector by θ_{2t}. Rewrite the vector θ_t as

$$\theta_t = \theta\left(\chi, s_t\right), \tag{3}$$

where θ maps $\mathbb{R} \times \{1, \ldots, n_s\}$ into $\mathbb{R}^{n_\theta}$. We have

$$\theta_t = \begin{bmatrix} \theta_{1t}^{\intercal} & \theta_{2t}^{\intercal} \end{bmatrix}^{\intercal} = \begin{bmatrix} \theta_1\left(\chi, s_t\right)^{\intercal} & \theta_2\left(\chi, s_t\right)^{\intercal} \end{bmatrix}^{\intercal}, \tag{4}$$

where $\intercal$ indicates transpose. The vector θ_{1t} and θ_{2t} have the dimensions $n_{\theta_1} \times 1$ and $n_{\theta_2} \times 1$ respectively, and functional forms

$$\theta_1\left(\chi, s_t\right) = \overline{\theta}_1 + \chi\widehat{\theta}_1\left(s_t\right), \tag{5}$$

$$\text{and } \theta_2\left(\chi, s_t\right) = \widehat{\theta}_2\left(s_t\right) \tag{6}$$

for all s_t.

The specific functional form (5) is chosen for tractable derivations in the rest of the paper.[2] We will discuss, below, how to choose the value of $\overline{\theta}_1$. We apply the same partition to the vector θ_{t+1}, where we simply replace the subscript t with $t+1$.

Two important features stand out from (5) and (6). First, $\widehat{\theta}_1(s_t)$ is a deviation of θ_{1t} from $\overline{\theta}_1$ in regime s_t. Second, θ_{2t} is not a function of the perturbation parameter χ. Thus, the perturbation parameter, χ, affects only a subset of Markov-switching parameters, θ_{1t}. Since the steady-state depends on $\bar{\theta}_1$ only, a natural choice for this point is the mean of the ergodic distribution across θ_{1t}.

Now we are ready to define the steady-state of the MSDSGE model.

Definition 1 *The $n_x \times 1$ vector x_{ss} and the $n_y \times 1$ vector y_{ss} are the steady-state variables if*

$$f\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2(s_{t+1}), \overline{\theta}_1, \widehat{\theta}_2(s_t)\right) = 0_{n_x+n_y}$$

holds for all s_{t+1} and s_t.

In principle, more than one partition of Markov-switching parameters can satisfy the condition, as stated in Definition 1, such that neither $\theta_2(0, s_{t+1}) = \widehat{\theta}_2(s_{t+1})$ nor $\theta_2(0, s_t) = \widehat{\theta}_2(s_t)$ enters in the calculation of the steady-state for any s_{t+1} or s_t. We recommend choosing θ_{1t} as the smallest set of Markov-switching parameters such that $\bar{\theta}_1$ influences the steady-state.

In our simple RBC model, there is no θ_{2t}, but $\theta_{1t} = \mu_t$ takes the functional form

$$\mu_t = \mu(\chi, s_t) = \overline{\mu} + \chi\widehat{\mu}(s_t).$$

We set both $\varepsilon_t = 0$ and $\chi = 0$. Thus, $\mu_t = \mu_{t+1} = \overline{\mu}$ and the steady-state of the RBC model consists of $\tilde{c}_{ss}$ and $\tilde{k}_{ss}$ such that

$$\begin{bmatrix} \frac{1}{c_{ss}} - \beta\frac{1}{c_{ss}}\exp\left(\frac{\bar{\mu}}{\alpha-1}\right)\left(\alpha\exp(\bar{\mu})\tilde{k}_{ss}^{\alpha-1} + 1 - \delta\right) \\ \tilde{c}_{ss} + \tilde{k}_{ss}\exp\left(\frac{\bar{\mu}}{1-\alpha}\right) - \exp(\bar{\mu})\tilde{k}_{ss}^{\alpha} - (1-\delta)\tilde{k}_{ss} \end{bmatrix} = 0_2,$$

[2] Any other functional form, so long as $\theta_1(0, s_t) = \overline{\theta}_1$ holds for *all* s_t, will be valid.

which produces the steady-state values

$$\begin{aligned}\tilde{k}_{ss} &= \left(\frac{1}{\alpha \exp(\bar{\mu})}\left(\frac{1}{\beta \exp\left(\frac{\bar{\mu}}{\alpha-1}\right)} - 1 + \delta\right)\right)^{\frac{1}{\alpha-1}}, \\ \text{and } \tilde{c}_{ss} &= \exp(\bar{\mu})\, \tilde{k}_{ss}^{\alpha} + (1-\delta)\, \tilde{k}_{ss} - \tilde{k}_{ss} \exp\left(\frac{\bar{\mu}}{1-\alpha}\right).\end{aligned}$$

2.3 Model Solution and Approximation

Once the concept of steady-state and the partition of Markov-switching parameters are clearly defined, we can now proceed to approximate the solution of a MSDSGE model. Consider model solutions of the form[3]

$$y_t = g\left(x_{t-1}, \varepsilon_t, \chi, s_t\right), \tag{7}$$

$$y_{t+1} = g\left(x_t, \chi\varepsilon_{t+1}, \chi, s_{t+1}\right), \tag{8}$$

$$\text{and } x_t = h\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) \tag{9}$$

where g maps $\mathbb{R}^{n_x+n_\varepsilon+1} \times \{1, \ldots, n_s\}$ into $\mathbb{R}^{n_y}$ and h maps $\mathbb{R}^{n_x+n_\varepsilon+1} \times \{1, \ldots, n_s\}$ into $\mathbb{R}^{n_x}$. In the RBC model, we have

$$\begin{aligned}\tilde{c}_t &= g\left(\tilde{k}_{t-1}, \varepsilon_t, \chi, s_t\right), \\ \tilde{c}_{t+1} &= g\left(\tilde{k}_t, \chi\varepsilon_{t+1}, \chi, s_{t+1}\right), \\ \text{and } \tilde{k}_t &= h\left(\tilde{k}_{t-1}, \varepsilon_t, \chi, s_t\right).\end{aligned}$$

In general, we do not know the explicit functional forms of g and h; thus, we need to approximate them by Taylor expansions around the steady-state.[4] In this paper, we present the results up to second-order Taylor expansions, but our procedure can be easily expanded to higher orders. For the rest of the paper, first-order Taylor expansions are referred to as first-order approximations and second-order Taylor expansions as second-order approximations.

[3] In theory, one could allow g and h to depend on the entire history of the regimes. In practice, this is not tractable. The assumption that g and h depend only on the current regime corresponds to the notion of a minimal state variable (MSV) solution in Farmer et al. (2011).

[4] We assume that the solutions g and h are unique.

Given the steady-state defined in Definition 1, we have

$$y_{ss} = g\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) \text{ and } x_{ss} = h\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)$$

for all s_t and

$$y_{ss} = g\left(x_{ss}, 0_{n_\varepsilon}, 0, s_{t+1}\right) \text{ and } x_{ss} = h\left(x_{ss}, 0_{n_\varepsilon}, 0, s_{t+1}\right)$$

for all s_{t+1}.

Using Equations (1), (7), (8), and (9), we write the function f as

$$\mathcal{F}\left(x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \chi, s_{t+1}, s_t\right) = f\left(\begin{array}{c} g\left(h\left(x_{t-1}, \varepsilon_t, \chi, s_t\right), \chi\varepsilon_{t+1}, \chi, s_{t+1}\right), g\left(x_{t-1}, \varepsilon_t, \chi, s_t\right), \\ h\left(x_{t-1}, \varepsilon_t, \chi, s_t\right), x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta\left(\chi, s_{t+1}\right), \theta\left(\chi, s_t\right) \end{array}\right)$$

for all $x_{t-1}, \varepsilon_{t+1}, \varepsilon_t, s_{t+1}$, and s_t. The function $\mathcal{F}$ maps $\mathbb{R}^{n_x+2n_\varepsilon+1} \times \{1, \ldots, n_s\} \times \{1, \ldots, n_s\}$ into $\mathbb{R}^{n_y+n_x}$. With the assumption that innovations to the exogenous predetermined variables, ε_t, are independent of the discrete Markov chain process, s_t, we write (1) as

$$\mathbb{E}_t f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right) = \tag{10}$$
$$\mathbb{G}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) = \sum_{s'=1}^{n_s} p_{s_t,s'} \int \mathcal{F}\left(x_{t-1}, \chi\varepsilon', \varepsilon_t, \chi, s', s_t\right) \mu\left(\varepsilon'\right) d\varepsilon' = 0_{n_y+n_x}$$

for all x_{t-1}, ε_t, and s_t, where μ is the density function of the innovations. The function $\mathbb{G}$ maps $\mathbb{R}^{n_x+n_\varepsilon+1} \times \{1, \ldots, n_s\}$ into $\mathbb{R}^{n_y+n_x}$.

This paper has two goals. First, we show how to derive first-order and second-order approximations to the functions g and h around the steady-state. Second, we show that obtaining these approximations requires solving a quadratic system. We propose a new methodology to find all the solutions to such a system. Each of the solutions to the system corresponds to a different set of first-order and second-order approximations. Finding all the solutions is a difficult task but is crucial to determine how many approximations are stable.[5] Sections 3-5, below, are devoted to accomplishing these two objectives.

[5]The stability definition is discussed in Section 4.2.3.

3 First-Order Approximations

This section gives a detailed description of how to derive first-order approximations to the model solution represented by (7), (8), and (9). We proceed in several steps. We first lay out the notation in Section 3.1 and then provide detailed derivations needed to find the approximations in Section 3.2. With the notation and derivations in hand, we show in Section 3.3 how to obtain first-order approximations to the model solution. In a final subsection, Section 3.4, we discuss one of the most important features of MSDSGE models: the result of no certainty equivalence of first-order approximations.

3.1 Notation

3.1.1 Partial Derivatives of f

We begin with the notation for the derivatives of

$$f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right)$$

evaluated at the steady-state. Let

$$\mathcal{D}f_{ss}\left(s_{t+1}, s_t\right) =$$
$$\left[\mathcal{D}_j f^i\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2\left(s_{t+1}\right), \overline{\theta}_1, \widehat{\theta}_2\left(s_t\right)\right)\right]_{1\leq i\leq n_y+n_x, 1\leq j\leq 2(n_y+n_x+n_\varepsilon+n_\theta)}$$

denote the $(n_y + n_x)\times(2\left(n_y + n_x + n_\varepsilon + n_\theta\right))$ matrix of first partial derivatives of f with respect to all its variables evaluated at

$$\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2\left(s_{t+1}\right), \overline{\theta}_1, \widehat{\theta}_2\left(s_t\right)\right)$$

for all s_{t+1} and s_t. To simplify notation, define

$$\mathcal{D}_{n,m}f_{ss}\left(s_{t+1}, s_t\right) = \left[\mathcal{D}_n f_{ss}\left(s_{t+1}, s_t\right)\cdots\mathcal{D}_m f_{ss}\left(s_{t+1}, s_t\right)\right]$$

for all s_{t+1} and s_t and $1 \leq n < m \leq 2\left(n_y + n_x + n_\varepsilon + n_\theta\right)$.[6]

[6] When $n = m$, $\mathcal{D}_{n,n}f_{ss}\left(s_{t+1}, s_t\right) = \mathcal{D}_n f_{ss}\left(s_{t+1}, s_t\right)$ for all s_{t+1} and s_t.

Appendix A shows that our derivations depend on the following matrices:

$$\mathcal{D}_{1,n_y} f_{ss}\left(s_{t+1}, s_t\right), \mathcal{D}_{n_y+1,2n_y} f_{ss}\left(s_{t+1}, s_t\right),$$

$$\mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}\left(s_{t+1}, s_t\right), \mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)} f_{ss}\left(s_{t+1}, s_t\right),$$

$$\mathcal{D}_{2(n_y+n_x)+1,2(n_y+n_x)+n_\varepsilon} f_{ss}\left(s_{t+1}, s_t\right), \mathcal{D}_{2(n_y+n_x)+n_\varepsilon+1,2(n_y+n_x+n_\varepsilon)} f_{ss}\left(s_{t+1}, s_t\right),$$

$$\mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}\left(s_{t+1}, s_t\right), \text{ and } \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}\left(s_{t+1}, s_t\right)$$

for all s_{t+1} and s_t.

Given the importance of these derivatives, we return to the RBC example for concrete illustration. It follows from (2) that

$$\mathcal{D}_{1,1} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} \frac{1}{c_{ss}^2} \\ 0 \end{bmatrix}, \mathcal{D}_{2,2} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} -\frac{1}{c_{ss}^2} \\ 1 \end{bmatrix},$$

$$\mathcal{D}_{3,3} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} (1-\alpha)\,\alpha\beta \exp\left(\frac{\alpha\bar{\mu}}{\alpha-1}\right) \frac{k_{ss}^{\alpha-2}}{c_{ss}} \\ \exp\left(\frac{\bar{\mu}}{1-\alpha}\right) \end{bmatrix}, \mathcal{D}_{4,4} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 \\ -\frac{\exp\left(\frac{\bar{\mu}}{1-\alpha}\right)}{\beta} \end{bmatrix},$$

$$\mathcal{D}_{5,5} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} -\alpha\beta \exp\left(\frac{\alpha\bar{\mu}}{\alpha-1}\right) \frac{k_{ss}^{\alpha-1}}{c_{ss}} \sigma \\ 0 \end{bmatrix}, \mathcal{D}_{6,6} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} \frac{\sigma}{(1-\alpha)c_{ss}} \\ \left(\frac{\exp\left(\frac{\bar{\mu}}{1-\alpha}\right)k_{ss}}{1-\alpha} - \exp\left(\bar{\mu}\right) k_{ss}^{\alpha}\right)\sigma \end{bmatrix}$$

$$\mathcal{D}_{7,7} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} -\alpha\beta \exp\left(\frac{\bar{\mu}\alpha}{\alpha-1}\right) \frac{k_{ss}^{\alpha-1}}{c_{ss}} \\ 0 \end{bmatrix}, \text{ and } \mathcal{D}_{8,8} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} \frac{1}{c_{ss}} \\ \exp\left(\frac{\bar{\mu}}{1-\alpha}\right) \frac{k_{ss}}{1-\alpha} - \exp\left(\bar{\mu}\right) k_{ss}^{\alpha} \end{bmatrix}$$

for all s_{t+1} and s_t.

3.1.2 Partial Derivatives of $\mathbb{G}$

We now introduce the notation for the derivatives of $\mathbb{G}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right)$. Let

$$\mathcal{D}\mathbb{G}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) = \left[\mathcal{D}_j \mathbb{G}^i\left(x_{t-1}, \varepsilon_t, \chi, s_t\right)\right]_{1 \le i \le n_y+n_x, 1 \le j \le n_x+n_\varepsilon+1}$$

refer to the $(n_y + n_x) \times (n_x + n_\varepsilon + 1)$ matrix of first partial derivatives of $\mathbb{G}$ with respect to $(x_{t-1}, \varepsilon_t, \chi)$ for all $x_{t-1}, \varepsilon_t, \chi$, and s_t. Note that there are no derivatives with respect to s_t since it is a discrete variable. Similarly, we let

$$\mathcal{D}\mathbb{G}\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) = \left[\mathcal{D}_j \mathbb{G}^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)\right]_{1 \le i \le n_y+n_x, 1 \le j \le n_x+n_\varepsilon+1}$$

refer to the $(n_y + n_x) \times (n_x + n_\varepsilon + 1)$ matrix of first partial derivatives of $\mathbb{G}$ with respect to $(x_{t-1}, \varepsilon_t, \chi)$ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t.

To simplify notation, we define

$$\mathcal{D}\mathbb{G}_{ss}(s_t) = \mathcal{D}\mathbb{G}(x_{ss}, 0_{n_\varepsilon}, 0, s_t) \text{ and } \mathcal{D}_j\mathbb{G}^i_{ss}(s_t) = \mathcal{D}_j\mathbb{G}^i(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$$

for all s_t and $1 \leq i \leq n_y + n_x$ and $1 \leq j \leq n_x + n_\varepsilon + 1$. Thus,

$$\mathcal{D}\mathbb{G}_{ss}(s_t) = \left[\mathcal{D}_j\mathbb{G}^i_{ss}(s_t)\right]_{1 \leq i \leq n_y + n_x, 1 \leq j \leq n_x + n_\varepsilon + 1}$$

for all s_t. Let $\mathcal{D}_j\mathbb{G}_{ss}(s_t)$ denote the jth column vector of $\mathcal{D}\mathbb{G}_{ss}(s_t)$ for all s_t and $1 \leq j \leq n_x + n_\varepsilon + 1$. It follows that the first partial derivatives of $\mathbb{G}$ with respect to x_{t-1} evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t can be expressed as

$$\left[\mathcal{D}_1\mathbb{G}_{ss}(s_t) \cdots \mathcal{D}_{n_x}\mathbb{G}_{ss}(s_t)\right],$$

the first partial derivatives of $\mathbb{G}$ with respect to ε_t evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t can be expressed as

$$\left[\mathcal{D}_{n_x+1}\mathbb{G}_{ss}(s_t) \cdots \mathcal{D}_{n_x+n_\varepsilon}\mathbb{G}_{ss}(s_t)\right],$$

and the first partial derivative of $\mathbb{G}$ with respect to χ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t is

$$\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t).$$

In contrast to one single first derivative in the constant parameter case, there are n_s first derivatives of $\mathbb{G}$, one for each possible value of s_t.

To simply notation further, we define

$$\mathcal{D}_{n,m}\mathbb{G}_{ss}(s_t) = \left[\mathcal{D}_n\mathbb{G}_{ss}(s_t) \cdots \mathcal{D}_m\mathbb{G}_{ss}(s_t)\right]$$

for all s_t and $1 \leq n < m \leq n_x + n_\varepsilon + 1$.[7] Therefore, $\mathcal{D}_{1,n_x}\mathbb{G}_{ss}(s_t)$ represents the first partial derivatives of $\mathbb{G}$ with respect to x_{t-1} evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, $\mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}(s_t)$ represents the first partial derivatives of $\mathbb{G}$ with respect to ε_t evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, and $\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t)$ is the first partial derivative of $\mathbb{G}$ with respect to χ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t.

[7] When $n = m$, then $\mathcal{D}_{n,n}\mathbb{G}_{ss}(s_t) = \mathcal{D}_n\mathbb{G}_{ss}(s_t)$ for all s_t.

3.1.3 Partial Derivatives of g and h

We are now ready to introduce the notation for the derivatives of g and h evaluated at the steady-state. Denote

$$\mathcal{D}g\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) = \left[\mathcal{D}_j g^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)\right]_{1\leq i\leq n_y, 1\leq j\leq n_x+n_\varepsilon+1}$$

as the $n_y \times (n_x + n_\varepsilon + 1)$ matrix of first partial derivatives of g with respect to $(x_{t-1}, \varepsilon_t, \chi)$ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t and

$$\mathcal{D}h\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) = \left[\mathcal{D}_j h^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)\right]_{1\leq i\leq n_x, 1\leq j\leq n_x+n_\varepsilon+1}$$

as the $n_x \times (n_x + n_\varepsilon + 1)$ matrix of first partial derivatives of h with respect to $(x_{t-1}, \varepsilon_t, \chi)$ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t.

Simplifying the notation further, we define

$$\mathcal{D}g_{ss}\left(s_t\right) = \mathcal{D}g\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) \text{ and } \mathcal{D}_j g^i_{ss}\left(s_t\right) = \mathcal{D}_j g^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)$$

for all s_t, $1 \leq i \leq n_y$, $1 \leq j \leq n_x + n_\varepsilon + 1$, and

$$\mathcal{D}h_{ss}\left(s_t\right) = \mathcal{D}h\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) \text{ and } \mathcal{D}_j h^i_{ss}\left(s_t\right) = \mathcal{D}_j h^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)$$

for all s_t, $1 \leq i \leq n_x$, and $1 \leq j \leq n_x + n_\varepsilon + 1$. Thus,

$$\begin{aligned} \mathcal{D}g_{ss}\left(s_t\right) &= \left[\mathcal{D}_j g^i_{ss}\left(s_t\right)\right]_{1\leq i\leq n_y, 1\leq j\leq n_x+n_\varepsilon+1} \\ \text{and } \mathcal{D}h_{ss}\left(s_t\right) &= \left[\mathcal{D}_j h^i_{ss}\left(s_t\right)\right]_{1\leq i\leq n_x, 1\leq j\leq n_x+n_\varepsilon+1} \end{aligned}$$

for all s_t. Let $\mathcal{D}_j g_{ss}\left(s_t\right)$ be the jth column vector of $\mathcal{D}g_{ss}\left(s_t\right)$ and $\mathcal{D}_j h_{ss}\left(s_t\right)$ be the jth column vector of $\mathcal{D}h_{ss}\left(s_t\right)$ for all s_t and $1 \leq j \leq n_x + n_\varepsilon + 1$. We then define

$$\mathcal{D}_{n,m} g_{ss}\left(s_t\right) = \left[\mathcal{D}_n g_{ss}\left(s_t\right) \cdots \mathcal{D}_m g_{ss}\left(s_t\right)\right] \text{ and } \mathcal{D}_{n,m} h_{ss}\left(s_t\right) = \left[\mathcal{D}_n h_{ss}\left(s_t\right) \cdots \mathcal{D}_m h_{ss}\left(s_t\right)\right]$$

for all s_t and $1 \leq n < m \leq n_x + n_\varepsilon + 1$.[8]

[8] When $n = m$, then $\mathcal{D}_{n,n} g_{ss}\left(s_t\right) = \mathcal{D}_n g_{ss}\left(s_t\right)$ and $\mathcal{D}_{n,n} h_{ss}\left(s_t\right) = \mathcal{D}_n h_{ss}\left(s_t\right)$ for all s_t.

As will be discussed in Section 3.3, the following matrices of first partial derivatives are our ultimate computational objects to obtain first-order approximations to the model solution

$$\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}, \quad \{\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t), \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t)\}_{s_t=1}^{n_s},$$
$$\text{and } \{\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t), \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t)\}_{s_t=1}^{n_s},$$

where $\mathcal{D}_{1,n_x} g_{ss}(s_t)$ represents the first partial derivatives of g with respect to x_{t-1} evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, $\mathcal{D}_{1,n_x} h_{ss}(s_t)$ represents the first partial derivatives of h with respect to x_{t-1} evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, $\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t)$ represents the first partial derivatives of g with respect to ε_t evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, $\mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t)$ represents the first partial derivatives of h with respect to ε_t evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, $\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t)$ is the first partial derivative of g with respect to χ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t, and $\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t)$ is the first partial derivative of h with respect to χ evaluated at $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$ for all s_t.

Note that

$$\begin{aligned}
\mathcal{D}_{1,n_x} g_{ss}(s_t) &\in \mathbb{C}^{n_y \times n_x}, \mathcal{D}_{1,n_x} h_{ss}(s_t) \in \mathbb{C}^{n_x \times n_x}, \\
\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t) &\in \mathbb{C}^{n_y \times n_\varepsilon}, \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t) \in \mathbb{C}^{n_x \times n_\varepsilon}, \\
\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) &\in \mathbb{C}^{n_y \times 1}, \text{ and } \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) \in \mathbb{C}^{n_x \times 1}
\end{aligned}$$

for all s_t. The symbol $\mathbb{C}^{m_1 \times m_2}$ denotes an $m_1 \times m_2$ matrix over the complex space. These computational objects can be found by solving systems of equations obtained through the chain rule using the first partial derivatives of $\mathbb{G}(x_{t-1}, \varepsilon_t, \chi, s_t)$. The next subsection shows the derivations needed to obtain such systems of equations.

3.2 Derivations

We now show how to use the chain rule using the first partial derivatives of $\mathbb{G}(x_{t-1}, \varepsilon_t, \chi, s_t)$ to obtain systems of equations that are needed to solve for the matrices

$$\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}, \quad \{\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t), \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t)\}_{s_t=1}^{n_s},$$
$$\text{and } \{\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t), \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t)\}_{s_t=1}^{n_s}.$$

Denote an $m_1 \times m_2$ matrix of zeros by $0_{m_1 \times m_2}$. Since

$$\mathbb{G}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) = 0_{n_y+n_x}$$

for all $x_{t-1}, \varepsilon_t, \chi$, and s_t, it follows that

$$\mathcal{D}\mathbb{G}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) = 0_{(n_y+n_x)\times(n_x+n_\varepsilon+1)}$$

for all $x_{t-1}, \varepsilon_t, \chi$, and s_t. In particular,

$$\mathcal{D}\mathbb{G}_{ss}\left(s_t\right) = 0_{(n_y+n_x)\times(n_x+n_\varepsilon+1)}$$

for all s_t so that

$$\begin{gathered} \mathcal{D}_{1,n_x}\mathbb{G}_{ss}\left(s_t\right) = 0_{(n_y+n_x)\times n_x}, \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}\left(s_t\right) = 0_{(n_y+n_x)\times n_\varepsilon}, \\ \text{and } \mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}\left(s_t\right) = 0_{n_y+n_x} \end{gathered} \tag{11}$$

for all s_t. Note that in contrast to one single derivative in the constant parameter case, there are n_s first partial derivatives of $\mathbb{G}$, one for each possible value of s_t.

The three expressions in (11) imply three systems of equations that will be used to find the needed matrices

$$\begin{gathered} \left\{\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right), \mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}, \ \left\{\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right), \mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}, \\ \text{and } \left\{\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}\left(s_t\right), \mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s} \end{gathered}$$

respectively. In what follows, we describe how to use these three systems of equations in (11) to obtain the needed matrices. We use the following steps to highlight the dependence between these systems.

Step 1 The condition

$$\mathcal{D}_{1,n_x}\mathbb{G}_{ss}\left(s_t\right) = 0_{(n_y+n_x)\times n_x}$$

implies a system of quadratic equations that determines $\left\{\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right), \mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$.

Step 2 The conditions

$$\mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}\left(s_t\right)=0_{(n_y+n_x)\times n_\varepsilon}\quad\text{and}\quad\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}\left(s_t\right)=0_{n_y+n_x}$$

imply two linear systems of equations that determine

$$\left\{\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right),\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$$

and

$$\left\{\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}\left(s_t\right),\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$$

as functions of $\left\{\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right),\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$.

Appendix A provides a detailed description of the derivations in Steps 1 and 2. In particular, it derives the first partial derivatives of $\mathbb{G}$ needed to build the three systems. It shows that $\left\{\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right),\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$ can be determined by solving n_s systems of quadratic equations of the form

$$A\left(s_t\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x}g_{ss}\left(1\right) \\ \vdots \\ \mathcal{D}_{1,n_x}g_{ss}\left(n_s\right)\end{bmatrix}\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)=B\left(s_t\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x}g_{ss}\left(s_t\right)\end{bmatrix} \tag{12}$$

for all s_t, where I_m denotes the identity matrix of size $m\times m$, and $A\left(s_t\right)$ and $B\left(s_t\right)$ are functions of the first partial derivatives of f evaluated at the steady-state as

$$A\left(s_t\right)=\begin{bmatrix}\sum_{s'=1}^{n_s}p_{s_t,s'}\mathcal{D}_{2n_y+1,2n_y+n_x}f_{ss}\left(s',s_t\right) & p_{s_t,1}\mathcal{D}_{1,n_y}f_{ss}\left(1,s_t\right) & \cdots & p_{s_t,n_s}\mathcal{D}_{1,n_y}f_{ss}\left(n_s,s_t\right)\end{bmatrix}$$

and

$$B\left(s_t\right)=-\sum_{s'=1}^{n_s}p_{s_t,s'}\begin{bmatrix}\mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)}f_{ss}\left(s',s_t\right) & \mathcal{D}_{n_y+1,2n_y}f_{ss}\left(s',s_t\right)\end{bmatrix}$$

for all s_t.

The fact that we need to solve a quadratic system to determine $\left\{\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right),\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$ is one of the key discoveries in the paper. The quadratic system has, in general, many solutions. Each solution corresponds to a different first-order approximation. Finding all the solutions to

this quadratic system is a difficult task but is crucial to ascertain how many of them imply stable approximations. In Section 4 we propose a new method to solve this quadratic system for all its solutions.

Appendix A shows how, after finding a solution to (12), Step 2 implies that $\{\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t), \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t)\}_{s_t=1}^{n_s}$ and $\{\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t), \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t)\}_{s_t=1}^{n_s}$ can be obtained by simply solving the following two systems of linear equations

$$\begin{bmatrix} \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(n_s) \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(n_s) \end{bmatrix} = \begin{bmatrix} \Theta_\varepsilon & \Phi_\varepsilon \end{bmatrix}^{-1} \Psi_\varepsilon \text{ and } \begin{bmatrix} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(n_s) \\ \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(n_s) \end{bmatrix} = \begin{bmatrix} \Theta_\chi & \Phi_\chi \end{bmatrix}^{-1} \Psi_\chi, \tag{13}$$

where

$$\Theta_\varepsilon = \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \mathcal{D}_{n_y+1,2n_y} f_{ss}(s',1) & \cdots & 0_{(n_x+n_y)\times n_y} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_y} & \cdots & p_{n_s,s'} \mathcal{D}_{n_y+1,2n_y} f_{ss}(s',n_s) \end{bmatrix},$$

$$\Phi_\varepsilon = \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \mathcal{D}_{1,n_y} f_{ss}(s',1) \mathcal{D}_{1,n_x} g_{ss}(s') & \cdots & 0_{(n_x+n_y)\times n_x} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_x} & \cdots & p_{n_s,s'} \mathcal{D}_{1,n_y} f_{ss}(s',n_s) \mathcal{D}_{1,n_x} g_{ss}(s') \end{bmatrix}$$
$$+ \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',1) & \cdots & 0_{(n_x+n_y)\times n_x} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_x} & \cdots & p_{n_s,s'} \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',n_s) \end{bmatrix},$$

$$\Psi_\varepsilon = -\sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \mathcal{D}_{2(n_y+n_x)+n_\varepsilon+1,2(n_y+n_x+n_\varepsilon)} f_{ss}(s',1) \\ \vdots \\ p_{n_s,s'} \mathcal{D}_{2(n_y+n_x)+n_\varepsilon+1,2(n_y+n_x+n_\varepsilon)} f_{ss}(s',n_s) \end{bmatrix},$$

$$\Theta_\chi = \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'}\mathcal{D}_{n_y+1,2n_y} f_{ss}(s',1) & \cdots & 0_{(n_x+n_y)\times n_y} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_y} & \cdots & p_{n_s,s'}\mathcal{D}_{n_y+1,2n_y} f_{ss}(s',n_s) \end{bmatrix}$$
$$+ \begin{bmatrix} p_{1,1}\mathcal{D}_{1,n_y} f_{ss}(1,1) & \cdots & p_{1,n_s}\mathcal{D}_{1,n_y} f_{ss}(n_s,1) \\ \vdots & \ddots & \vdots \\ p_{n_s,1}\mathcal{D}_{1,n_y} f_{ss}(1,n_s) & \cdots & p_{n_s,n_s}\mathcal{D}_{1,n_y} f_{ss}(n_s,n_s) \end{bmatrix},$$

$$\Phi_\chi = \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'}\mathcal{D}_{1,n_y} f_{ss}(s',1)\,\mathcal{D}_{1,n_x} g_{ss}(s') & \cdots & 0_{(n_x+n_y)\times n_x} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_x} & \cdots & p_{n_s,s'}\mathcal{D}_{1,n_y} f_{ss}(s',n_s)\,\mathcal{D}_{1,n_x} g_{ss}(s') \end{bmatrix}$$
$$+ \sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'}\mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',1) & \cdots & 0_{(n_x+n_y)\times n_x} \\ \vdots & \ddots & \vdots \\ 0_{(n_x+n_y)\times n_x} & \cdots & p_{n_s,s'}\mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',n_s) \end{bmatrix},$$

$$\text{and } \Psi_\chi = -\sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',1)\,\mathcal{D}\theta_{ss}(s') + \ldots \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',1)\,\mathcal{D}\theta_{ss}(1) \end{pmatrix} \\ \vdots \\ p_{n_s,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',n_s)\,\mathcal{D}\theta_{ss}(s') + \ldots \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',n_s)\,\mathcal{D}\theta_{ss}(n_s) \end{pmatrix} \end{bmatrix},$$

where $\mathcal{D}\theta_{ss}(s_t)$ denotes the derivative of $\theta(\chi, s_t)$ with respect to χ evaluated at $\chi = 0$ for all s_t. That is,

$$\mathcal{D}\theta_{ss}(s_t) = \mathcal{D}\theta(0, s_t) = \left[\mathcal{D}_j\theta^i(0, s_t)\right]_{1\le i\le n_\theta, j=1}$$

for all s_t.

Since Φ_ε and Φ_χ depend on $\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}$, we must solve (12) prior to solving (13). Since (13) is just a linear problem, we have that, first, there is a different solution to (13) for each solution to (12) and, second, the bottleneck to finding first-order approximations is obtaining $\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}$ by solving the quadratic system (12). The bottleneck will be fully discussed in Section 4.

3.3 Approximations

Given a solution to (12) and (13), it is straightforward to build a first-order approximation to the solutions of the MSDSGE model represented by Equation (1). Let a $g^{\mathbf{first}}$ and $h^{\mathbf{first}}$ be first-order approximations to g and h around the point $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$. It follows that

$$\begin{aligned} g^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - y_{ss} &= \left[\mathcal{D}_1 g_{ss}\left(s_t\right) \cdots \mathcal{D}_{n_x} g_{ss}\left(s_t\right)\right]\left(x_{t-1} - x_{ss}\right) \\ &+ \left[\mathcal{D}_{n_x+1} g_{ss}\left(s_t\right) \cdots \mathcal{D}_{n_x+n_\varepsilon} g_{ss}\left(s_t\right)\right]\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\chi \end{aligned}$$

and

$$\begin{aligned} h^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - x_{ss} &= \left[\mathcal{D}_1 h_{ss}\left(s_t\right) \cdots \mathcal{D}_{n_x} h_{ss}\left(s_t\right)\right]\left(x_{t-1} - x_{ss}\right) \\ &+ \left[\mathcal{D}_{n_x+1} h_{ss}\left(s_t\right) \cdots \mathcal{D}_{n_x+n_\varepsilon} h_{ss}\left(s_t\right)\right]\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\chi \end{aligned}$$

for all x_{t-1}, ε_t, and s_t. Hence, first-order approximations can be rewritten as

$$g^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - y_{ss} = \mathcal{D}_{1,n_x} g_{ss}\left(s_t\right)\left(x_{t-1} - x_{ss}\right) + \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}\left(s_t\right)\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\chi$$

and

$$h^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - x_{ss} = \mathcal{D}_{1,n_x} h_{ss}\left(s_t\right)\left(x_{t-1} - x_{ss}\right) + \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}\left(s_t\right)\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\chi$$

for all x_{t-1}, ε_t, and s_t. To express them in compact form, we have

$$g^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - y_{ss} = \mathcal{D} g_{ss}\left(s_t\right)\mathcal{S}_t,$$

where $\mathcal{S}_t = \begin{bmatrix} \left(x_{t-1} - x_{ss}\right)^\intercal & \varepsilon_t^\intercal & \chi \end{bmatrix}^\intercal$ and

$$h^{\mathbf{first}}\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) - x_{ss} = \mathcal{D} h_{ss}\left(s_t\right)\mathcal{S}_t$$

for all x_{t-1}, ε_t, χ, and s_t.

We summarize what we have developed and what our next task is. For first-order approximations, Section 3.2 lays out a procedure for obtaining the matrices $\mathcal{D} g_{ss}\left(s_t\right)$ and $\mathcal{D} h_{ss}\left(s_t\right)$ for all s_t. The bottleneck is to obtain $\left\{\mathcal{D}_{1,n_x} g_{ss}\left(s_t\right), \mathcal{D}_{1,n_x} h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$ by solving the quadratic system (12). Once the bottleneck is removed, the task of obtaining first-order approximations

only involves solving linear systems. As mentioned, the quadratic system has, in general, many solutions. Each solution corresponds to a different first-order approximation. Finding all the solutions to a quadratic system is central to ascertaining how many approximations are stable. Section 4 is devoted to dealing with this bottleneck by finding all the solutions to the quadratic system (12).

3.4 Feature of No Certainty Equivalence

As pointed out by Schmitt-Grohe and Uribe (2004), the certainty equivalence of first-order approximations is a main result for a constant parameter model. This result implies that first-order approximations to constant parameter models are inadequate for analyzing interesting behavior such as economic agents' responses to risk. For example, van Binsbergen et al. (2008) and Rudebusch and Swanson (2008) argue that, when using constant parameter models, at least second-order approximations are needed to analyze the effects of volatility on agents' decisions.

While second-order approximations nullify the result of certainty equivalence, they result in a substantially higher degree of computational difficulty in performing likelihood-based estimation, as documented by Fernández-Villaverde and Rubio-Ramirez (2007). We show in this section, however, that first-order approximations to the solutions of MSDSGE models are not necessarily certainty equivalent. This salient feature opens the door to analyzing risk-related behaviors using first-order approximations.

To see how certainty equivalence arises in first-order approximations in constant parameter models, consider Equation (13) with only one regime so that $n_s = 1$. In this degenerate case, we have

$$\begin{bmatrix} \Theta_\chi & \Phi_\chi \end{bmatrix} \begin{bmatrix} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(1) \\ \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(1) \end{bmatrix} = \Psi_\chi, \tag{14}$$

where

$$\begin{bmatrix} \Theta_\chi & \Phi_\chi \end{bmatrix} =$$

$$\begin{bmatrix} \mathcal{D}_{n_y+1,2n_y} f_{ss}(1,1) + \mathcal{D}_{1,n_y} f_{ss}(1,1) & \mathcal{D}_{1,n_y} f_{ss}(1,1)\, \mathcal{D}_{1,n_x} g_{ss}(1) + \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(1,1) \end{bmatrix}$$

$$\text{and } \Psi_\chi = -\left[\begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(1,1)\,\mathcal{D}\theta_{ss}(1) + \dots \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(1,1)\,\mathcal{D}\theta_{ss}(1) \end{pmatrix} \right].$$

Because $\theta(\chi,1) = \bar{\theta}$ in any constant parameter model, we have $\mathcal{D}_{ss}\theta(1) = 0_{n_\theta}$, implying that $\Psi_\chi = 0_{n_x+n_y}$. Therefore the linear system (14) is homogeneous. If a unique solution exists, it is given by

$$\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(1) = 0_{n_y} \text{ and } \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(1) = 0_{n_x}. \tag{15}$$

For the constant parameter case, first-order approximations to policy rules are

$$g^{\textbf{first}}(x_{t-1},\varepsilon_t,\chi,1) - y_{ss} = \mathcal{D}_{1,n_x} g_{ss}(1)(x_{t-1}-x_{ss}) + \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(1)\,\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(1)\,\chi$$

and

$$h^{\textbf{first}}(x_{t-1},\varepsilon_t,\chi,1) - x_{ss} = \mathcal{D}_{1,n_x} h_{ss}(1)(x_{t-1}-x_{ss}) + \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(1)\,\varepsilon_t + \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(1)\,\chi$$

for all x_{t-1} and ε_t. Using (15) and these policy rules evaluated at $x_{t-1} = x_{ss}$ and $\varepsilon_t = 0_{n_\varepsilon}$, we have

$$\begin{aligned} g^{\textbf{first}}(x_{ss},0_{n_\varepsilon},1,1) - y_{ss} &= 0_{n_y}, \\ h^{\textbf{first}}(x_{ss},0_{n_\varepsilon},1,1) - x_{ss} &= 0_{n_x}. \end{aligned}$$

That is, first-order approximations to the solutions of the constant parameter model are certainty equivalent.

With this insight we turn to the MSDSGE case. It is clear from Equation (13) that a necessary condition for first-order approximations not to display certainty equivalence is $\Psi_\chi \neq 0_{n_s(n_x+n_y)}$. Let us analyze the circumstance under which the condition $\Psi_\chi \neq 0_{n_s(n_x+n_y)}$ is true. For visual convenience, we rewrite the expression for Ψ_χ

$$\Psi_\chi = -\sum_{s'=1}^{n_s} \begin{bmatrix} p_{1,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',1)\,\mathcal{D}\theta_{ss}(s') + \dots \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',1)\,\mathcal{D}\theta_{ss}(1) \end{pmatrix} \\ \vdots \\ p_{n_s,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',n_s)\,\mathcal{D}\theta_{ss}(s') + \dots \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',n_s)\,\mathcal{D}\theta_{ss}(n_s) \end{pmatrix} \end{bmatrix}.$$

Clearly, if $\mathcal{D}\theta_{ss}(s_t) = 0_{n_\theta}$ for all s_t, then $\Psi_\chi = 0_{n_s(n_x+n_y)}$. Thus, a necessary condition for $\Psi_\chi \neq 0_{n_s(n_x+n_y)}$ to hold is $\mathcal{D}\theta_{ss}(s_t) \neq 0_{n_\theta}$ for some s_t. Given the partition of θ_t described in (5) and (6), we have

$$\mathcal{D}\theta_{ss}(s_t) = \begin{bmatrix} \widehat{\theta}_1(s_t)^\intercal & 0_{n_{\theta_2}}^\intercal \end{bmatrix}^\intercal.$$

It follows that $\mathcal{D}\theta_{ss}(s_t) \neq 0_{n_\theta}$ for some s_t if and only if $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ for some s_t. In sum, a necessary condition for $\Psi_\chi \neq 0_{n_s(n_x+n_y)}$ to hold is $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ for some s_t.

The condition $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ for some s_t, however, is insufficient for $\Psi_\chi \neq 0_{n_s(n_x+n_y)}$ to be true. A sufficient condition is

$$\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',s_t)\,\mathcal{D}\theta_{ss}(s') + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',s_t)\,\mathcal{D}\theta_{ss}(s_t) \end{pmatrix} \neq 0_{n_x+n_y}$$

for some s_t. This additional condition holds if θ_{1t} does not enter the equilibrium conditions multiplicatively with a variable to which the first partial derivative of f is zero when evaluated at the steady-state. The following proposition summarizes our findings.

Proposition 2 *First-order approximations to the solution of an MSDSGE model are not certainty equivalent if and only if both* $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ *and*

$$\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s',s_t)\,\mathcal{D}\theta_{ss}(s') + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s',s_t)\,\mathcal{D}\theta_{ss}(s_t) \end{pmatrix} \neq 0_{n_x+n_y}$$

for some s_t.

Proof. The "if" part of the proof has been provided in the analysis prior to the stated proposition.

For the "only if" part, we prove it by contradiction. Note that the only way for

$$\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) = 0_{n_y} \text{ and } \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) = 0_{n_x}$$

is for $\Psi_\chi = 0_{n_s(n_x+n_y)}$. Suppose $\widehat{\theta}_1(s_t) = 0_{n_{\theta_1}}$ for all s_t. Then $\Psi_\chi = 0_{n_s(n_x+n_y)}$ and

$$\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) = 0_{n_y} \text{ and } \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) = 0_{n_x}$$

for all s_t. It follows that first-order approximations are certainty equivalent.

Now suppose $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ for some s_t but

$$\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s', s_t)\, \mathcal{D}\theta_{ss}(s') + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s', s_t)\, \mathcal{D}\theta_{ss}(s_t) \end{pmatrix} = 0_{n_x+n_y}$$

for all s_t. In this case, it is straightforward to see that $\Psi_\chi = 0_{n_s(n_x+n_y)}$ and

$$\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) = 0_{n_y} \text{ and } \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) = 0_{n_x}$$

for all s_t. It follows that first-order approximations are certainty equivalent.

These contradictions establish the proof of the "only if" portion. ■

Proposition 2 implies that if first-order approximations are not certainty equivalent, approximated policy rules evaluated at $x_{t-1} = x_{ss}$ and $\varepsilon_t = 0_{n_\varepsilon}$ are either

$$g^{\textbf{first}}(x_{ss}, 0_{n_\varepsilon}, 1, s_t) - y_{ss} = \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) \neq 0_{n_y}$$

or

$$h^{\textbf{first}}(x_{ss}, 0_{n_\varepsilon}, 1, s_t) - x_{ss} = \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) \neq 0_{n_x}$$

for some s_t.

4 Generalized Quadratic System and New Solution Method

One main discovery in this paper is that we identify solving a system of quadratic equations as the sole bottleneck in obtaining first-order approximations to the solutions of MSDSGE models. Once this bottleneck is removed, we need solve only two systems of linear equations to obtain first-order approximations. In this section we show how to find all the solutions to this quadratic system by introducing a new method. Finding all the solutions is essential to determining how many first-order approximations are stable. Section 4.1 characterizes the quadratic system. Section 4.2 proposes a new solution method by applying the theory of Gröbner bases. We show that the number of solutions to the quadratic system is finite in most cases and define both the concept and the condition of stability to be used in the paper. Section 4.3 uses our RBC example to demonstrate how to apply our methodology.

4.1 Generalized Quadratic System

To understand the difficulty of solving a system of quadratic equations represented by (12), we begin with the constant parameter case in which $n_s = 1$. In this case, it is well understood that one can map this special system of quadratic equations to the generalized Schur decomposition problem related to the matrices A(1) and B(1). Thus, solving this special system of quadratic equations is equivalent to using the standard matrix algebra procedure to find all the solutions.

In the Markov-switching case when $n_s > 1$, there are n_s systems of quadratic equations represented by (12). Stacking these systems all together expands to a generalized system of $(n_y + n_x)\, n_s n_x$ quadratic equations with unknowns $\{\mathcal{D}_{1,n_x} g_{ss}(s), \mathcal{D}_{1,n_x} h_{ss}(s)\}_{s=1}^{n_s}$. Because $\{\mathcal{D}_{1,n_x} g_{ss}(s)\}_{s=1}^{n_s}$ appear in every system of the n_s systems, the generalized quadratic system can no longer be mapped to the generalized Schur decomposition problem.[9]

The quadratic system represented by (12) has multiple solutions in general.[10] Each solution implies a different first-order approximation. It is crucial that we find all the solutions and then determine how many imply stable approximations. For this purpose we develop, in the next section, a new method by applying Gröbner bases to our problem.

4.2 New Solution Method

4.2.1 Gröbner Bases

The quadratic system discussed in Section 4.1 is simply a system of quadratic polynomials. Gröbner bases provide a computationally practical means to obtain all the solutions to a system of polynomial equations. A more detailed description of Gröbner bases is provided in Appendix B. In this section we provide an intuitive explanation of how to apply Gröbner bases to solving a system of multivariate polynomials.

Suppose one wishes to find all the solutions to a system of n polynomial equations in n

[9] The intuition is that the generalized Schur decomposition cannot deal with the $2n_s$ matrices $A(1), \ldots, A(n_s)$ and $B(1), \ldots, B(n_s)$ when $n_s > 1$.

[10] For the remainder of the paper, we refer to the generalized quadratic system (12) as the quadratic system.

unknowns

$$f_1(x_1, \ldots, x_n) = 0, \ldots, f_n(x_1, \ldots, x_n) = 0.$$

There exist a number of computationally efficient routines that can transform the original system of n polynomial equations to an alternative system of n polynomial equations with the same set of solutions. The following proposition, known as the Shape Lemma, describes a useful transformation into a reduced Gröbner basis.

Proposition 3 *Let*

$$f_1(x_1, \ldots, x_n) = 0, \ldots, f_n(x_1, \ldots, x_n) = 0$$

be a system of n polynomials in n unknowns. Under certain regularity conditions, there exists the following set of n polynomials in n unknowns with the same set of roots:

$$x_1 - q_1(x_n) = 0, \ldots, x_{n-1} - q_{n-1}(x_n) = 0, q_n(x_n) = 0,$$

where each $q_i(x_n)$ is a univariate polynomial.

Proof. See Appendix B for details. ■

There are several important aspects to Proposition 3. First, the alternative set of polynomials in Proposition 3 is known as a Shape basis, a special kind of reduced Gröbner basis. Second, the regularity conditions, referred to in Proposition 3 and detailed in Appendix B, are satisfied by the quadratic system in most economic problems. Third, the roots of the univariate polynomial $q_n(x_n)$ can be found by any standard root finding algorithm. Fourth, once a root of $q_n(x_n)$ is found, one can easily compute $q_i(x_n)$ to obtain $x_1, \ldots, x_{n-1}$.

A large strand of the literature deals with the computation of reduced Gröbner bases. Buchberger (1998)'s algorithm is the original technique. Subsequently, many more efficient variants have been proposed. We refer the interested reader to Cox et al. (1997). In this paper we use Mathematica to find a Shape basis.

To illustrate how powerful Proposition 3 is, consider the following example featuring a system

of quadratic polynomials in four unknown variables $x_1, \ldots, x_4$:

$$\begin{aligned} x_1x_2 + x_3x_4 + 2 &= 0, \\ x_1x_2 + x_2x_3 + 3 &= 0, \\ x_1x_3 + x_4x_1 + x_4x_2 + 6 &= 0, \\ x_1x_3 + 2x_1x_2 + 3 &= 0. \end{aligned}$$

A Shape basis is

$$\begin{aligned} x_1 - \frac{1}{28}(9x_4^5 + 6x_4^3 - 15x_4) &= 0, \\ x_2 - \frac{1}{28}(-9x_4^5 - 6x_4^3 + 99x_4) &= 0, \\ x_3 - \frac{1}{14}(-3x_4^5 - 9x_4^3 - 2x_4) &= 0, \\ 3x_4^6 + 9x_4^4 - 19x_4^2 - 49 &= 0. \end{aligned}$$

The last polynomial is univariate of degree six in x_4. The six roots of this polynomial are

$$\{1.55461, -1.55461, 1.39592i, -1.39592i, 1.86232i, -1.86232i\}.$$

Each of these roots can be substituted into the first three equations to obtain the following six solutions:

$$\{x_1 = 2.89104,\ x_2 = 1.7728,\ x_3 = -4.58328,\ x_4 = 1.55461\},$$

$$\{x_1 = -2.89104,\ x_2 = -1.7728,\ x_3 = 4.58328,\ x_4 = -1.55461\},$$

$$\{x_1 = 0.372997i,\ x_2 = 3.81477i,\ x_3 = 0.41342i,\ x_4 = 1.39592i\},$$

$$\{x_1 = -0.372997i,\ x_2 = -3.81477i,\ x_3 = -0.41342i,\ x_4 = -1.39592i\},$$

$$\{x_1 = 4.81861i,\ x_2 = 0.768342i,\ x_3 = -0.914097i,\ x_4 = 1.86232i\},$$

$$\{x_1 == -4.81861i,\ x_2 = -0.768342i,\ x_3 = 0.914097i,\ x_4 = -1.86232i\}.$$

It is straightforward to show that these roots solve the original system of quadratic equations.

4.2.2 A Finite Number of Solutions

One of the regularity conditions for finding reduced Gröbner bases is that the quadratic system has finitely many solutions. This condition is met in most economic problems. To see why this result is true, let us first consider the constant parameter case when $n_s = 1$. If the solution of the model has a unique stable first-order approximation, the usual practice, as in Schmitt-Grohe and Uribe (2004), involves constructing this stable approximation by ordering the generalized eigenvalues of the two matrices $A(1)$ and $B(1)$ in a particular way. In general, however, the quadratic system has multiple solutions. Each solution implies a different first-order approximation. Some of these solutions may imply unstable approximations. For most economic problems, the full set of approximations (stable and unstable) can be found by changing the order of generalized eigenvalues. Hence, the number of approximations is related to the number of possible orderings of eigenvalues. Therefore, the number of solutions to the quadratic system, and therefore the number of approximations (stable and unstable), is bounded by

$$\begin{pmatrix} rank\,(A\,(1)) - n_{exo} \\ n_x - n_{exo} \end{pmatrix} = \frac{(rank\,(A\,(1)) - n_{exo})!}{(n_x - n_{exo})!\,(rank\,(A\,(1)) - n_x)!}, \tag{16}$$

where n_{exo} is the number of exogenous predetermined variables so that $0 \leq n_{exo} \leq n_x$, $rank\,(A\,(1))$ stands for the rank of the matrix $A\,(1)$.[11] This result is familiar to most readers.

Now consider the Markov-switching case when $n_s > 1$. We have the quadratic system or n_s quadratic systems of the form (12). If s_t were fixed for all t, the number of solutions to the quadratic system, and therefore the number of approximations (stable and unstable), would be bounded as in (16). Since we have n_s regimes, the total number of solutions to the quadratic

[11] Note that the matrix $A\,(1)$ may not be of full rank when there are redundant variables that are linearly dependent upon others and consequently can be eliminated from the quadratic system. A simple example is a leisureless RBC model with three variables (capital, consumption, and output) and three equations (the Euler condition, the resource constraint, and the output definition). If we eliminate the output definition equation and the output variable, the newly formed matrix $A(1)$ will be of full rank.

system for most parameter configurations is bounded by

$$\max_{s_t} \begin{pmatrix} rank\left(A\left(s_t\right)\right) - n_{exo} \\ n_x - n_{exo} \end{pmatrix}^{n_s} = \max_{s_t} \left(\frac{\left(rank\left(A\left(s_t\right)\right) - n_{exo}\right)!}{\left(n_x - n_{exo}\right)!\left(rank\left(A\left(s_t\right)\right) - n_x\right)!} \right)^{n_s}$$

where $rank\left(A\left(s_t\right)\right)$ stands for the rank of the matrix $A\left(s_t\right)$ for all s_t.[12]

Once we know that the number of solutions to the quadratic system is finite, we can use Mathematica to obtain reduced Gröbner bases for finding all solutions to the quadratic system and, hence, all first-order approximations. The next question is whether any of the approximations are stable and if so, how many. We address this question in the following section.

4.2.3 Mean Square Stability

In the constant parameter case, whether a first-order approximation is stable or not can be determined by verifying whether its largest absolute generalized eigenvalue is greater than or equal to one, a condition that holds for most concepts of stability. In the Markov-switching case, the problem is subtle and complicated, and there are alternative concepts of stability. Given a first-order approximation, we use the concept of mean square stability (MSS) as defined in Costa et al. (2005). Farmer et al. (2009) discuss several advantages of using the MSS concept over alternative ones such as the bounded stability. First, under MSS, even if the largest generalized eigenvalues associated with one particular regime is greater than one, the system, as a whole, can nonetheless be stable. Second, MSS permits applications in which the system has unbounded errors and hence unbounded state variables. This unbounded feature holds in our RBC example with the normally distributed shocks to TFP. Third, in the case of Markov-switching, necessary and sufficient conditions for the MSS are easily verifiable, whereas other stability concepts do

[12]For rare configurations of parameters, there may exist infinitely many solutions or the algorithm used by Mathematica to find reduced Gröbner bases ceases to converge. In such a case, numerical procedures can be used to solve the generalized quadratic system (12) to find one or possibly more solutions, although none of those proecdures is guaranteed to find all solutions. Appendix C describes one such numerical procedure. Alternative numerical methods, such as Newton's algorithm used in the RISE toolbox of Maih (2013) (https://github.com/jmaih/RISE_toolbox), may be used to find solutions, but again with no guarantee of finding all solutions.

not have such conditions. Specifically, the MSS requires verifying whether the following matrix has all its eigenvalues inside the unit circle

$$T = \left(P^{\intercal} \otimes I_{n_x^2}\right) \Upsilon, \tag{17}$$

where

$$\Upsilon = \begin{bmatrix} \mathcal{D}_{1,n_x} h_{ss}(1) \otimes \mathcal{D}_{1,n_x} h_{ss}(1) & \cdots & 0_{n_x^2 \times n_x^2} \\ 0_{n_x^2 \times n_x^2} & \cdots & 0_{n_x^2 \times n_x^2} \\ \vdots & \ddots & \vdots \\ 0_{n_x^2 \times n_x^2} & \cdots & \mathcal{D}_{1,n_x} h_{ss}(n_s) \otimes \mathcal{D}_{1,n_x} h_{ss}(n_s) \end{bmatrix}.$$

In the Markov-switching case, after we use a reduced Gröbner basis to obtain all the solutions for $\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}$, we verify, for each solution, whether the matrix Υ has all its eigenvalues less than one. If only one solution satisfies the MSS criterion, the model has a unique stable first-order approximation. If there is more than one solution that satisfies the MSS criterion, the model has multiple stable first-order approximations. If none of the solutions satisfies the MSS criterion or if there is no solution to the reduced Gröbner basis, the model does not have any stable first-order approximations. At this point, it is worth emphasizing that the stability of a second-order approximation (to be defined below) depends only on the stability of its first-order approximation component. In other words, stability depends only on the eigenvalues of the matrix Υ. The same is true for higher-order approximations.

4.3 The RBC model

At this junction we view it as instructive to use the previous RBC example to illustrate our methodology developed thus far. Consider the following parameterization:

α	β	δ	σ	$\bar{\mu}$	$\widehat{\mu}(1)$	$\widehat{\mu}(2)$	$p_{1,1}$	$p_{2,2}$
0.3300	0.9976	0.0250	0.0002	0.00333	0.00167	−0.00163	0.90	0.90

The growth rates $\bar{\mu}+\widehat{\mu}(1)$ and $\bar{\mu}+\widehat{\mu}(2)$ correspond to regimes where annual output growth rates are 3 percent and 1 percent respectively, β corresponds to a risk free annual rate of 3 percent in the steady-state, and σ is set to match the total volatility of TFP growth as estimated in

Fernández-Villaverde and Rubio-Ramirez (2007). The standard calibration of α implies a capital share of one third, and the value of δ implies an annual depreciation rate of approximately 10 percent, both in the steady-state. Note that regimes are symmetric in the sense that $p_{1,1} = p_{2,2}$.

Given this parameterization, the steady-state values of capital and consumption are $k_{ss} = 32.0986$ and $c_{ss} = 2.18946$. We compute the first partial derivatives of f with respect to all the variables evaluated at the steady-state:

$$\mathcal{D}_{1,1} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0.20861 \\ 0 \end{bmatrix}, \ \mathcal{D}_{2,2} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} -0.2086 \\ 1 \end{bmatrix}, \ \mathcal{D}_{3,3} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0.00031 \\ 1.00499 \end{bmatrix}$$

$$\mathcal{D}_{4,4} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 \\ -1.0074 \end{bmatrix}, \ \mathcal{D}_{5,5} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathcal{D}_{6,6} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0.00014 \\ 0.00900 \end{bmatrix}$$

$$\mathcal{D}_{7,7} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} -0.0147 \\ 0 \end{bmatrix}, \text{ and } \mathcal{D}_{8,8} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0.68169 \\ 44.9953 \end{bmatrix}$$

for all s_{t+1} and s_t.

Using these results, one can build the quadratic system as in (12) and solve for $\{\mathcal{D}_{1,1} g_{ss}(s_t), \mathcal{D}_{1,1} h_{ss}(s_t)\}_{s_t=1}^{n_s}$. Remember that in this example, $n_x = 1$, $n_y = 1$, and $n_s = 2$. The new method leads to the following four solutions

	$D_{1,1} h_{ss}(1)$	$D_{1,1} g_{ss}(1)$	$D_{1,1} h_{ss}(2)$	$D_{1,1} g_{ss}(2)$
(1)	0.96364	0.03896	0.96364	0.03896
(2)	1.04023	-0.0380	1.04023	-0.0380
(3)	$1.11326 + 0.116871i$	$-0.1114 - 0.11745i$	$1.11326 + 0.11687i$	$-0.1114 - 0.11745i$
(4)	$1.11326 - 0.116871i$	$-0.1114 + 0.11745i$	$1.11326 - 0.11687i$	$-0.1114 + 0.11745i$

Mathematica finds the four solutions in less than a hundredth of a second. Using the results in Section 4.2.3, one can verify that only solution (1) implies a stable first-order approximation under the MSS criterion. Normally, when you have a unique stable first-order approximation, you call it the first-order approximation. Thus, if we let $\hat{c}_t = c_t - c_{ss}$, $\hat{k}_{t-1} = k_{t-1} - k_{ss}$, and

$\mathcal{S}_t = \begin{bmatrix} \hat{k}_{t-1} & \varepsilon_t & \chi \end{bmatrix}^{\intercal}$, the first-order approximation to the solution of the model is

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03896 & 0.00028 & 0.00972 \\ 0.96364 & -0.0092 & -0.0843 \end{bmatrix} S_t$$

if $s_t = 1$, and

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03896 & 0.00028 & -0.00972 \\ 0.96364 & -0.0092 & 0.0843 \end{bmatrix} \mathcal{S}_t$$

if $s_t = 2$, where the rest of the derivatives used to form the first-order approximation can be obtained by solving the linear system defined in (13).

The approximation highlights Proposition 2. First-order approximations are, in general, not certainty equivalent. Since $\widehat{\mu}(s_t) \neq 0$ for all s_t and

$$\sum_{s'=1}^{n_s} p_{s_t,s'} \left(\mathcal{D}_{7,7} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s') + \mathcal{D}_{8,8} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s_t) \right) \neq 0_2$$

for all s_t, first partial derivatives of g and h with respect to χ will be nonzero. That is, $\mathcal{D}_3 g_{ss}(s_t) \neq 0$ and $\mathcal{D}_3 h_{ss}(s_t) \neq 0$.

5 Second-Order Approximations

Having shown how to construct first-order approximations to the solutions of the MSDSGE model, we now show how to construct second-order approximations. In Section 5.1, we introduce additional notation. In Section 5.2, we use the notation to derive second-order approximations to the model solution. We show that given first-order approximations, second-order approximations can be obtained by simply solving linear systems. This fact emphasizes that the bottleneck to find both first-order and second-order approximations is solving the quadratic system defined by (12). Section 5.3 returns to the RBC model for illustration.

5.1 Additional Notation

5.1.1 Second Partial Derivatives of f

We begin with additional notation in regard to second partial derivatives of f^i. Denote

$$\mathcal{H}f^i_{ss}\left(s_{t+1}, s_t\right) =$$
$$\left[\mathcal{D}_k\mathcal{D}_j f^i\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2\left(s_{t+1}\right), \overline{\theta}_1, \widehat{\theta}_2\left(s_t\right)\right)\right]_{1\leq j,k\leq 2(n_y+n_x+n_\varepsilon+n_\theta)}$$

for all s_{t+1} and s_t as the $2\left(n_y+n_x+n_\varepsilon+n_\theta\right)\times 2\left(n_y+n_x+n_\varepsilon+n_\theta\right)$ matrix of second partial derivatives of f^i for $1\leq i\leq n_y+n_x$ with respect to all its variables evaluated at

$$\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2\left(s_{t+1}\right), \overline{\theta}_1, \widehat{\theta}_2\left(s_t\right)\right).$$

To conserve space, we do not represent the second partial derivatives of f for our RBC example (they are available upon request).

5.1.2 Second Partial Derivatives of $\mathbb{G}$

Let

$$\mathcal{H}\mathbb{G}^i\left(x_{t-1}, \varepsilon_t, \chi, s_t\right) = \left[\mathcal{D}_k\mathcal{D}_j\mathbb{G}^i\left(x_{t-1}, \varepsilon_t, \chi, s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

be the $\left(n_x+n_\varepsilon+1\right)\times\left(n_x+n_\varepsilon+1\right)$ matrix of second partial derivatives of $\mathbb{G}^i$ with respect to $\left(x_{t-1}, \varepsilon_t, \chi\right)$ for all x_{t-1}, ε_t, and s_t and $1\leq i\leq n_y+n_x$. It follows that

$$\mathcal{H}\mathbb{G}^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) = \left[\mathcal{D}_k\mathcal{D}_j\mathbb{G}^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

is the $\left(n_x+n_\varepsilon+1\right)\times\left(n_x+n_\varepsilon+1\right)$ matrix of second partial derivatives of $\mathbb{G}^i$ with respect to $\left(x_{t-1}, \varepsilon_t, \chi\right)$ evaluated at $\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)$ for all s_t and $1\leq i\leq n_y+n_x$.

To simplify notation we define

$$\mathcal{H}\mathbb{G}^i_{ss}\left(s_t\right) = \mathcal{H}\mathbb{G}^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right) \text{ and } \mathcal{D}_k\mathcal{D}_j\mathbb{G}^i_{ss}\left(s_t\right) = \mathcal{D}_k\mathcal{D}_j\mathbb{G}^i\left(x_{ss}, 0_{n_\varepsilon}, 0, s_t\right)$$

for all s_t and $1\leq i\leq n_y+n_x$ and $1\leq j,k\leq n_x+n_\varepsilon+1$. Thus,

$$\mathcal{H}\mathbb{G}^i_{ss}\left(s_t\right) = \left[\mathcal{D}_k\mathcal{D}_j\mathbb{G}^i_{ss}\left(s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

for all s_t and $1\leq i\leq n_y+n_x$.

5.1.3 Second Partial Derivatives of g and h

We now introduce the second partial derivatives of g and h. Let

$$\mathcal{H}g^i\left(x_{t-1},\varepsilon_t,\chi,s_t\right)=\left[\mathcal{D}_k\mathcal{D}_jg^i\left(x_{t-1},\varepsilon_t,\chi,s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

be the $(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)$ matrix of second partial derivatives of g^i with respect to $(x_{t-1},\varepsilon_t,\chi)$ for all x_{t-1},ε_t, and s_t and $1\leq i\leq n_y$. It follows that

$$\mathcal{H}g^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)=\left[\mathcal{D}_k\mathcal{D}_jg^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

is the $(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)$ matrix of second partial derivatives of g^i with respect to $(x_{t-1},\varepsilon_t,\chi)$ evaluated at $(x_{ss},0_{n_\varepsilon},0,s_t)$ for all s_t and $1\leq i\leq n_y$.

To put in compact notation we define

$$\mathcal{H}g^i_{ss}\left(s_t\right)=\mathcal{H}g^i_{ss}\left(x_{ss},0_{n_\varepsilon},0,s_t\right)\text{ and }\mathcal{D}_k\mathcal{D}_jg^i_{ss}\left(s_t\right)=\mathcal{D}_k\mathcal{D}_jg^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)$$

for all s_t and $1\leq i\leq n_y$ and $1\leq j,k\leq n_x+n_\varepsilon+1$. Hence,

$$\mathcal{H}g^i_{ss}\left(s_t\right)=\left[\mathcal{D}_k\mathcal{D}_jg^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

for all s_t and $1\leq i\leq n_y$.

Let

$$\mathcal{H}h^i\left(x_{t-1},\varepsilon_t,\chi,s_t\right)=\left[\mathcal{D}_k\mathcal{D}_jh^i\left(x_{t-1},\varepsilon_t,\chi,s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

be the $(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)$ matrix of second partial derivatives of h^i with respect to $(x_{t-1},\varepsilon_t,\chi)$ for all x_{t-1},ε_t, and s_t and $1\leq i\leq n_x$. It follows that

$$\mathcal{H}h^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)=\left[\mathcal{D}_k\mathcal{D}_jh^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)\right]_{1\leq j,k\leq n_x+n_\varepsilon+1}$$

is the $(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)$ matrix of second partial derivatives of h^i with respect to $(x_{t-1},\varepsilon_t,\chi)$ evaluated at $(x_{ss},0_{n_\varepsilon},0,s_t)$ for all s_t and $1\leq i\leq n_x$.

If we define

$$\mathcal{H}h^i_{ss}\left(s_t\right)=\mathcal{H}h^i_{ss}\left(x_{ss},0_{n_\varepsilon},0,s_t\right)\text{ and }\mathcal{D}_k\mathcal{D}_jh^i_{ss}\left(s_t\right)=\mathcal{D}_k\mathcal{D}_jh^i\left(x_{ss},0_{n_\varepsilon},0,s_t\right)$$

for all s_t, $1 \leq i \leq n_x$, and $1 \leq j, k \leq n_x + n_\varepsilon + 1$, then we have

$$\mathcal{H}h^i_{ss}(s_t) = \left[\mathcal{D}_k \mathcal{D}_j h^i(x_{ss}, 0_{n_\varepsilon}, 0, s_t)\right]_{1 \leq j,k \leq n_x + n_\varepsilon + 1}$$

for all s_t and $1 \leq i \leq n_x$.

With these definitions, the second partial derivatives of g and h evaluated at the steady-state can be expressed as

$$\mathcal{H}g_{ss}(s_t) = \begin{bmatrix} vec\left(\mathcal{H}g^1_{ss}(s_t)\right)^\intercal \\ \vdots \\ vec\left(\mathcal{H}g^{n_y}_{ss}(s_t)\right)^\intercal \end{bmatrix}$$

and

$$\mathcal{H}h_{ss}(s_t) = \begin{bmatrix} vec\left(\mathcal{H}h^1_{ss}(s_t)\right)^\intercal \\ \vdots \\ vec\left(\mathcal{H}h^{n_x}_{ss}(s_t)\right)^\intercal \end{bmatrix}$$

for all s_t.

5.2 Approximations

Let $g^{\textbf{second}}$ and $h^{\textbf{second}}$ be a second-order approximation to g and h around the point $(x_{ss}, 0_{n_\varepsilon}, 0, s_t)$. With the additional notation introduced in Section 5.1, we have

$$g^{\textbf{second}}(x_{t-1}, \varepsilon_t, \chi, s_t) - y_{ss} = \mathcal{D}g_{ss}(s_t)\,\mathcal{S}_t + \frac{1}{2}\mathcal{H}g_{ss}(s_t)\left(\mathcal{S}_t \otimes \mathcal{S}_t\right),$$

where $\mathcal{S}_t = \begin{bmatrix} (x_{t-1} - x_{ss})^\intercal & \varepsilon_t^\intercal & \chi \end{bmatrix}^\intercal$ is an $(n_x + n_\varepsilon + 1) \times 1$ vector. Similarly, we have

$$h^{\textbf{second}}(x_{t-1}, \varepsilon_t, \chi, s_t) - x_{ss} = \mathcal{D}h_{ss}(s_t)\,\mathcal{S}_t + \frac{1}{2}\mathcal{H}h_{ss}(s_t)\left(\mathcal{S}_t \otimes \mathcal{S}_t\right).$$

Similar to our analysis in Section 3.2, given the first-order approximations, the remaining task is to derive the matrices

$$\left\{\left\{\mathcal{H}g^i_{ss}(s_t)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}h^i_{ss}(s_t)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s} \tag{18}$$

for all s_t, where $\mathcal{H}g^i_{ss}(s_t) \in \mathbb{C}^{(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)}$ for all s_t and $1 \leq i \leq n_y$ and $\mathcal{H}h^i_{ss}(s_t) \in \mathbb{C}^{(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)}$ for all s_t and and $1 \leq i \leq n_x$.

As in the case of first-order approximations, we use the chain rule and the second partial derivatives of $\mathbb{G}\left(x_{t-1},\varepsilon_t,\chi,s_t\right)$ to obtain the systems of equations that can be used to solve for the Hessian matrices expressed in (18). Since

$$\mathbb{G}\left(x_{t-1},\varepsilon_t,\chi,s_t\right)=0_{n_y+n_x}$$

for all $x_{t-1},\varepsilon_t,\chi$, and s_t, it must be the case that

$$\mathcal{H}\mathbb{G}^i\left(x_{t-1},\varepsilon_t,\chi,s_t\right)=0_{(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)}$$

for all $x_{t-1},\varepsilon_t,\chi$, s_t and $1\leq i\leq n_y+n_x$, and in particular

$$\mathcal{H}\mathbb{G}^i_{ss}\left(s_t\right)=0_{(n_x+n_\varepsilon+1)\times(n_x+n_\varepsilon+1)}$$

for all s_t and $1\leq i\leq n_y+n_x$.[13] It follows that

$$\begin{aligned}
\mathcal{H}_{1,n_x;1,n_x}\mathbb{G}^i_{ss}\left(s_t\right) &= 0_{n_x\times n_x},\ \mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}\left(s_t\right)=0_{n_x\times n_\varepsilon},\\
\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}\left(s_t\right) &= 0_{n_x},\ \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}\left(s_t\right)=0_{n_\varepsilon\times n_\varepsilon},\\
\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}\left(s_t\right) &= 0_{n_\varepsilon},\text{ and }\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}\left(s_t\right)=0
\end{aligned}\tag{19}$$

for all s_t and $1\leq i\leq n_y+n_x$, where we have used the following definition

$$\mathcal{H}_{n_1,n_2;m_1,m_2}\mathbb{G}^i_{ss}\left(s_t\right)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}^i_{ss}\left(s_t\right) & \dots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}\mathbb{G}^i_{ss}\left(s_t\right)\\ \vdots & \ddots & \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}^i_{ss}\left(s_t\right) & \dots & \mathcal{D}_{n_2}\mathcal{D}_{m_2}\mathbb{G}^i_{ss}\left(s_t\right)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<n_2,m_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_y+n_x$.[14]

[13] By Young's Theorem, all the Hessian matrices in (18) and $\left\{\mathcal{H}\mathbb{G}^i_{ss}\left(s_t\right)\right\}_{i=1}^{n_x+n_y}$ for all s_t are symmetric. In the following, we exploit this fact whenever it is applicable and focus on only the relevant portion of the Hessian matrices.

[14] When $m_1=m_2$ we have

$$\mathcal{H}_{n_1,n_2;m_1}\mathbb{G}^i_{ss}\left(s_t\right)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}^i_{ss}\left(s_t\right)\\ \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}^i_{ss}\left(s_t\right)\end{bmatrix}$$

All the equations in the systems represented in (19) depend on

$$\left\{\mathcal{D}_{1,n_x} g_{ss}\left(s_t\right), \mathcal{D}_{1,n_x} h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}, \quad \left\{\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}\left(s_t\right), \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s},$$
$$\text{and } \left\{\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right), \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}.$$

Thus, one must obtain first-order approximations prior to obtaining second-order approximations. The following steps describe how to use the systems in (19) to obtain the Hessian matrices expressed in (18) and highlight the dependence between these systems.

Step 1 The condition $\mathcal{H}_{1,n_x;1,n_x}\mathbb{G}_{ss}^i\left(s_t\right) = 0_{n_x\times n_x}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{1,n_x;1,n_x} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{1,n_x;1,n_x} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

Step 2 Given the solution from Step 1, the condition $\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}^i\left(s_t\right) = 0_{n_x\times n_\varepsilon}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

Step 3 Given the solution from Step 2, the condition $\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}\mathbb{G}_{ss}^i\left(s_t\right) = 0_{n_x}$ for $1 \leq i \leq n_y+n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

Step 4 Given the solution from Step 1, the condition $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}^i\left(s_t\right) = 0_{n_\varepsilon\times n_\varepsilon}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

Step 5 Given the solutions from Steps 1 and 3, the condition $\mathcal{H}_{n_x+1,n_x+n_\varepsilon,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}_{ss}^i\left(s_t\right) = 0_{n_\varepsilon}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

---

for all s_t, $1 \leq n_1, m_1 < n_2 \leq n_x + n_\varepsilon + 1$, and $1 \leq i \leq n_y + n_x$. When $n_1 = n_2$ we have

$$\mathcal{H}_{n_1;m_1,m_2}\mathbb{G}_{ss}^i\left(s_t\right) = \left[\begin{array}{ccc} \mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}_{ss}^i\left(s_t\right) & \ldots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}\mathbb{G}_{ss}^i\left(s_t\right) \end{array}\right]$$

for all s_t, $1 \leq n_1, m_1 < m_2 \leq n_x + n_\varepsilon + 1$, and $1 \leq i \leq n_y + n_x$. When $m_1 = m_2$ and $n_1 = n_2$ we have

$$\mathcal{H}_{n_1;m_1}\mathbb{G}_{ss}^i\left(s_t\right) = \mathcal{D}_{n_1}\mathcal{D}_{m_1}\mathbb{G}_{ss}^i\left(s_t\right)$$

for all s_t, $1 \leq n_1, m_1 \leq n_x + n_\varepsilon + 1$, and $1 \leq i \leq n_y + n_x$.

Step 6 Given the solutions from Steps 1, 3 and 4, the condition $\mathcal{H}_{n_x+n_\varepsilon+1,n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t)=0$ for $1\leq i\leq n_y+n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{n_x+n_\varepsilon+1,n_x+n_\varepsilon+1}g^i_{ss}(s_t)\right\}_{i=1}^{n_y},\left\{\mathcal{H}_{n_x+n_\varepsilon+1,n_x+n_\varepsilon+1}h^i_{ss}(s_t)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

In the steps described above, we have used the following definitions:

$$\mathcal{H}_{n_1,n_2;m_1,m_2}g^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t) & \dots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}g^i_{ss}(s_t)\\ \vdots & \ddots & \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t) & \dots & \mathcal{D}_{n_2}\mathcal{D}_{m_2}g^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<n_2,m_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_y$;

$$\mathcal{H}_{n_1,n_2;m_1,m_2}h^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t) & \dots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}h^i_{ss}(s_t)\\ \vdots & \ddots & \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t) & \dots & \mathcal{D}_{n_2}\mathcal{D}_{m_2}h^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<n_2,m_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_x$.[15]

[15] When $m_1=m_2$, we have

$$\mathcal{H}_{n_1,n_2;m_1}g^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t)\\ \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<n_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_y$;

$$\mathcal{H}_{n_1,n_2;m_1}h^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t)\\ \vdots\\ \mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<n_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_x$. When $n_1=n_2$, we have

$$\mathcal{H}_{n_1;m_1,m_2}g^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t) & \dots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}g^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<m_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_y$;

$$\mathcal{H}_{n_1;m_1,m_2}h^i_{ss}(s_t)=\begin{bmatrix}\mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t) & \dots & \mathcal{D}_{n_1}\mathcal{D}_{m_2}h^i_{ss}(s_t)\end{bmatrix}$$

for all s_t, $1\leq n_1,m_1<m_2\leq n_x+n_\varepsilon+1$, and $1\leq i\leq n_x$. When $m_1=m_2$ and $n_1=n_2$, we have

$$\mathcal{H}_{n_1;m_1}g^i_{ss}(s_t)=\mathcal{D}_{n_1}\mathcal{D}_{m_1}g^i_{ss}(s_t)$$

Appendix D provides a detailed description of the derivations for Steps 1 to 6. In particular, it derives the second partial derivatives of $\mathbb{G}$ needed to construct the six systems. It shows that once we obtain first-order approximations to the model solution, each of the equations in Steps 1 to 6 becomes linear. As a result, for each first-order approximation, a second-order approximation can be easily derived from the set of linear systems described above.

In addition, Appendix D shows that if first-order approximations are certainty equivalent, the systems of equations represented by $\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t)$ (cross partial derivatives between x_{t-1} and χ) and $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}(s_t)$ (cross partial derivatives between ε_t and χ) for all s_t and $1 \leq i \leq n_y + n_x$ are homogeneous in the unknown variables

$$\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}g^i_{ss}(s_t),\ \mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}h^j_{ss}(s_t),$$
$$\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}g^i_{ss}(s_t),\ \text{and}\ \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}h^j_{ss}(s_t)$$

for all s_t, $1 \leq i \leq n_y$, and $1 \leq j \leq n_x$. This property is formally stated in the following proposition.

Proposition 4 *If first-order approximations are certainty equivalent, then the systems of equations*

$$\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t) = 0_{n_x}\ \textit{and}\ \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t) = 0_{n_x}$$

for all s_t and $1 \leq i \leq n_y + n_x$ are homogeneous in their unknown variables and hence

$$\begin{aligned}\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}g^i_{ss}(s_t) &= 0_{n_x},\ \mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}h^j_{ss}(s_t) = 0_{n_x},\\ \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}g^i_{ss}(s_t) &= 0_{n_\varepsilon}\ \textit{and}\ \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}h^j_{ss}(s_t) = 0_{n_\varepsilon}\end{aligned}$$

for all s_t, $1 \leq i \leq n_y$, and $1 \leq j \leq n_x$.

Proof. If first-order approximations are certainty equivalent, one can see from Proposition 2 that

$$\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s_t) = 0_{n_y}\ \text{and}\ \mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t) = 0_{n_x}$$

for all s_t, $1 \leq n_1, m_1 \leq n_x + n_\varepsilon + 1$, and $1 \leq i \leq n_y$;

$$\mathcal{H}_{n_1;m_1}h^i_{ss}(s_t) = \mathcal{D}_{n_1}\mathcal{D}_{m_1}h^i_{ss}(s_t)$$

for all s_t, $1 \leq n_1, m_1 \leq n_x + n_\varepsilon + 1$, and $1 \leq i \leq n_x$.

for all s_t. The proof follows directly from the above results. ■

In the constant parameter case, the cross partial derivatives with respect to χ and other state variables are always zero because $\widehat{\theta}_1(1) = 0_{n_{\theta_1}}$. In the Markov-switching case, Proposition 4 implies that if $\widehat{\theta}_1(s_t) \neq 0_{n_{\theta_1}}$ for some s_t, then it may be the case that $\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} g^i_{ss}(s_t) \neq 0_{n_x}$, $\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} h^j_{ss}(s_t) \neq 0_{n_x}$, $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} g^i_{ss}(s_t) \neq 0_{n_\varepsilon}$ or $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} h^j_{ss}(s_t) \neq 0_{n_\varepsilon}$ for some s_t, some $1 \leq i \leq n_y$, and some $1 \leq j \leq n_x$. These non-trivial extra terms may result in more accurate second-order approximations.

5.3 Returning to the RBC Model

Let us now return to the RBC example. As shown above, given our parameterization, there is only one stable first-order approximation. Thus, there is only one stable second-order approximation. When that is the case, we call it the second-order approximation. We can find the second-order approximation by solving a set of linear systems described in Steps 1 to 6. We have

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03896 & 0.00028 & 0.00972 \\ 0.96364 & -0.0092 & -0.0843 \end{bmatrix} \mathcal{S}_t$$
$$+\frac{1}{2} \begin{bmatrix} -0.0004 & 4\times10^{-6} & 0.00016 & 4\times10^{-6} & 4\times10^{-8} & 1\times10^{-6} & 0.00016 & 1\times10^{-6} & -0.0003 \\ -0.0002 & -0.0003 & -0.0025 & -0.0003 & 2.7\times10^{-6} & 2\times10^{-5} & -0.0025 & 0.00002 & 0.00057 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

if $s_t = 1$ or

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03896 & 0.00028 & -0.00972 \\ 0.96364 & -0.0092 & 0.0843 \end{bmatrix} \mathcal{S}_t$$
$$+\frac{1}{2} \begin{bmatrix} -0.0004 & 4\times10^{-6} & -0.0002 & 4\times10^{-6} & 4\times10^{-8} & -1\times10^{-6} & -0.0002 & -1\times10^{-6} & -0.0003 \\ -0.0002 & -0.0002 & 0.00251 & -0.0003 & 3\times10^{-6} & -2\times10^{-5} & 0.00251 & -2\times10^{-5} & 0.00057 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

if $s_t = 2$.

Since the first-order approximation is not certainty equivalent in this example, the second-order approximation highlights the result in Proposition 4. The cross derivatives of g and h with respect to χ and other state variables evaluated at the steady-state are

$$\mathcal{H}_{i,3} g_{ss}(s_t) = \mathcal{H}_{3,i} g_{ss}(s_t) \neq 0, \ \mathcal{H}_{i,3} h_{ss}(s_t) = \mathcal{H}_{3,i} h_{ss}(s_t) \neq 0$$

for all $s_t = 1, 2$ and $1 \leq i \leq 2$. Moreover, since regimes are symmetric, the following terms are symmetric across regimes

$$\begin{aligned}
\mathcal{D}_3 g_{ss}(1) &= -\mathcal{D}_3 g_{ss}(2), \mathcal{D}_3 h_{ss}(1) = \mathcal{D}_3 h_{ss}(2), \\
\mathcal{H}_{i,3} g_{ss}(1) &= -\mathcal{H}_{i,3} g_{ss}(2), \mathcal{H}_{i,3} g_{ss}(1) = -\mathcal{H}_{i,3} g_{ss}(2), \\
\mathcal{H}_{i,3} h_{ss}(1) &= -\mathcal{H}_{i,3} h_{ss}(2), \text{ and } \mathcal{H}_{i,3} h_{ss}(1) = -\mathcal{H}_{i,3} h_{ss}(2)
\end{aligned}$$

for $1 \leq i \leq 2$.

6 Applications

Since our theoretical results are new, the purpose of this section is to guide the reader by illustrating how to apply our methodology in practice. For this purpose we first continue the previous RBC model with new parameterization and then study two versions of the New-Keynesian model.

6.1 Continuing the RBC Model

Stochastic neoclassical growth models, such as the one discussed in previous sections, are the foundation of modern macroeconomics. Understanding how to solve MSDSGE models of this prototype would enable us to work on richer models such as those commonly used for policy analysis. For pedagogical reasons we consider the shock standard deviation, σ, to be constant across regimes in all the examples studied in the paper. But our approach can be easily extended, without much computational burden, to cases allowing for Markov-switching volatilities.

The previous RBC example restricts regimes to be symmetric such that $p_{1,1} = p_{2,2}$. As a result, the cross partial derivatives of g and h evaluated at the steady-state are symmetric across regimes. In this section, we consider an asymmetric-regime case in which $p_{1,1} \neq p_{2,2}$. As will be shown, the cross partial derivatives of g and h evaluated at the steady-state are asymmetric across regimes in this case.

Consider the same parameter configuration as in Section 4.3, except $p_{1,1} = 0.5$, $\bar{\mu} = 0.00222$, $\mu(1) = 0.00278$, and $\mu(2) = 0.00052$. In this case, Regime 1 has a shorter expected duration, and Regime 2 occurs more frequently in the ergodic distribution. With this alternative parameter configuration, the steady-state is represented by $k_{ss} = 34.6774$, and $c_{ss} = 2.24769$ and the first partial derivatives of f evaluated at this steady-state are different from those in the symmetric-regime case.[16] Using these results we can solve the quadratic system (12).

There are four solutions to the quadratic system under this alternative parameter configuration

	$D_{1,1}h_{ss}(1)$	$D_{1,1}g_{ss}(1)$	$D_{1,1}h_{ss}(2)$	$D_{1,1}g_{ss}(2)$
(1)	0.96545	0.0370821	0.96545	0.0370821
(2)	1.03828	-0.035996	1.03828	-0.035996
(3)	$2.00373 - 0.7042i$	$-1.00465 + 0.70654i$	$1.11318 + 0.39122i$	$-0.111145 - 0.39252i$
(4)	$2.00373 + 0.7042i$	$-1.00465 - 0.70654i$	$1.11318 - 0.39122i$	$-0.111145 + 0.39252i$

As before, Mathematica finds the four solutions in less than a hundredth of a second. Solution (1) is the only one associated with a stable first-order approximation. Remember that the rest of the coefficients necessary to construct the second-order approximation can be obtained by solving linear systems and that, given that the first-order approximation is stable, the second-order approximation is also stable. The second-order approximation to the model solution is

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03708 & 0.00029 & 0.00637 \\ 0.96545 & -0.0100 & -0.1412 \end{bmatrix} S_t$$
$$+\frac{1}{2}\begin{bmatrix} -0.0004 & 4\times10^{-6} & 0.00009 & 4\times10^{-6} & 5\times10^{-8} & 1\times10^{-6} & 0.00009 & 1\times10^{-6} & -5\times10^{-6} \\ -0.0002 & -0.0003 & -0.0040 & -0.0003 & 3\times10^{-6} & 0.00004 & -0.0040 & 0.00004 & 0.00065 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

if $s_t = 1$ and

$$\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} 0.03708 & 0.00029 & -0.0013 \\ 0.96545 & -0.0100 & 0.02823 \end{bmatrix} \mathcal{S}_t$$
$$+\frac{1}{2}\begin{bmatrix} -0.0004 & 4\times10^{-6} & -1\times10^{-5} & 4\times10^{-6} & 5\times10^{-8} & -2\times10^{-7} & -1\times10^{-5} & -2\times10^{-7} & -6\times10^{-5} \\ -0.0002 & -0.0003 & 0.00080 & -0.0003 & 3\times10^{-6} & -8\times10^{-6} & 0.00080 & -8\times10^{-6} & 0.00009 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

if $s_t = 2$.

[16] In the interest of space, we omit many matrices from this section. They are available upon request.

Clearly, the first-order approximation can be constructed by only considering the first matrix of the right hand side the above expressions.

As in the symmetric regime case, the first-order approximation is not certainty equivalent and the cross derivatives of g and h with respect to χ and other state variables evaluated at the steady-state are non-zero. These results follow directly from Propositions 2 and 4. Because $\widehat{\mu}(s_t) \neq 0$ for all s_t, we have that

$$\sum_{s'=1}^{n_s} p_{s_t,s'} \left(\mathcal{D}_{7,7} f_{ss}(s', s_t)\, \mathcal{D}\theta_{ss}(s') + \mathcal{D}_{8,8} f_{ss}(s', s_t)\, \mathcal{D}\theta_{ss}(s_t)\right) \neq 0_2$$

for all s_t. Consequently, $\mathcal{D}_3 g_{ss}(s_t) \neq 0$, $\mathcal{D}_3 h_{ss}(s_t) \neq 0$, and

$$\mathcal{H}_{i,3} g_{ss}(s_t) = \mathcal{H}_{3,i} g_{ss}(s_t) \neq 0,\ \mathcal{H}_{i,3} h_{ss}(s_t) = \mathcal{H}_{3,i} h_{ss}(s_t) \neq 0$$

for all $s_t = 1, 2$ and $1 \leq i \leq 2$.

The results show that $\widehat{\mu}(s_t)$ plays a crucial role in determining whether the first-order approximations are certainty equivalent and whether the cross partial derivatives of g and h with respect to χ and other state variables evaluated at the steady-state are zero. Unlike the symmetric regime case, however, the cross partial derivatives of g and f at the steady-state are not the same across regimes. Specifically,

$$\begin{aligned}
\mathcal{D}_3 g_{ss}(1) &\neq -\mathcal{D}_3 g_{ss}(2),\ \mathcal{D}_3 h_{ss}(1) \neq \mathcal{D}_3 h_{ss}(2), \\
\mathcal{H}_{i,3} g_{ss}(1) &\neq -\mathcal{H}_{i,3} g_{ss}(2),\ \mathcal{H}_{3,i} g_{ss}(1) \neq -\mathcal{H}_{3,i} g_{ss}(2), \\
\mathcal{H}_{i,3} h_{ss}(1) &\neq -\mathcal{H}_{i,3} h_{ss}(2), \text{ and } \mathcal{H}_{3,i} h_{ss}(1) \neq -\mathcal{H}_{3,i} h_{ss}(2)
\end{aligned}$$

for $1 \leq i \leq 3$.

Given the first-order and second-order approximations, we can compute and compare their accuracy using Euler equation errors as suggested in Judd (1998) and Aruoba et al. (2006). The Euler equation error at point $\left(\tilde{k}_{t-1}, \varepsilon_t; s_t\right)$ for the approximation *order* $\in$ {first, second} is

$$EE^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right) =$$

$$1 - \beta \int \sum_{s'=1}^{2} p_{s_t,s'} \left(\begin{array}{c} \frac{\tilde{c}^{order}\left(\tilde{k}_{t-1},\varepsilon_t,1;s_t\right)}{\tilde{c}^{order}\left(\tilde{k}^{order}\left(\tilde{k}_{t-1},\varepsilon_t,1;s_t\right),\varepsilon';1\right)} \exp\left(\frac{\mu(s_t)+\sigma\varepsilon_t}{\alpha-1}\right) \\ \times \left(\alpha \exp\left(\mu(s') + \sigma\varepsilon'\right) \tilde{k}^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right)^{\alpha-1} + 1 - \delta\right) \end{array} \right) \mu(\varepsilon')\, d\varepsilon'$$

where $\tilde{c}^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right)$ and $\tilde{k}^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right)$ are the appropriate approximations to the policy functions. The unconditional absolute Euler equation error is

$$EE^{order} = \sum_{s_t} \int \left| EE^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right) \right| \mu^{order}\left(\tilde{k}_{t-1}, \varepsilon_t, 1; s_t\right) d\tilde{k}_{t-1} d\varepsilon_t$$

where μ^{order} is the unconditional distribution of the variables $\tilde{k}$ and ε.

To approximate this integral numerically, we simulate a long path for the economy, saving $\tilde{k}_t$, ε_t, and s_t along the path. The simulation length is 10,000 periods, with the first 1,000 periods discarded as a burn-in. Then, for each state $\left(\tilde{k}_{t-1}, \varepsilon_t, s_t\right)$, we draw 10,000 normally distributed ε_{t+1}'s to compute the expectation over ε_{t+1}, and use the transition values $p_{s_t, s_{t+1}}$ to compute the expectation over s_{t+1}. This entire process takes approximately an hour.

The following table shows the base-10 logarithms of absolute values of Euler equation errors for the first-order and second-order approximations in both symmetric and asymmetric cases. Both the first-order and second-order approximations produce a high degree of accuracy: a value of -5 implies an error of \$1 for each \$100,000 of consumption. As can be seen, the second-order approximation achieves an even higher degree of accuracy.

Unconditional Absolute Euler Equation Errors ($\log_{10}$)

	$p_{1,1} = 0.9$	$p_{1,1} = 0.5$
$EE^{\mathbf{first}}$	-4.6099	-5.3929
$EE^{\mathbf{second}}$	-5.4158	-6.1983

6.2 A New-Keynesian Model

New-Keynesian models, such as those in Woodford (2003) and Christiano et al. (2005), are often used in policy analysis. This section discusses a version of the New-Keynesian model, illustrates how to partition the vector of Markov-switching parameters θ_t, and discusses the conditions under which multiple stable approximations exist.

6.2.1 The Model

We consider a New-Keynesian model with quadratic price adjustment costs. The monetary authority follows a Taylor Rule. There are two Markov-switching parameters: the technology drift parameter and the coefficient on inflation in the Taylor rule. Davig and Leeper (2007), Farmer et al. (2011), and Bianchi (2010), among others, argue that Markov-switching in the Taylor rule coefficient on inflation captures a switch in U.S. policy regime since the mid 1980s.

The model features the representative consumer maximizing the expected lifetime utility over consumption C_t and hours worked H_t

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left(\log C_t - H_t\right)$$

subject to the budget constraint

$$C_t + \frac{B_t}{P_t} = W_t H_t + R_{t-1}\frac{B_{t-1}}{P_t} + T_t + D_t,$$

where B_t is nominal bonds, W_t is the real wage, R_{t-1} is the nominal return on bonds, T_t is lump-sum transfers, and D_t is profits from firms. The first-order conditions are

$$1 = \beta \mathbb{E}_t \frac{C_t}{C_{t+1}} \frac{R_t}{\Pi_{t+1}}$$

$$\text{and } C_t = W_t.$$

The competitive final goods producer combines a continuum of intermediate goods $Y_{j,t}$ into a final good Y_t according to the constant elasticity of substitution (CES) aggregation technology

$$Y_t = \left(\int_0^1 Y_{j,t}^{\frac{\eta-1}{\eta}} dj\right)^{\frac{\eta}{\eta-1}}.$$

Intermediate goods are produced by firms taking the wage and the demand function

$$Y_{j,t} = \left(\frac{P_{j,t}}{P_t}\right)^{-\eta} Y_t$$

as given. The price $P_{j,t}$ is set and hours $H_{j,t}$ are demanded according to

$$Y_{j,t} = A_t H_{j,t},$$

where A_t is a technology shock following the law of motion

$$\log A_t = \mu_t + \log A_{t-1},$$

where, similar to the RBC model, the drift μ_t takes two discrete values dictated by the Markov chain process represented by $s_t \in \{1, 2\}$.

These intermediate-goods firms face quadratic price adjustment costs according to

$$AC_{j,t} = \frac{\kappa}{2}\left(\frac{P_{j,t}}{P_{j,t-1}} - 1\right)^2 Y_t.$$

The firms maximize the expected discounted profits subject to the demand function, the production function, and adjustment costs. In the symmetric equilibrium, $P_{j,t} = P_t$, $Y_{j,t} = Y_t$, and $H_{j,t} = H_t$ for all j, and the optimality conditions are

$$W_t = mc_t A_t$$

$$\text{and } \kappa\left(\Pi_t - 1\right)\Pi_t = (1-\eta) + \eta mc_t + \beta\kappa\mathbb{E}_t\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{C_t}{C_{t+1}}\frac{Y_{t+1}}{Y_t},$$

where mc_t is the marginal cost faced by the firms.

The monetary authority sets the interest rate R_t by the following Taylor rule

$$\frac{R_t}{R_{ss}} = \left(\frac{R_{t-1}}{R_{ss}}\right)^{\rho}\Pi_t^{(1-\rho)\psi_t}\exp\left(\sigma\varepsilon_t\right)$$

where the coefficient on inflation, ψ_t, takes two discrete values dictated by the same Markov chain process represented by s_t.

The market clearing condition is

$$Y_t = C_t + \frac{\kappa}{2}\left(\Pi_t - 1\right)^2 Y_t.$$

Substituting out mc_t, W_t, and C_t leads to the equilibrium conditions

$$1 = \beta\mathbb{E}_t\frac{\left(1 - \frac{\kappa}{2}\left(\Pi_t - 1\right)^2\right)}{\left(1 - \frac{\kappa}{2}\left(\Pi_{t+1} - 1\right)^2\right)}\frac{Y_t}{Y_{t+1}}\frac{R_t}{\Pi_{t+1}},$$

$$\kappa\left(\Pi_t - 1\right)\Pi_t = (1-\eta) + \eta\left(1 - \frac{\kappa}{2}\left(\Pi_t - 1\right)^2\right)\frac{Y_t}{A_t} + \beta\mathbb{E}_t\kappa\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{\left(1 - \frac{\kappa}{2}\left(\Pi_t - 1\right)^2\right)}{\left(1 - \frac{\kappa}{2}\left(\Pi_{t+1} - 1\right)^2\right)},$$

$$\text{and } \frac{R_t}{R_{ss}} = \left(\frac{R_{t-1}}{R_{ss}}\right)^{\rho} \Pi_t^{(1-\rho)\psi_t} \exp\left(\sigma\varepsilon_t\right).$$

Since the technology shock has a unit root, the model is non-stationary. To obtain a stationary equilibrium, we define $\tilde{Y}_t = Y_t/A_t$, which leads to the stationary equilibrium conditions as

$$1 = \beta\mathbb{E}_t \frac{\left(1-\frac{\kappa}{2}\left(\Pi_t-1\right)^2\right)\tilde{Y}_t}{\left(1-\frac{\kappa}{2}\left(\Pi_{t+1}-1\right)^2\right)\tilde{Y}_{t+1}} \frac{1}{\exp\left(\mu_{t+1}\right)} \frac{R_t}{\Pi_{t+1}},$$

$$\kappa\left(\Pi_t-1\right)\Pi_t = (1-\eta) + \eta\left(1-\frac{\kappa}{2}\left(\Pi_t-1\right)^2\right)\tilde{Y}_t + \beta\mathbb{E}_t\kappa\frac{\left(1-\frac{\kappa}{2}\left(\Pi_t-1\right)^2\right)}{\left(1-\frac{\kappa}{2}\left(\Pi_{t+1}-1\right)^2\right)}\left(\Pi_{t+1}-1\right)\Pi_{t+1},$$

$$\text{and } \frac{R_t}{R_{ss}} = \left(\frac{R_{t-1}}{R_{ss}}\right)^{\rho} \Pi_t^{(1-\rho)\psi_t} \exp\left(\sigma\varepsilon_t\right).$$

We partition θ_t into $\theta_{1t} = \mu_t$ and $\theta_{2t} = \psi_t$. We have

$$\mu_t = \mu\left(\chi, s_t\right) = \overline{\mu} + \chi\widehat{\mu}\left(s_t\right)$$

$$\text{and } \psi_t = \psi\left(\chi, s_t\right) = \widehat{\psi}\left(s_t\right).$$

Hence, the drift parameter μ_t depends on the perturbation parameter χ, while the Taylor-rule coefficient on inflation ψ_t does not. We choose this partition because μ_t would enter the definition of steady-state in a constant parameter model, while ψ_t would not.

Using the notation in Section 2, $y_t = \left[\Pi_t, \tilde{Y}_t\right]^{\intercal}$ and $x_{t-1} = R_{t-1}$, we express the stationary equilibrium condition as

$$f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right) =$$

$$\begin{bmatrix} 1-\beta\frac{\left(1-\frac{\kappa}{2}(\Pi_t-1)^2\right)\tilde{Y}_t}{\left(1-\frac{\kappa}{2}(\Pi_{t+1}-1)^2\right)\tilde{Y}_{t+1}}\frac{1}{\exp\left(\mu_{t+1}\right)}\frac{R_t}{\Pi_{t+1}} \\ (1-\eta)+\eta\left(1-\frac{\kappa}{2}\left(\Pi_t-1\right)^2\right)\tilde{Y}_t + \beta\kappa\frac{\left(1-\frac{\kappa}{2}(\Pi_t-1)^2\right)}{\left(1-\frac{\kappa}{2}(\Pi_{t+1}-1)^2\right)}\left(\Pi_{t+1}-1\right)\Pi_{t+1} - \kappa\left(\Pi_t-1\right)\Pi_t \\ \left(\frac{R_{t-1}}{R_{ss}}\right)^{\rho}\Pi_t^{(1-\rho)\psi_t}\exp\left(\sigma\varepsilon_t\right) - \frac{R_t}{R_{ss}} \end{bmatrix}.$$

The model solution takes the form of

$$\begin{bmatrix} \Pi_t & \tilde{Y}_t \end{bmatrix}^{\intercal} = g\left(R_{t-1}, \varepsilon_t, \chi, s_t\right),$$

$$\begin{bmatrix} \Pi_{t+1} & \tilde{Y}_{t+1} \end{bmatrix}^{\intercal} = g\left(R_t, \chi\varepsilon_{t+1}, \chi, s_{t+1}\right),$$

$$\text{and } R_t = h\left(R_{t-1}, \varepsilon_t, \chi, s_t\right).$$

6.2.2 Solving the Model

The following subsections show the sequential steps of obtaining approximations to the solution of the model. First, we find the steady-state. Second, we derive the first partial derivatives of f with respect to all its variables evaluated at steady-state. Third, we construct second-order approximations to the solution. We present the results for two parameter configurations to show how to use reduced Gröbner bases to obtain approximations and the MSS criterion to ascertain whether we have a unique stable approximation.

Steady-State To obtain the steady-state, we set $\chi = 0$ and $\varepsilon_t = 0$, so that $\Pi_t = \Pi_{t+1} = \Pi_{ss}$, $\tilde{Y}_t = \tilde{Y}_{t+1} = \tilde{Y}_{ss}$, $R_t = R_{t-1} = R_{ss}$, and $\mu_{t+1} = \mu_t = \bar{\mu}$. Thus the equilibrium conditions in the steady-state become

$$\begin{bmatrix} 1 - \beta \frac{\left(1-\frac{\kappa}{2}(\Pi_{ss}-1)^2\right)\tilde{Y}_{ss}}{\left(1-\frac{\kappa}{2}(\Pi_{ss}-1)^2\right)\tilde{Y}_{ss}} \frac{1}{\exp(\bar{\mu})} \frac{R_{ss}}{\Pi_{ss}} \\ (1-\eta) + \eta\left(1 - \frac{\kappa}{2}\left(\Pi_{ss}-1\right)^2\right)\tilde{Y}_{ss} + \beta\kappa \frac{\left(1-\frac{\kappa}{2}(\Pi_{ss}-1)^2\right)}{\left(1-\frac{\kappa}{2}(\Pi_{ss}-1)^2\right)}\left(\Pi_{ss}-1\right)\Pi_{ss} - \kappa\left(\Pi_{ss}-1\right)\Pi_{ss} \\ \left(\frac{R_{ss}}{R_{ss}}\right)^{\rho} \Pi_{ss}^{(1-\rho)\psi_t} - \frac{R_{ss}}{R_{ss}} \end{bmatrix} = 0_{3\times 1}.$$

Assuming $\Pi_{ss} = 1$, we have the steady-state values as

$$R_{ss} = \frac{\exp\left(\bar{\mu}\right)}{\beta}, \ \tilde{Y}_{ss} = \frac{\eta - 1}{\eta}.$$

This result confirms our partition such that $\bar{\mu}$ affects the steady-state, while ψ_t does not. In principle, we could let $\theta_{1t} = (\mu_t, \psi_t)$ so that perturbation would apply to ψ_t as well. To obtain a numerical solution that is as accurate as possible, however, one should keep the number of perturbed parameters at the minimum. The converse is not true. That is, one cannot let $\theta_{2t} = (\mu_t, \psi_t)$ because the steady-state in the constant parameter case would depend on technology drift.

Partial Derivatives of f In this example, $n_y = 2$, $n_x = 1$, $n_\varepsilon = 1$, and $n_\theta = 2$. Thus, the first partial derivatives of f with respect to all its variables evaluated at the steady-state are

$$\mathcal{D}_{1,2} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} \frac{\eta}{\eta-1} & 1 \\ 0 & \beta\kappa \\ 0 & 0 \end{bmatrix}, \mathcal{D}_{3,4} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} \frac{\eta}{1-\eta} & 0 \\ \eta & -\kappa \\ 0 & (1-\rho)\widehat{\psi}(s) \end{bmatrix}$$

$$\mathcal{D}_{5,5} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} -\beta e^{-\bar{\mu}} \\ 0 \\ -\beta e^{-\bar{\mu}} \end{bmatrix}, \mathcal{D}_{6,6} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 \\ 0 \\ \rho\beta e^{-\bar{\mu}} \end{bmatrix}, \mathcal{D}_{7,7} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathcal{D}_{8,8} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 \\ 0 \\ \sigma \end{bmatrix}, \mathcal{D}_{9,10} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathcal{D}_{11,12} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

for all s_{t+1} and s_t. Note that $\widehat{\psi}(s_t)$ enters the matrices because we do not perturb ψ_t.

Unique Stable Approximation Consider the following parameterization

β	κ	η	ρ	σ	$\bar{\mu}$	$\hat{\mu}(1)$	$\hat{\mu}(2)$	$\widehat{\psi}(1)$	$\widehat{\psi}(2)$	$p_{1,1}$	$p_{2,2}$
0.9976	161	10	0.8	0.0025	0.005	0.0025	−0.0025	3.1	0.9	0.90	0.90

The growth rates $\bar{\mu} + \hat{\mu}(1)$ and $\bar{\mu} + \hat{\mu}(2)$ correspond to regimes where the annual growth rate are 3 percent and 1 percent respectively, β corresponds to an annual risk-free rate of 3 percent, η has the steady-state markup of 11 percent, and ρ and σ match the estimates in Fernandez-Villaverde et al. (2009). The two monetary policy parameters $\widehat{\psi}(1)$ and $\widehat{\psi}(2)$ are such that $\widehat{\psi}(1)$ would imply a unique stable approximation if $\psi_t = \widehat{\psi}(1)$ for all t and $\widehat{\psi}(2)$ would lead to multiple stable approximations if $\psi_t = \widehat{\psi}(2)$ for all t.

Given this parameter configuration, we can calculate the steady-state values of the nominal rate and output as $R_{ss} = 1.0074$ and $\tilde{Y}_{ss} = 0.90$. The resulting first partial derivatives of f with respect to all its variables evaluated at the steady-state are

$$\mathcal{D}_{1,2} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} 1.11111 & 1 \\ 0 & 160.614 \\ 0 & 0 \end{bmatrix}, \mathcal{D}_{3,4} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} -1.11111 & 0 \\ 10 & -161. \\ 0 & 0.2\widehat{\psi}(s_t) \end{bmatrix},$$

$$\mathcal{D}_{5,5}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}-0.9926\\0\\-0.9926\end{bmatrix},\mathcal{D}_{6,6}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0\\0\\0.77941\end{bmatrix},\mathcal{D}_{7,7}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

$$\mathcal{D}_{8,8}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0\\0\\0.0025\end{bmatrix},\mathcal{D}_{9,10}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}1&0\\0&0\\0&0\end{bmatrix},\text{ and }\mathcal{D}_{11,12}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0&0\\0&0\\0&0\end{bmatrix}$$

for all s_{t+1} and s_t.

In this example, we obtain the following nine solutions for $\left\{\mathcal{D}_{1,1}g_{ss}\left(s_t\right),\mathcal{D}_{1,1}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$

	$D_{1,1}h_{ss}(1)$	$D_{1,1}g_{ss}(1)^{\intercal}$		$D_{1,1}h_{ss}(2)$	$D_{1,1}g_{ss}(2)^{\intercal}$	
(1)	0.59517	-1.92815	-0.327932	0.699414	-2.9541	-0.554689
(2)	0.77508	-3.64018	-0.0398952	1.3018	-7.43725	2.76721
(3)	0.79559	-1.82393	-0.00706061	1.05423	1.21892	1.40196
(4)	$1.0939-0.4363i$	$-0.8264+4.2641i$	$0.4706-0.6986i$	$1.3311+0.0574i$	$-10.008-1.9739i$	$2.9287+0.3165i$
(5)	$1.0939+0.4363i$	$-0.8264-4.2641i$	$0.4706+0.6986i$	$1.3311-0.0574i$	$-10.008+1.9739i$	$2.9287-0.3165i$
(6)	$1.0952-0.2105i$	$-0.9833+1.9595i$	$0.4727-0.3370i$	$1.0240-0.0200i$	$0.8689+0.7833i$	$1.2351-0.1103i$
(7)	$1.0952+0.2105i$	$-0.9833-1.9595i$	$0.4727+0.3370i$	$1.0240+0.0200i$	$0.8689-0.7833i$	$1.2351+0.1103i$
(8)	$1.2360-0.2511i$	$0.7554+3.0821i$	$0.6980-0.4020i$	$0.7507+0.0047i$	$-2.2696+0.6345i$	$-0.2718+0.0260i$
(9)	$1.2360+0.2511i$	$0.7554-3.0821i$	$0.6980+0.4020i$	$0.7507-0.0047i$	$-2.2696-0.6345i$	$-0.2718-0.0260i$

Mathematica finds the nine solutions in less than three hundredths of a second. The only solution that produces a stable first-order approximation is (1) and hence there is a unique stable approximation. Given solution (1), we can solve the linear systems that allow us to obtain the second-order approximation. Let $\widehat{\tilde{Y}}_t=\tilde{Y}_t-Y_{ss}$, $\hat{\Pi}_t=\Pi_t-\Pi_{ss}$, $\hat{R}_t=R_t-R_{ss}$, and define $S_t=\begin{bmatrix}\hat{R}_{t-1} & \varepsilon_t & \chi\end{bmatrix}^{\intercal}$. The second-order approximation is

$$\begin{bmatrix}\widehat{\tilde{Y}}_t\\\hat{\Pi}_t\\\hat{R}_t\end{bmatrix}=\begin{bmatrix}-1.9282&-0.0062&0.00481\\-0.3279&-0.0011&0.00014\\0.59517&0.00191&0.00008\end{bmatrix}\mathcal{S}_t$$

$$+\frac{1}{2}\begin{bmatrix}21.3771&0.06247&-0.0188&0.06247&0.00020&-0.0001&-0.0188&0.00020&-0.0004\\0.49793&0.00056&-0.0008&0.00056&2\times10^{-6}&-2\times10^{-6}&-0.0008&2\times10^{-6}&-0.0003\\-0.1986&0.00124&-0.0004&0.00124&4\times10^{-6}&-1\times10^{-6}&-0.0004&4\times10^{-6}&-0.0002\end{bmatrix}\left(\mathcal{S}_t\otimes\mathcal{S}_t\right)$$

if $s_t = 1$ and

$$\begin{bmatrix} \widehat{\widehat{Y}}_t \\ \hat{\Pi}_t \\ \hat{R}_t \end{bmatrix} = \begin{bmatrix} -2.9541 & -0.0090 & -0.0094 \\ -0.5547 & -0.0017 & -0.0026 \\ 0.69941 & 0.00214 & -0.0006 \end{bmatrix} \mathcal{S}_t$$

$$+\frac{1}{2}\begin{bmatrix} 56.9733 & 0.16487 & 0.23174 & 0.16487 & 0.00050 & 0.00071 & 0.23174 & 0.00050 & -0.0016 \\ 0.99333 & 0.00136 & 0.00033 & 0.00136 & 4\times10^{-6} & 1\times10^{-6} & 0.00033 & 4\times10^{-6} & -0.0010 \\ -0.1842 & 0.00160 & -0.0005 & 0.00160 & 5\times10^{-6} & -2\times10^{-6} & -0.0005 & 5\times10^{-6} & -0.0002 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

if $s_t = 2$.

Again, the first-order approximation is easily obtained from the above expressions. As in the RBC model, there is no certainty equivalence and the cross derivatives of g and h with respect to χ and other state variables evaluated at the steady-state are non-zero. Note that the first partial derivatives of g and h with respect to R_{t-1} and ε_t are different across regimes. This result occurs because ψ_t affects the first derivatives of f. Perturbing ψ_t would make these derivatives equal and the approximation less accurate.

Accuracy: Euler Equation Errors The Euler equation error at the point $(R_{t-1}, \varepsilon_t; s_t)$ is that for $order \in \{\text{first, second}\}$,

$$EE^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right) = 1 - \beta \int \sum_{s'=1}^{2} p_{s_t,s'} \times$$
$$\left(\begin{array}{c} \frac{1-\frac{\kappa}{2}\left(\Pi^{order}(R_{t-1},\varepsilon_t,1;s_t)-1\right)^2}{1-\frac{\kappa}{2}\left(\Pi^{order}\left(R^{order}(R_{t-1},\varepsilon_t,1;s_t),\varepsilon_{t+1},1;s_{t+1}\right)-1\right)^2} \times \\ \frac{\tilde{Y}^{order}(R_{t-1},\varepsilon_t,1;s_t)}{\tilde{Y}^{order}\left(R^{order}(R_{t-1},\varepsilon_t,1;s_t),\varepsilon_{t+1},1;s_{t+1}\right)} \times \\ \frac{1}{\exp(\mu')} \frac{R^{order}(R_{t-1},\varepsilon_t,1;s_t)}{\Pi^{order}\left(R^{order}(R_{t-1},\varepsilon_t,1;s_t),\varepsilon_{t+1},1;s_{t+1}\right)} \end{array} \right) \mu\left(\varepsilon'\right) d\varepsilon'$$

where $\tilde{Y}^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right)$, $\Pi^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right)$, and $R^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right)$ are the approximations to the policy functions. The unconditional absolute Euler equation error is

$$EE^{order} = \sum_{s_t} \int \left|EE^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right)\right| \mu^{order}\left(R_{t-1}, \varepsilon_t, 1; s_t\right) dR_{t-1} d\varepsilon_t$$

where μ^{order} is the unconditional distribution of the variables R and ε.

Numerical approximation of this integral simulates a $10,000$-period path, with the first $1,000$ periods discarded as a burn-in. For each state $(R_{t-1}, \varepsilon_t, s_t)$ along the simulated path, we draw

10,000 normally distributed ε_{t+1}'s to compute the expectation over ε_{t+1}, and use the transition values $p_{s_t,s_{t+1}}$ to compute the expectation over s_{t+1}. This entire process takes about an hour.

The following table shows the base-10 logarithms of absolute Euler equation errors for the first-order and second-order approximations. Both the first-order and the second-order approximations produce a high degree of accuracy: a value of -4 implies an error of \$1 for each \$10,000 of consumption.

Unconditional Absolute Euler Equation Errors ($\log_{10}$)	
$EE^{\textbf{first}}$	-3.7395
$EE^{\textbf{second}}$	-4.7485

Multiple Stable Approximations As an alternative parameter configuration, we consider the same parameters as in the previous section except $\widehat{\psi}(2) = 0.7$. That is, the second regime now has a slightly lower response by the monetary authority to inflation. In this case, the steady-state is still the same. As before, Regime 1 would imply a unique stable approximation if considered in isolation, whereas Regime 2 would imply multiple stable approximations.

The first partial derivatives of f with respect to all its variables evaluated at the steady-state are the same as those in the previous section except

$$\mathcal{D}_{3,4} f_{ss}(s_{t+1}, s_t) = \begin{bmatrix} -1.11111 & 0 \\ 10 & -161. \\ 0 & 0.2\widehat{\psi}(s_t) \end{bmatrix}$$

for all s_{t+1} and s_t, which depends on $\widehat{\psi}(s_t)$. Using these results, we obtain the following 9

solutions for $\{\mathcal{D}_{1,1}g_{ss}(s_t), \mathcal{D}_{1,1}h_{ss}(s_t)\}_{s_t=1}^{n_s}$.

	$D_{1,1}h_{ss}(1)$	$D_{1,1}g_{ss}(1)^{\intercal}$		$D_{1,1}h_{ss}(2)$	$D_{1,1}g_{ss}(2)^{\intercal}$	
(1)	0.59067	-1.9452	-0.3351	0.71244	-3.2185	-0.6209
(2)	0.79733	-4.7813	-0.0043	1.32443	-11.313	3.71833
(3)	0.85231	-1.7727	0.08374	1.01525	2.03718	1.52618
(4)	$1.0444 - 0.4989i$	$-0.7000 + 4.6327i$	$0.3912 - 0.7987i$	$1.3523 + 0.0442i$	$-14.210 - 1.8733i$	$3.9161 + 0.3132i$
(5)	$1.0444 + 0.4989i$	$-0.7000 - 4.6327i$	$0.3912 + 0.7987i$	$1.3523 - 0.0442i$	$-14.210 + 1.8733i$	$3.9161 - 0.3132i$
(6)	$1.0670 - 0.1894i$	$-1.2845 + 1.5032i$	$0.4274 - 0.3033i$	$0.9995 - 0.0089i$	$1.6635 + 0.5703i$	$1.4141 - 0.0629i$
(7)	$1.0670 + 0.1894i$	$-1.2845 - 1.5032i$	$0.4274 + 0.3033i$	$0.9995 + 0.0089i$	$1.6635 - 0.5703i$	$1.4141 + 0.0629i$
(8)	$1.2374 - 0.2527i$	$0.8018 + 3.0980i$	$0.7004 - 0.4046i$	$0.7582 + 0.0045i$	$-2.3764 + 0.6699i$	$-0.2963 + 0.0317i$
(9)	$1.2374 + 0.2527i$	$0.8018 - 3.0980i$	$0.7004 + 0.4046i$	$0.7582 - 0.0045i$	$-2.3764 - 0.6699i$	$-0.2963 - 0.0317i$

Now we have two solutions – (1) and (3) – that produce stable approximations. Thus the solution to the model has multiple stable approximations. This example illustrates how we can use reduced Gröbner bases and the MSS criterion to determine the number of stable approximations.

6.3 A New-Keynesian Model with Habit

Now consider a slight variant of the previously discussed New-Keynesian model, but with the representative household having habit formation. We consider this application because deriving first-order and second-order approximations is more involved when habit formation is present and the reader would be interested in knowing how to accomplish this task. The household in this economy maximizes

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left(\log\left(C_t - \varphi C_{t-1}\right) - H_t\right)$$

where φ denotes habit persistence. Letting λ_t denote the Lagrange multiplier on the budget constraint, the first-order conditions are

$$\lambda_t = \frac{1}{C_t - \varphi C_{t-1}} - \beta \mathbb{E}_t \frac{\varphi}{C_{t+1} - \varphi C_t},$$

$$\lambda_t = \beta \mathbb{E}_t \lambda_{t+1} \frac{R_t}{\Pi_{t+1}},$$

$$\text{and } 1 = \lambda_t W_t.$$

Final good and intermediate goods producers face problems identical to the previous example and

$$\log A_t = \mu_t + \log A_{t-1}.$$

Thus, the firm's optimality conditions are

$$W_t = mc_t A_t$$

$$\text{and } \kappa\left(\Pi_t - 1\right)\Pi_t = (1-\eta) + \eta mc_t + \beta\kappa\mathbb{E}_t\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{\lambda_{t+1}}{\lambda_t}\frac{Y_{t+1}}{Y_t}.$$

The resource constraint is

$$Y_t = C_t + \frac{\kappa}{2}\left(\Pi_t - 1\right)^2 Y_t.$$

For clarity we assume that the monetary authority does not smooth interest rates and follows the rule

$$R_t = R_{ss}\Pi_t^{\psi_t}\exp\left(\sigma\varepsilon_t\right).$$

With habits, consumption appears at different dates, as C_{t-1}, C_t, and C_{t+1}, in the equilibrium conditions. Substituting out mc_t, W_t, Y_t, and R_t, we have the equilibrium conditions as

$$\lambda_t = \frac{1}{C_t - \varphi C_{t-1}} - \beta\mathbb{E}_t\frac{\varphi}{C_{t+1} - \varphi C_t},$$

$$\lambda_t = \beta\mathbb{E}_t\lambda_{t+1}\frac{R_{ss}\Pi_t^{\psi_t}}{\Pi_{t+1}},$$

$$\text{and } \kappa\left(\Pi_t - 1\right)\Pi_t = (1-\eta) + \frac{\eta}{A_t\lambda_t} + \beta\kappa\mathbb{E}_t\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{\lambda_{t+1}}{\lambda_t}\frac{C_{t+1}}{C_t}\frac{1 - \frac{\kappa}{2}\left(\Pi_t - 1\right)^2}{1 - \frac{\kappa}{2}\left(\Pi_{t+1} - 1\right)^2}.$$

The economy is nonstationary because of the presence of the unit root. If we define $\tilde{\lambda}_t = \lambda_t A_t$ and $\tilde{C}_t = C_t/A_t$, we have the stationary equilibrium conditions as

$$\tilde{\lambda}_t = \frac{1}{\tilde{C}_t - \varphi\exp\left(-\mu_t\right)\tilde{C}_{t-1}} - \beta\mathbb{E}_t\frac{\varphi}{\tilde{C}_{t+1}\exp\left(\mu_{t+1}\right) - \varphi\tilde{C}_t},$$

$$\tilde{\lambda}_t = \beta\mathbb{E}_t\frac{\tilde{\lambda}_{t+1}}{\exp\left(\mu_{t+1}\right)}\frac{R_{ss}\Pi_t^{\psi_t}\exp\left(\sigma\varepsilon_t\right)}{\Pi_{t+1}},$$

$$\text{and } \kappa\left(\Pi_t - 1\right)\Pi_t = (1-\eta) + \frac{\eta}{\tilde{\lambda}_t} + \beta\kappa\mathbb{E}_t\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{\tilde{\lambda}_{t+1}}{\tilde{\lambda}_t}\frac{\tilde{C}_{t+1}}{\tilde{C}_t}\frac{1 - \frac{\kappa}{2}\left(\Pi_t - 1\right)^2}{1 - \frac{\kappa}{2}\left(\Pi_{t+1} - 1\right)^2}.$$

To solve the model, we define the two auxiliary variables $\tilde{X}_t = \tilde{C}_t$, and $\tilde{X}_{t+1} = \tilde{C}_{t+1}$. Using the notation in Section 2, we have $y_t = \begin{bmatrix} \Pi_t & \tilde{X}_t & \tilde{\lambda}_t \end{bmatrix}^\intercal$, $x_{t-1} = \tilde{C}_{t-1}$, $\theta_{2t} = \psi_t$, and the

equilibrium conditions are

$$f\left(y_{t+1}, y_t, x_t, x_{t-1}, \chi\varepsilon_{t+1}, \varepsilon_t, \theta_{t+1}, \theta_t\right) =$$

$$\begin{bmatrix} \frac{1}{\tilde{C}_t - \varphi \exp(-\mu_t)\tilde{C}_{t-1}} - \beta \frac{\varphi}{\tilde{X}_{t+1}\exp\left(\mu_{t+1}\right) - \varphi\tilde{C}_t} - \tilde{\lambda}_t \\ \beta \frac{\tilde{\lambda}_{t+1}}{\exp\left(\mu_{t+1}\right)} \frac{R_{ss}\Pi_t^{\psi_t}\exp(\sigma\varepsilon_t)}{\Pi_{t+1}} - \tilde{\lambda}_t \\ (1-\eta) + \frac{\eta}{\tilde{\lambda}_t} + \beta\kappa\left(\Pi_{t+1} - 1\right)\Pi_{t+1}\frac{\tilde{\lambda}_{t+1}}{\tilde{\lambda}_t}\frac{\tilde{X}_{t+1}}{\tilde{C}_t}\frac{1-\frac{\kappa}{2}(\Pi_t - 1)^2}{1-\frac{\kappa}{2}(\Pi_{t+1}-1)^2} - \kappa\left(\Pi_t - 1\right)\Pi_t \\ \tilde{X}_t - \tilde{C}_t \end{bmatrix}$$

6.3.1 Solving the Model

Similar to the previous examples, the following subsections show how to solve the model: find the steady-state, define the matrices of first partial derivatives of f with respect to all its variables evaluated at the steady-state, and find the second-order approximation to the policy functions. As in the basic New-Keynesian model, we consider two parameter configurations to show how to use reduced Gröbner bases to find all the approximations and how to use the MSS criterion to determine how many of them are stable.

Steady-State Calculating the steady-state involves setting $\chi = 0$ and $\varepsilon_t = 0$. Therefore, we have $\Pi_t = \Pi_{t+1} = \Pi_{ss}$, $\tilde{X}_t = \tilde{X}_{t+1} = \tilde{X}_{ss}$, $\tilde{C}_t = \tilde{C}_{t-1} = \tilde{C}_{ss}$, and $\mu_{t+1} = \mu_t = \bar{\mu}$. The equilibrium conditions in the steady-state are

$$\begin{bmatrix} \frac{1}{\tilde{C}_{ss} - \varphi \exp(-\bar{\mu})\tilde{C}_{ss}} - \beta \frac{\varphi}{\tilde{X}_{ss}\exp(\bar{\mu}) - \varphi\tilde{C}_{ss}} - \tilde{\lambda}_{ss} \\ \beta \frac{\tilde{\lambda}_{ss}}{\exp(\bar{\mu})} \frac{R_{ss}\Pi_{ss}^{\psi_t}}{\Pi_{ss}} - \tilde{\lambda}_{ss} \\ (1-\eta) + \frac{\eta}{\tilde{\lambda}_{ss}} + \beta\kappa\left(\Pi_{ss} - 1\right)\Pi_{ss}\frac{\tilde{\lambda}_{ss}}{\tilde{\lambda}_{ss}}\frac{\tilde{X}_{ss}}{\tilde{C}_{ss}}\frac{1-\frac{\kappa}{2}(\Pi_{ss} - 1)^2}{1-\frac{\kappa}{2}(\Pi_{ss}-1)^2} - \kappa\left(\Pi_{ss} - 1\right)\Pi_{ss} \\ \tilde{X}_{ss} - \tilde{C}_{ss} \end{bmatrix} = 0_{4\times 1}.$$

Assuming $\Pi_{ss} = 1$, the steady-state satisfies

$$R_{ss} = \frac{\exp\left(\bar{\mu}\right)}{\beta},$$

$$\tilde{\lambda}_{ss} = \frac{\eta}{\eta - 1},$$

$$\text{and } \tilde{C}_{ss} = \tilde{X}_{ss} = \frac{\exp\left(\bar{\mu}\right) - \beta\varphi}{\exp\left(\bar{\mu}\right) - \varphi}\frac{\eta - 1}{\eta}.$$

Partial Derivatives of f In this example, $n_y = 3$, $n_x = 1$, $n_\varepsilon = 1$, and $n_\theta = 2$. Thus, the first partial derivatives of f with respect to all its variables evaluated at the steady-state are

$$\mathcal{D}_{1,3} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 & \frac{\varphi\beta\exp(\bar{\mu})}{\tilde{C}_{ss}^2(\exp(\bar{\mu})-\varphi)^2} & 0 \\ -\lambda_{ss} & 0 & 1 \\ \beta\kappa & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathcal{D}_{4,6} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 & 0 & -1 \\ \lambda_{ss}\psi\left(s_t\right) & 0 & -1 \\ -\kappa & 0 & -\frac{\eta}{\tilde{\lambda}_{ss}^2} \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathcal{D}_{7,7} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} -\frac{\exp(2\bar{\mu})+\beta\varphi^2}{\tilde{C}_{ss}^2(\exp(\bar{\mu})-\varphi)^2} \\ 0 \\ 0 \\ -1 \end{bmatrix}, \mathcal{D}_{8,8} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} \frac{\exp(\bar{\mu})\varphi}{\tilde{C}_{ss}^2(\exp(\bar{\mu})-\varphi)^2} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathcal{D}_{9,9} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{D}_{10,10} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 \\ \lambda_{ss}\sigma \\ 0 \\ 0 \end{bmatrix}, \mathcal{D}_{11,12} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} \frac{\exp(\bar{\mu})\beta\varphi}{\tilde{C}_{ss}(\exp(\bar{\mu})-\varphi)^2} & 0 \\ -\lambda_{ss} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\text{and } \mathcal{D}_{13,14} f_{ss} = \begin{bmatrix} -\frac{\exp(\bar{\mu})\varphi}{\tilde{C}_{ss}(\exp(\bar{\mu})-\varphi)^2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

for all s_{t+1} and s_t.

Unique Stable Approximation Consider the first parameter configuration

β	κ	η	φ	σ	$\bar{\mu}$	$\hat{\mu}(1)$	$\hat{\mu}(2)$	$\widehat{\psi}(1)$	$\widehat{\psi}(2)$	$p_{1,1}$	$p_{2,2}$
0.9976	161	10	0.7	0.0025	0.005	0.0025	−0.0025	3.1	0.9	0.90	0.90

All parameters other than φ (the degree of habit formation) are similar to those in the previous New-Keynesian example. The steady-state values are $\tilde{X}_{ss} = \tilde{C}_{ss} = 0.904957$ and $\tilde{\lambda}_{ss} = 1.11111$.

Consequently the numerical values of the matrices of first partial derivatives of f with respect to all its variables evaluated at the steady-state are

$$\mathcal{D}_{1,3}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0 & 9.21159 & 0\\ -1.1111 & 0 & 1\\ 160.614 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},\ \mathcal{D}_{4,6}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0 & 0 & -1\\ 1.11111\psi\left(s_t\right) & 0 & -1\\ -161. & 0 & -8.1\\ 0 & 1 & 0\end{bmatrix},$$

$$\mathcal{D}_{7,7}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}-19.6731\\ 0\\ 0\\ -1\end{bmatrix},\ \mathcal{D}_{8,8}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}9.23375\\ 0\\ 0\\ 0\end{bmatrix},\mathcal{D}_{9,9}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0\\ 0\\ 0\\ 0\end{bmatrix},$$

$$\mathcal{D}_{10,10}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}0\\ 0.00277778\\ 0\\ 0\end{bmatrix},\mathcal{D}_{11,12}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}8.33609 & 0\\ -1.11111 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix},$$

$$\text{and } \mathcal{D}_{13,14}f_{ss}\left(s_{t+1},s_t\right)=\begin{bmatrix}-8.35615 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}$$

for all s_{t+1} and s_t.

Using these matrices, we can solve the quadratic system for $\left\{\mathcal{D}_{1,1}g_{ss}\left(s_t\right),\mathcal{D}_{1,1}h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$.

In this example there are a total of 16 solutions. To conserve space, we report $D_{1,1}h_{ss}(s_t)$ only:

	$D_{1,1}h_{ss}(1)$	$D_{1,1}h_{ss}(2)$
(1)	0.69651	0.69651
(2)	1.43919	1.43919
(3)	X0.79309	1.5799
(4)	1.5799	0.79309
(5)	$0.7613 - 0.0895i$	$1.1350 - 0.1129i$
(6)	$0.7613 + 0.0895i$	$1.1350 + 0.1129i$
(7)	$1.0207 - 0.5488i$	$1.0926 - 0.0218i$
(8)	$1.0207 + 0.5488i$	$1.0926 + 0.0218i$
(9)	$1.1188 - 0.4326i$	$1.0051 + 0.0997i$
(10)	$1.1188 + 0.4326i$	$1.0051 - 0.0997i$
(11)	$1.5474 - 0.04355i$	$1.1735 - 0.1404i$
(12)	$1.5474 + 0.04355i$	$1.1735 + 0.1404i$
(13)	$1.1021 + 0.3496i$	$0.8230 + 0.0927i$
(14)	$1.1021 - 0.3496i$	$0.8230 - 0.0927i$
(15)	$1.1786 + 0.3971i$	$1.6102 - 0.0786i$
(16)	$1.1786 - 0.3971i$	$1.6102 + 0.0786i$

Mathematica finds 16 solutions in approximately 5 seconds. The only solution that produces a stable approximation is (1). Given Solution (1), we calculate the rest of the matrices needed to obtain the second-order approximation through solving systems of linear equations. Let $\widehat{\tilde{C}}_t = \tilde{C}_t - \tilde{C}_{ss}$, $\hat{\Pi}_t = \Pi_t - \Pi_{ss}$, $\widehat{\tilde{X}}_t = \tilde{X}_t - \tilde{X}_{ss}$, $\widehat{\tilde{\lambda}}_t = \tilde{\lambda}_t - \lambda_{ss}$, and define $S_t = \begin{bmatrix} \widehat{\tilde{C}}_{t-1} & \varepsilon_t & \chi \end{bmatrix}^{\intercal}$. The

second-order approximation is

$$\begin{bmatrix} \widehat{\tilde{C}}_t \\ \hat{\Pi}_t \\ \widehat{\tilde{X}}_t \\ \widehat{\tilde{\lambda}}_t \end{bmatrix} = \begin{bmatrix} 0.69651 & -0.0002 & 0.00045 \\ 0. & -0.0001 & 0.00021 \\ 0.69651 & -0.0002 & 0.00045 \\ 0. & 0.00239 & -0.0069 \end{bmatrix} \mathcal{S}_t$$

$$+ \begin{bmatrix} 0 & 0 & -0.0020 & 0 & -2\times10^{-7} & -1\times10^{-6} & -0.0020 & -1\times10^{-6} & 5\times10^{-5} \\ 0 & 0 & 0.00020 & 0 & 2\times10^{-7} & -4\times10^{-7} & 0.00020 & -4\times10^{-7} & -6\times10^{-5} \\ 0 & 0 & -0.0020 & 0 & -2\times10^{-7} & -1\times10^{-6} & -0.0020 & -1\times10^{-6} & 5\times10^{-5} \\ 0 & 0 & 0.00143 & 0 & 6\times10^{-6} & -1\times10^{-5} & 0.00143 & -1\times10^{-5} & -0.0003 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)$$

for $s_t = 1$ and

$$\begin{bmatrix} \widehat{\tilde{C}}_t \\ \hat{\Pi}_t \\ \widehat{\tilde{X}}_t \\ \widehat{\tilde{\lambda}}_t \end{bmatrix} = \begin{bmatrix} 0.69651 & -0.0002 & -0.0005 \\ 0 & -0.0001 & -0.0033 \\ 0.69651 & -0.0002 & -0.0005 \\ 0 & 0.00261 & 0.00707 \end{bmatrix} \mathcal{S}_t$$

$$+ \begin{bmatrix} 0 & 0 & 0.00151 & 0 & -2\times10^{-7} & 1\times10^{-6} & 0.00151 & 1\times10^{-6} & -6\times10^{-5} \\ 0 & 0 & 0.00165 & 0 & 3\times10^{-7} & 6\times10^{-6} & 0.00165 & 6\times10^{-6} & -0.0002 \\ 0 & 0 & 0.00151 & 0 & -2\times10^{-7} & 1\times10^{-6} & 0.00151 & 1\times10^{-6} & -6\times10^{-5} \\ 0 & 0 & 0.00156 & 0 & 7\times10^{-6} & 2\times10^{-5} & 0.00156 & 2\times10^{-5} & 0.00036 \end{bmatrix} (\mathcal{S}_t \otimes \mathcal{S}_t)\,.$$

for $s_t = 2$. Again, the first-order approximation can be easily obtained from the above expressions.

Accuracy: Euler Equation Errors As in the previous examples, we compare the accuracies of the first-order and second-order approximations. The Euler equation error at point

$\left(\tilde{C}_{t-1}, \varepsilon_t; s_t\right)$ for an approximation *order* $\in$ {first,second} is

$$EE^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, 1; s_t\right) = 1 - \beta \int \sum_{s'=1}^{2} p_{s_t,s'} \times$$
$$\left(\begin{array}{c} \beta \frac{\tilde{\lambda}^{order}\left(\tilde{C}^{order}\left(\tilde{C}_{t-1},\varepsilon_t,1;s_t\right),\chi\varepsilon_{t+1},1;s_{t+1}\right)}{\Pi^{order}\left(\tilde{C}^{order}\left(\tilde{C}_{t-1},\varepsilon_t,1;s_t\right),\chi\varepsilon_{t+1},1;s_{t+1}\right)} \times \\ \frac{R_{ss}\Pi^{order}\left(\tilde{C}_{t-1},\varepsilon_t,1;s_t\right)^{\psi_t} \exp(\sigma\varepsilon_t)}{\exp(\mu(s'))\tilde{\lambda}^{order}\left(\tilde{C}_{t-1},\varepsilon_t,1;s_t\right)} \end{array} \right) \mu\left(\varepsilon'\right) d\varepsilon'$$

where $\tilde{C}^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, \chi; s_t\right)$, $\Pi^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, \chi; s_t\right)$, and $\tilde{\lambda}^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, \chi; s_t\right)$ are the approximations to the policy functions. The unconditional absolute Euler equation error is

$$EE^{order} = \int \left| EE^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, \chi; s_t\right) \right| \mu^{order}\left(\tilde{C}_{t-1}, \varepsilon_t, \chi; s_t\right) d\tilde{C}_{t-1} d\varepsilon_t ds_t$$

where μ^{order} is the unconditional distribution of the variables implied by the approximated solution.

Numerical approximation of this integral simulates a $10,000$-period path, with the first $1,000$ periods discarded as a burn-in. For each state $\left(\tilde{C}_{t-1}, \varepsilon_t, s_t\right)$ along the simulated path, we draw 10,000 normally distributed ε_{t+1}'s to compute the expectation over ε_{t+1}, and use the transition values $p_{s_t,s_{t+1}}$ to compute the expectation over s_{t+1}. The entire process takes approximately an hour.

The table below reports the base-10 logarithms of absolute Euler equation errors for the first-order and the second-order approximations. Both the first-order and the second-order approximations produce a high degree of accuracy: a value of -3 implies an error of \$1 for each \$1,000.

Unconditional Absolute Euler Equation Errors ($\log_{10}$)

$EE^{\textbf{first}}$	-2.9261
$EE^{\textbf{second}}$	-2.9527

Multiple Stable Approximations As an alternative parameterization, consider the same parameters as above except that $\widehat{\psi}(2) = 0.6$. The second regime now has a slightly lower response by the monetary authority to inflation. The numerical values of the matrices of

partial derivatives of f with respect to all its variables evaluated at steady-state are unchanged except

$$\mathcal{D}_{4,6} f_{ss}\left(s_{t+1}, s_t\right) = \begin{bmatrix} 0 & 0 & -1 \\ 1.11111\psi\left(s_t\right) & 0 & -1 \\ -161. & 0 & -8.1 \\ 0 & 1 & 0 \end{bmatrix}$$

for all s_{t+1} and s_t, which depends on $\widehat{\psi}\left(s_t\right)$. Using this new matrix as well as the other matrices, we solve the quadratic system for $\left\{\mathcal{D}_{1,1} g_{ss}\left(s_t\right), \mathcal{D}_{1,1} h_{ss}\left(s_t\right)\right\}_{s_t=1}^{n_s}$. There are 16 solutions to the quadratic system, and to conserve space we present $\mathcal{D}_{1,1} h_{ss}\left(s_t\right)$ only:

	$D_{1,1} h_{ss}(1)$	$D_{1,1} h_{ss}(2)$
(1)	0.69651	0.69651
(2)	1.43919	1.43919
(3)	0.79309	1.57990
(4)	1.57990	0.79309
(5)	0.65550	1.03904
(6)	1.67928	1.10504
(7)	$0.8703 - 0.1497i$	$1.1985 + 0.0323i$
(8)	$0.8703 + 0.1497i$	$1.1985 - 0.0323i$
(9)	$1.1236 - 0.4088i$	$0.9377 + 0.1268i$
(10)	$1.1236 + 0.4088i$	$0.9377 - 0.1268i$
(11)	$1.4751 - 0.1292i$	$1.2161 - 0.0448i$
(12)	$1.4751 + 0.1292i$	$1.2161 + 0.0448i$
(13)	$1.1062 + 0.3393i$	$0.7984 + 0.1078i$
(14)	$1.1062 - 0.3393i$	$0.7984 - 0.1078i$
(15)	$1.1817 + 0.3950i$	$1.6177 - 0.0855i$
(16)	$1.1817 - 0.3950i$	$1.6177 + 0.0855i$

There are two solutions – (1) and (5) – that produce stable approximations. The examples in the previous New-Keynesian model and the current variant with habit might suggest that the

only parameter affecting the uniqueness of a stable approximation is $\hat{\psi}(s)$, which governs the monetary authority's response to inflation. This conclusion is incorrect.

Consider a higher degree of habit persistence, $\varphi = 0.9$, while keeping $\hat{\psi}(1) = 3.1$ and $\hat{\psi}(2) = 0.6$. Given these parameter values, the steady-state is $\tilde{C}_{ss} = \tilde{X}_{ss} = 0.918512$. There are 16 solutions to the quadratic system. The following table presents $\mathcal{D}_{1,1}h_{ss}(s_t)$ only:

	$D_{1,1}h_{ss}(1)$	$D_{1,1}h_{ss}(2)$
(1)	0.89551	0.895511
(2)	1.11937	1.11937
(3)	0.82810	1.05334
(4)	1.47489	1.16828
(5)	$1.1194 - 0.0017i$	$1.1194 + 0.0017i$
(6)	$1.1194 + 0.0017i$	$1.1194 - 0.0017i$
(7)	$1.1104 + 0.3027i$	$0.9173 + 0.1984i$
(8)	$1.1104 - 0.3027i$	$0.9173 - 0.1984i$
(9)	$1.1445 - 0.3980i$	$0.9790 + 0.1850i$
(10)	$1.1445 + 0.3980i$	$0.9790 - 0.1850i$
(11)	$1.0486 - 0.0839i$	$1.1523 + 0.0536i$
(12)	$1.0486 + 0.0839i$	$1.1523 - 0.0536i$
(13)	$1.1767 + 0.0742i$	$1.1435 - 0.0715i$
(14)	$1.1767 - 0.0742i$	$1.1435 + 0.0715i$
(15)	$1.1584 + 0.4344i$	$1.4032 - 0.1185i$
(16)	$1.1584 - 0.4344i$	$1.4032 + 0.1185i$

There is only one solution, which is (1), that produces stable approximations. Remember that there are two stable approximations when we set $\varphi = 0.7$ as in the previous example.

7 Conclusion

Markov switching has been introduced as an essential ingredient to a large class of models usable for analyses of structural breaks and regime changes in policy, ranging from backward-looking models (Hamilton (1989) and Sims and Zha (2006)) to forward-looking rational expectations models (Clarida et al. (2000), Lubik and Schorfheide (2004), Davig and Leeper (2007), Farmer et al. (2011)). This paper expands this literature by developing a general methodology for constructing first-order and second-order approximations to the solutions of MSDSGE models. We resolve conceptual issues related to the steady-state and certainty equivalence of first-order approximations; we reduce the potentially very complicated problem to a task of solving a system of quadratic equations; we propose using Gröbner bases to solve such a system; and we apply the MSS criterion to verify the existence and uniqueness of a stable approximation to the solution.

The contribution of this paper is not only theoretical but also practical. We show that after the quadratic system is successfully dealt with, obtaining up to second-order approximations is straightforward, as the remaining task involves finding a solution to only a system of linear equations. Our application to three MSDSGE models illustrates the practical value of our methodology. It is our hope that the advance made in this paper enables applied researchers to estimate MSDSGE models by focusing on improving the efficiency of our methods.

References

Aruoba, S., J. Fernandez-Villaverde, and J. Rubio-Ramirez (2006). Comparing Solution Methods for Dynamic Equilibrium Economies. *Journal of Economic Dynamics and Control 30*(12), 2477–2508.

Becker, E., M. Marianari, T. Mora, and C. Treverso (1993). The Shape of the Shape Lemma. In *International Symposium on Symbolic and Algebraic Computation*, pp. 129–133. ACM Press.

Becker, T., V. Weispfenning, and H. Kredel (1998). *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer-Verlag.

Bi, H. and N. Traum (2012). Estimating Sovereign Default Risk. *European Economic Review 102*(3), 161–66.

Bianchi, F. (2010). Regime Switches, Agents' Beliefs, and Post-World War II US Macroeconomic Dynamics. Working Paper 12-04, Duke University.

Blake, A. and F. Zampolli (2006). Optimal Monetary Policy in Markov-Switching Models with Rational Expectations Agents. Working Paper 298, Bank of England.

Buchberger, B. (1998). An Algorithmic Criterion for the Solvability of Algebraic Systems of Equations. *Gröbner Bases and Applications 251*, 535–545.

Cho, S. (2011). Characterizing Markov-Switching Rational Expectation Models. Working Paper.

Christiano, L., M. Eichenbaum, and C. Evans (2005). Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy. *Journal of Political Economy 113*(1), 1–45.

Clarida, R., J. Gali, and M. Gertler (2000). Monetary Policy Rules and Macroeconomic Stability: Evidence and Some Theory. *Quarterly Journal of Economics 115*(1), 147–180.

Costa, O., M. Fragoso, and R. Marques (2005). *Discrete-Time Markov Jump Linear Systems*. Springer.

Cox, D., J. Little, and D. O'Shea (1997). *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra* (2 ed.). Undergraduate Texts in Mathematics, Springer-Verlag.

Datta, R. (2010). Finding all Nash Equilibria of a Finite Game using Polynomial Algebra. *Economic Theory 42*(1), 55–96.

Davig, T. and T. Doh (2008). Monetary Policy Regime Shifts and Inflation Persistence. Research Working Paper 08-16, Federal Reserve Bank of Kansas City.

Davig, T. and E. Leeper (2007). Generalizing the Taylor Principle. *American Economic Review 97*(3), 607–635.

Davig, T., E. M. Leeper, and T. B. Walker (2010). Unfunded Liabilities and Uncertain Fiscal Financing. *Journal of Monetary Economics 57*(5), 600–619.

Davig, T., E. M. Leeper, and T. B. Walker (2011). Inflation and the Fiscal Limit. *European Economic Review 55*(1), 31–47.

Farmer, R., D. Waggoner, and T. Zha (2009). Understanding Markov-Switching Rational Expectations Models. *Journal of Economic Theory 144*(5), 1849–1867.

Farmer, R., D. Waggoner, and T. Zha (2011). Minimal State Variable Solutions to Markov-Switching Rational Expectations Models. *Journal of Economic Dynamics and Control 35*(12), 2150–2166.

Fernandez-Villaverde, J., P. Guerron-Quintana, and J. Rubio-Ramirez (2009). Fortune or Virtue: Time-Variant Volatilites versus Parameter Drifting in U.S. Data. Working Paper.

Fernández-Villaverde, J. and J. Rubio-Ramirez (2007). Estimating Macroeconomic Models: A Likelihood Approach. *Review of Economic Studies 74*(4), 1059–1087.

Hamilton, J. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica 57*(2), 357–384.

Judd, K. (1998). *Numerical Methods in Economics*. MIT Press.

Kubler, F. and K. Schmedders (2010a). Competitive Equilibria in Semi-Algebraic Economies. *Journal of Economic Theory 145*(1), 301–330.

Kubler, F. and K. Schmedders (2010b). Tackling Multiplicity of Equilibria with Grbner Bases. *Operations Research 58*(4), 1037–1050.

Leeper, E. and T. Zha (2003). Modest Policy Interventions. *Journal of Monetary Economics 50*(8), 1673–1700.

Liu, Z., D. F. Waggoner, and T. Zha (2011). Sources of Macroeconomic Fluctuations: A Regime-Switching DSGE Approach. *Quantitative Economics 2*(2), 251–301.

Lubik, T. and F. Schorfheide (2004). Testing for Indeterminacy: An Application to US Monetary Policy. *American Economic Review 94*(1), 190–217.

Maih, J. (2013). Rationality in a Switching Environment. mimeo.

Rudebusch, G. and E. Swanson (2008). Examining the Bond Premium Puzzle with a DSGE Model. *Journal of Monetary Economics 55*(S1), S111–S126.

Schmitt-Grohe, S. and M. Uribe (2004). Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function. *Journal of Economic Dynamics and Control 28*(4), 755–775.

Schorfheide, F. (2005). Learning and Monetary Policy Shifts. *Review of Economic Dynamics 8*, 392–419.

Sims, C. and T. Zha (2006). Were There Regime Switches in US Monetary Policy? *American Economic Review 96*(1), 54–81.

Svensson, L. and N. Williams (2007). Monetary Policy with Model Uncertainty: Distribution Forecast Targeting. Discussion Paper 6331, CEPR.

van Binsbergen, J., J. Fernandez-Villaverde, R. Koijen, and J. Rubio-Ramirez (2008). Likelihood Estimation of DSGE Models with Epstein-Zin Preferences. Working Paper.

Woodford, M. (2003). *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton University Press.

8 Appendix A:

This appendix derives, in detail, the two steps described in Section 3.2.

8.1 Step 1: Obtaining the derivatives of x_{t-1}

Taking first partial derivatives of $\mathbb{G}$ with respect to x_{t-1} in (10) produces the expression for $\mathcal{D}_{1,n_x}\mathbb{G}_{ss}(s_t)$

$$\mathcal{D}_{1,n_x}\mathbb{G}_{ss}(s_t) = \sum_{s'=1}^{n_s} p_{s_t,s'} \int \begin{pmatrix} \mathcal{D}_{1,n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s')\, \mathcal{D}_{1,n_x} h_{ss}(s_t) \\ +\mathcal{D}_{n_y+1,2n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s_t) \\ +\mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} h_{ss}(s_t) \\ +\mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)} f_{ss}(s', s_t) \end{pmatrix} \mu(\varepsilon')\, d\varepsilon'$$

for all s_t. Taking $\int \mu(\varepsilon')\, d\varepsilon' = 1$ into account, one can simplify the above expression to

$$\mathcal{D}_{1,n_x}\mathbb{G}_{ss}(s_t) = \sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \mathcal{D}_{1,n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s')\, \mathcal{D}_{1,n_x} h_{ss}(s_t) \\ +\mathcal{D}_{n_y+1,2n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s_t) \\ +\mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} h_{ss}(s_t) \\ +\mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)} f_{ss}(s', s_t) \end{pmatrix}$$

for all s_t. Rearranging the above expression for each s_t leads to

$$\mathcal{D}_{1,n_x}\mathbb{G}_{ss}(s_t) = \tag{20}$$
$$\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \left(\mathcal{D}_{1,n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s') + \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s', s_t)\right) \mathcal{D}_{1,n_x} h_{ss}(s_t) \\ +\mathcal{D}_{n_y+1,2n_y} f_{ss}(s', s_t)\, \mathcal{D}_{1,n_x} g_{ss}(s_t) + \mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)} f_{ss}(s', s_t) \end{pmatrix}.$$

Putting together the n_s versions of (20), one for each value of s_t, and equating them to zero as implied by (11) yields a quadratic system of $(n_y + n_x)\, n_x n_s$ equations. The number of equations is exactly the same as the number of unknowns $\{\mathcal{D}_{1,n_x} g_{ss}(s_t), \mathcal{D}_{1,n_x} h_{ss}(s_t)\}_{s_t=1}^{n_s}$. Writing (20) in matrix form leads to a system of quadratic equations expressed in (12).

8.2 Step 2: Obtaining the derivatives of ε_t and χ

8.2.1 Obtaining the derivatives of ε_t

By taking first partial derivatives with respect to ε_t in (10) we obtain the expression for $\mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}(s_t)$

$$\mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}(s_t) =$$
$$\sum_{s'=1}^{n_s} p_{s_t,s'} \int \begin{pmatrix} \mathcal{D}_{1,n_y} f_{ss}(s',s_t)\,\mathcal{D}_{1,n_x} g_{ss}(s')\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t) + \\ \mathcal{D}_{n_y+1,2n_y} f_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t) + \\ \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t) + \\ \mathcal{D}_{2(n_y+n_x)+n_\varepsilon+1,2(n_y+n_x+n_\varepsilon)} f_{ss}(s',s_t) \end{pmatrix} \mu(\varepsilon')\,d\varepsilon'$$

for all s_t. With $\int \mu(\varepsilon')\,d\varepsilon' = 1$, this expression simplifies to

$$\mathcal{D}_{n_x+1,n_x+n_\varepsilon}\mathbb{G}_{ss}(s_t) =$$
$$\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{pmatrix} \left(\mathcal{D}_{1,n_y} f_{ss}(s',s_t)\,\mathcal{D}_{1,n_x} g_{ss}(s') + \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s',s_t)\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t) \\ \mathcal{D}_{n_y+1,2n_y} f_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t) \\ \mathcal{D}_{2(n_y+n_x)+n_\varepsilon+1,2(n_y+n_x+n_\varepsilon)} f_{ss}(s',s_t) \end{pmatrix} \tag{21}$$

for all s_t.

Putting together the n_s versions of (21), one for each value of s_t, and equating them to zero as implied by (11) yields a system of $(n_y+n_x)\,n_\varepsilon n_s$ equations. The number of equations is the same as that of unknowns $\{\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s_t), \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(s_t)\}_{s_t=1}^{n_s}$. The system is linear and can be written in matrix form as

$$\begin{bmatrix} \Theta_\varepsilon & \Phi_\varepsilon \end{bmatrix} \begin{bmatrix} \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(n_s) \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}(n_s) \end{bmatrix} = \Psi_\varepsilon. \tag{22}$$

The solution to this system is given in (13).

8.2.2 Obtaining the derivatives of χ

We obtain the expression for $\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t)$ by taking first partial derivatives with respect to χ in (10)

$$\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t) =$$
$$\sum_{s'=1}^{n_s} p_{s_t,s'} \int \left(\begin{array}{c} \mathcal{D}_{1,n_y} f_{ss}(s', s_t) \left[\begin{array}{c} \mathcal{D}_{1,n_x} g_{ss}(s') \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) \\ +\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}(s') \varepsilon' + \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s') \end{array} \right] + \\ \mathcal{D}_{n_y+1,2n_y} f_{ss}(s', s_t) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) + \\ \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s', s_t) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) + \\ \mathcal{D}_{2n_y+n_x+1,2n_y+n_x+n_\varepsilon} f_{ss}(s', s_t) \varepsilon' + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s') + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s_t) \end{array} \right) \mu(\varepsilon') d\varepsilon'$$

for all s_t. Note that $\mathcal{D}\theta_{ss}(s_t)$ is the derivative of $\theta(\chi, s_t)$ with respect to χ evaluated at $\chi = 0$. That is,

$$\mathcal{D}\theta_{ss}(s_t) = \mathcal{D}\theta(0, s_t) = \left[\mathcal{D}_j \theta^i(0, s_t)\right]_{1 \le i \le n_\theta, j=1}$$

for all s_t.

Using the two equalities $\int \mu(\varepsilon') d\varepsilon' = 1$ and $\int \varepsilon' \mu(\varepsilon') d\varepsilon' = 0$, one can simplify the expression for $\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t)$ as

$$\mathcal{D}_{n_x+n_\varepsilon+1}\mathbb{G}_{ss}(s_t) = \tag{23}$$
$$\sum_{s'=1}^{n_s} p_{s_t,s'} \left(\begin{array}{c} \mathcal{D}_{1,n_y} f_{ss}(s', s_t) \left\{ \mathcal{D}_{1,n_x} g_{ss}(s') \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) + \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s') \right\} + \\ \mathcal{D}_{n_y+1,2n_y} f_{ss}(s', s_t) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t) + \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}(s', s_t) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t) \\ +\mathcal{D}_{2(n_y+n_x+n_\varepsilon)+1,2(n_y+n_x+n_\varepsilon)+n_\theta} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s') + \\ \mathcal{D}_{2(n_y+n_x+n_\varepsilon)+n_\theta+1,2(n_y+n_x+n_\varepsilon+n_\theta)} f_{ss}(s', s_t) \mathcal{D}\theta_{ss}(s_t) \end{array} \right)$$

for all s_t.

Putting together the n_s versions of (23), one for each value of s_t, and equating them to zero as implied by (11), yields a system of linear $(n_y + n_x) n_s$ equations. The number of these equations is the same as that of unknowns $\{\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(s_t), \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(s_t)\}_{s_t=1}^{n_s}$. The linear

system can be written in matrix form as

$$\begin{bmatrix} \Theta_\chi & \Phi_\chi \end{bmatrix} \begin{bmatrix} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}(n_s) \\ \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(1) \\ \vdots \\ \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}(n_s) \end{bmatrix} = \Psi_\chi. \tag{24}$$

The solution to (24) is given in (13).

9 Appendix B: Gröbner bases

In this appendix we give an overview of Gröbner bases and describe how they can be applied to our problem. See Becker et al. (1998) for a more detailed description and other applications.

We wish to find all the solutions of a system of n polynomials in n variables. Let the polynomial system under study be

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, \ldots, x_n) &= 0. \end{aligned}$$

Each equation in this system defines a manifold of dimension $(n-1)$ in $\mathbb{R}^n$ and the set of solutions of the system is the intersection of these manifolds.

When all the f_i's are linear, the solution set consists of a linear subspace of $\mathbb{R}^n$. It is well known that there are three possible outcomes: a unique solution, no solutions, or infinitely many solutions. More importantly, the set of linear systems with no solution or infinitely many solutions is of measure zero in the set of all linear systems. When there is a unique solution, it can be easily found.

When the f_i's are higher-order polynomials, the solution set is more complicated, but the intuition from the linear case still holds. For almost all polynomial systems of n equations in n variables, there are only finitely many solutions and these solutions can be easily found.

To describe how solutions are computed for polynomial systems, we need to develop the concept of an ideal and its Gröbner basis. Given a set of polynomials in n variables, $\{f_1, \ldots, f_m\}$, the *ideal* generated by $\{f_1, \ldots, f_m\}$ is the set of all polynomials of the form

$$g_1(x_1, \ldots, x_n)f_1(x_1, \ldots, x_n) + \cdots + g_m(x_1, \ldots, x_n)f_m(x_1, \ldots, x_n) \tag{25}$$

where $g_1, \ldots, g_m$ vary over all polynomials in n variables. We denote this ideal by $\langle f_1, \ldots, f_m \rangle$. For our purpose we focus on one important feature of an ideal. The point $(a_1, \ldots, a_n) \in \mathbb{R}^n$ is a zero of the polynomials $f_1, \ldots, f_m$ if and only if it is a zero of every polynomial in the ideal $\langle f_1, \ldots, f_m \rangle$. This feature implies that if two different sets of polynomials generate the same ideal, then they have the same zeros. The goal is to find a generating set for which it is easy to compute zeros.

Before giving the definition of a Gröbner basis, we must first define what we mean by the leading term of a polynomial. A polynomial in $x_1, \ldots, x_n$ is a sum of *terms* of the form $cx_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$, where k_i is a non-negative integer and c is a non-zero real number. The product $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ is called a *monomial* in $x_1 \cdots x_n$. The degree of a term is the sum of its exponents, $k_1 + \cdots + k_n$. For polynomials in a single variable, the leading term is defined to be the one of highest degree. For polynomials of several variables, there may be many terms of the same degree. Thus, one defines the leading term relative to a monomial ordering. For instance, the lexicographical ordering of monomials implies that $x_1^{k_1} \cdots x_n^{k_n} < x_1^{m_1} \cdots x_n^{m_n}$ if and only if there is an i such that $k_i < m_i$ and $k_j = m_j$ for $j < i$. In general, a monomial order must satisfy

1. The monomial $x_1^0 x_2^0 \cdots x_n^0 = 1$ is the smallest monomial.

2. If X, Y, and Z are monomials with $X < Y$, then $XZ < YZ$.

The *leading term* of a polynomial is the largest term with respect to the monomial ordering. With these notions in hand, we are ready to define a Gröbner basis. The set $\{h_1, \ldots, h_k\}$ is a *Gröbner basis* for the ideal $\langle f_1, \ldots, f_m \rangle$ if

1. $\langle f_1, \ldots, f_m \rangle = \langle h_1, \ldots, h_k \rangle$

2. The leading term of any polynomial in $\langle f_1, \ldots, f_m \rangle$ is divisible by the leading term of h_i for some i.

Consider the following example. Consider the ideal generated by

$$\{2x_1x_2 - x_1, x_2 - x_3, x_3^2 - 1\}.$$

Note that the first term of each polynomial is the leading term with respect to the lexicographical order. Is this generating set a Gröbner basis? The answer is negative because the leading term in $\frac{1}{2}(2x_1x_2 - x_1) - x_1(x_2 - x_3) = x_1x_3 - \frac{1}{2}x_1$ is not divisible by any of the leading terms in the generating set.

As an illustration, we show how to use Buchberger's Algorithm to construct a Gröbner basis. There are many other algorithms, most of which are based on Buchberger's Algorithm, that can also be used to construct Gröbner bases. The algorithm begins with constructing S-polynomials. The polynomial $\frac{1}{2}(2x_1x_2 - x_1) - x_1(x_2 - x_3) = x_1x_3 - \frac{1}{2}x_1$ is called the S-polynomial of $2x_1x_2 - x_1$ and $x_2 - x_3$ because factors $\frac{1}{2}$ and x_1 were chosen so that the leading terms would cancel. After the S-polynomial has been formed, it must be *reduced* using the elements of the generating set. The reduction step is illustrated by the following example.

Consider the S-polynomial of $2x_1x_2 - x_1$ and $x_3^2 - 1$, which is $\frac{1}{2}x_3^2(2x_1x_2 - x_1) - x_1x_2(x_3^2 - 1) = x_1x_2 - \frac{1}{2}x_1x_3^2$. This polynomial is reduced by using the leading terms in the generating set to eliminate terms in the S-polynomial. In this case the reduction proceeds as follows:

$$\begin{aligned} x_1x_2 - \frac{1}{2}x_1x_3^2 &\Rightarrow (x_1x_2 - \frac{1}{2}x_1x_3^2) - \frac{1}{2}(2x_1x_2 - x_1) = -\frac{1}{2}x_1x_3^2 + \frac{1}{2}x_1 \\ -\frac{1}{2}x_1x_3^2 + \frac{1}{2}x_1 &\Rightarrow (-\frac{1}{2}x_1x_3^2 + \frac{1}{2}x_1) + \frac{1}{2}x_1(x_3^2 - 1) = 0 \end{aligned}$$

So the reduction of the S-polynomial of $2x_1x_2 - x_1$ and $x_3^2 - 1$ gives the zero polynomial. Readers should convince themselves that the S-polynomial of $2x_1x_2 - x_1$ and $x_2 - x_3$ given above cannot be reduced further and that the S-polynomial of $x_2 - x_3$ and $x_3^2 - 1$ can be reduced to zero. Note that the reduction is in general not unique. It can depend on the order in which the terms are eliminated and on particular elements of the generating set that are used to eliminate the terms. One can always devise an algorithm to reduce any polynomial in finite steps.

Buchberger's Algorithm proceeds as follows. Successively form the S-polynomials from pairs of polynomials in the generating set and reduce them. If a reduced non-zero S-polynomial is obtained, add it to the generating set. Continue until all S-polynomials formed from pairs of the enlarged generating set can be reduced to zero. This algorithm is guaranteed to terminate in a Gröbner basis. See Buchberger (1998) or Becker et al. (1998) for details.

Continuing with our example, the reduced S-polynomial of $2x_1x_2-x_1$ and x_2-x_3 is $x_1x_3-\frac{1}{2}x_1$. We add it to our generating set to obtain

$$\{2x_1x_2 - x_1, x_2 - x_3, x_3^2 - 1, x_1x_3 - \frac{1}{2}x_1\}.$$

As discussed above, the S-polynomials of both the pair $2x_1x_2-x_1$ and x_3^2-1 and the pair x_2-x_3 and x_3^2-1 are zero. Note also that the S-polynomial of $2x_1x_2-x_1$ and $x_1x_3-\frac{1}{2}x_1$ reduces to zero, but the S-polynomial of x_2-x_3 and $x_1x_3-\frac{1}{2}x_1$ is $x_1x_3(x_2-x_3)-x_2(x_1x_3-\frac{1}{2}x_1) = \frac{1}{2}x_1x_2-x_1x_3^2$ and reduces to

$$\begin{aligned}
\frac{1}{2}x_1x_2 - x_1x_3^2 &\Rightarrow (\frac{1}{2}x_1x_2 - x_1x_3^2) - \frac{1}{4}(2x_1x_2 - x_1) = -x_1x_3^2 + \frac{1}{4}x_1 \\
-x_1x_3^2 + \frac{1}{4}x_1 &\Rightarrow (-x_1x_3^2 + \frac{1}{4}x_1) + x_1(x_3^2 - 1) = -\frac{3}{4}x_1.
\end{aligned}$$

We add this non-zero polynomial to our generating set to obtain

$$\{2x_1x_2 - x_1, x_2 - x_3, x_3^2 - 1, 2x_1x_3 - x_1, -\frac{3}{4}x_1\}.$$

The reader should verify that all S-polynomials of pairs from this generating set will reduce to zero. Thus we have obtained a Gröbner basis.

Gröbner bases are not unique, because adding any element from the ideal generated by a Gröbner basis will result in another Gröbner basis. To obtain uniqueness, with respect to the monomial ordering, we work with a reduced Gröbner basis. A Gröbner basis is said to be *reduced* if

1. The coefficient of the leading term of each polynomial in the basis is one.

2. Each polynomial in the basis cannot be further reduced with respect to the other polynomials in the basis.

Any Gröbner basis can be easily transformed to a reduced Gröbner basis by first reducing each polynomial in the basis with respect to the other polynomials in the basis and then dividing the resultant leading term by the leading coefficient. For instance, the Gröbner basis obtained above is not reduced because both $2x_1x_2 - x_1$ and $2x_1x_3 - x_1$ can be reduced to zero. Thus these polynomials must be eliminated to obtain the reduced Gröbner basis

$$\{x_1, x_2 - x_3, x_3^2 - 1\}.$$

The reduced basis above is called a *Shape basis* because it is of the form

$$\{x_1 - q_1(x_n), \ldots, x_{n-1} - q_{n-1}(x_n), q_n(x_n)\},$$

where $q_1, \ldots, q_n$ are polynomials in a single variable with the degree of q_i strictly less than the degree of q_n for $1 \leq i \leq n-1$. Shape bases are particularly useful because it is straightforward to find all the zeros from this representation. One first finds the values of x_n that are zeros of $q_n(x_n)$ and then substitutes each of these values into q_1 through q_{n-1} to obtain the values of x_1 through x_{n-1}.

Not all reduced Gröbner bases are Shape bases. The Shape lemma, below, gives the conditions under which the reduced Gröbner basis is a Shape basis.

Lemma 5 *Let $f_1, \ldots, f_n$ be polynomials in $x_1, \ldots, x_n$. The reduced Gröbner basis with respect to the lexicographical ordering of the ideal $\langle f_1, \ldots, f_n \rangle$ is a Shape basis if and only if the following conditions hold.*

1. *The system $f_1, \ldots, f_n$ has only finitely many zeros.*
2. *If $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ are two distinct zeros, then $a_n \neq b_n$.*
3. *Each zero is either a simple point or a multiple point of local dimension one.*
4. *If a zero is a multiple point, then the tangent line at the zero does not contain the hyperplane $x_n = 0$.*

The meaning of Conditions 1 and 2 is clear, but Conditions 3 and 4 need further explanation. The point $(a_1, \ldots, a_n) \in \mathbb{R}^n$ is a zero of the polynomial system $f_1, \ldots, f_n$ if and only if $\langle f_1, \ldots, f_n \rangle \subseteq \langle x_1 - a_1, \ldots, x_n - a_n \rangle$. If there exists an i such that

$$\langle f_1, \ldots, f_n \rangle \subseteq \langle x_1 - a_1, \ldots, (x_i - a_i)^2, \ldots, x_n - a_n \rangle \subset \langle x_1 - a_1, \ldots, x_n - a_n \rangle,$$

then we say the zero is a *multiple point*; otherwise, the zero is *simple*. One can verify that the zero $(a_1, \ldots, a_n)$ is a multiple point if and only if there exists an i such that $\partial f_j / \partial x_i|_{(a_1,\ldots,a_n)} = 0$ for all $1 \leq j \leq n$. The *tangent space* at the zero $(a_1, \ldots, a_n)$ is the set of all $(x_1, \ldots, x_n) \in \mathbb{R}^n$ such that

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1}|_{(a_1,\ldots,a_n)} & \cdots & \frac{\partial f_1}{\partial x_n}|_{(a_1,\ldots,a_n)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}|_{(a_1,\ldots,a_n)} & \cdots & \frac{\partial f_n}{\partial x_n}|_{(a_1,\ldots,a_n)} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = 0.$$

Note that this matrix of partial derivatives is the Jacobian. If the zero is simple, then this definition of the tangent space corresponds to our usual geometric notion of a tangent space, but the correspondence breaks down if the zero is a multiple point. The *local dimension* of a zero is the dimension of the tangent space. Note that the local dimension is zero if and only if the Jacobian is of full rank. Thus, if the Jacobian is of full rank, then the zero will be simple. The converse, however, is not necessarily true.

One can verify that if the reduced Gröbner basis is a Shape basis, then Conditions 1-4 will hold. The converse is also true, but the verification requires much more work. See Becker et al. (1993) for details. If the Jacobian at each zero is of full rank, then each zero is isolated and there can only be finitely many zeros. Thus, Conditions 1-4 hold in this case. Since the set of polynomial systems of n equations in n unknowns whose Jacobian is not of full rank is of measure zero in the set of all such systems, Conditions 1-4 hold almost surely.

In summary, for almost all polynomial systems, there are only finitely many zeros and Buchberger's Algorithm can be used to find them. While Buchberger's Algorithm is instructive, it can be inefficient for certain problems. Active research continues to develop variants of this algorithm that improve efficiency.

10 Appendix C: Iterative algorithm

This section describes an iterative procedure to find solutions of the quadratic system (12). In general, this method will not provide us with all the solutions to the system. Recall the quadratic system to be solved is

$$A\left(s_t\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x} g_{ss}\left(1\right) \\ \vdots \\ \mathcal{D}_{1,n_x} g_{ss}\left(n_s\right) \end{bmatrix} \mathcal{D}_{1,n_x} h_{ss}\left(s_t\right) = B\left(s_t\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x} g_{ss}\left(s_t\right) \end{bmatrix}$$

for all s_t, where

$$A\left(s_t\right) = \begin{bmatrix} \sum_{s'=1}^{n_s} p_{s_t,s'} \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}\left(s', s_t\right) & p_{s_t,1}\mathcal{D}_{1,n_y} f_{ss}\left(1, s_t\right) & \cdots & p_{s_t,n_s}\mathcal{D}_{1,n_y} f_{ss}\left(n_s, s_t\right) \end{bmatrix}$$

and

$$B\left(s_t\right) = -\sum_{s'=1}^{n_s} p_{s_t,s'} \begin{bmatrix} \mathcal{D}_{2n_y+n_x+1,2(n_y+n_x)} f_{ss}\left(s', s_t\right) & \mathcal{D}_{n_y+1,2n_y} f_{ss}\left(s', s_t\right) \end{bmatrix}$$

for all s_t.

The idea of the algorithm is to guess a set of policy functions and solve for each regime's policy functions as in the constant parameter model case using the generalized Schur decomposition. When the solutions are close to the guesses, then a solution has been found.

Algorithm 6 *Let* $\left\{\mathcal{D}_{1,n_x} g_{ss}^{(j)}\left(s_t\right), \mathcal{D}_{1,n_x} h_{ss}^{(j)}\left(s_t\right)\right\}$ *denote solution at iteration* j.

1. *Set* $j = 1$ *and initialize* $\mathcal{D}_{1,n_x} g_{ss}^{(0)}\left(s_t\right) = 0_{n_y \times n_x}$ *and* $\mathcal{D}_{1,n_x} h_{ss}^{(0)}\left(s_t\right) = 0_{n_x \times n_x}$.

2. *For each* s_t, *construct the transformed system*

$$A\left(s_t, \left\{\mathcal{D}_{1,n_x} g_{ss}^{(j-1)}\left(s'\right)\right\}_{s'=1, s' \neq s_t}^{n_s}\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x} g_{ss}^{(j)}\left(s_t\right) \end{bmatrix} \mathcal{D}_{1,n_x} h_{ss}^{(j)}\left(s_t\right) = B\left(s_t\right)\begin{bmatrix} I_{n_x} \\ \mathcal{D}_{1,n_x} g_{ss}^{(j)}\left(s_t\right) \end{bmatrix} \tag{26}$$

where

$$A\left(s_t, \left\{\mathcal{D}_{1,n_x} g_{ss}^{(j)}\left(s'\right)\right\}_{s'=1,s'\neq s_t}^{n_s}\right) =$$
$$\left[\begin{array}{c} \left(\begin{array}{c} \sum_{s'=1}^{n_s} p_{s_t,s'} \mathcal{D}_{2n_y+1,2n_y+n_x} f_{ss}\left(s', s_t\right) \\ + \sum_{s'=1,s'\neq s_t}^{n_s} p_{s_t,s'} \mathcal{D}_{1,n_y} f_{ss}\left(s', s_t\right) \mathcal{D}_{1,n_x} g_{ss}^{(0)}\left(s'\right) \end{array}\right) \quad p_{s_t,s_t} \mathcal{D}_{1,n_y} f_{ss}\left(s_t, s_t\right) \end{array}\right]$$

3. *The set of systems (26) is identical to a constant parameter model case. When finding* $\mathcal{D}_{1,n_x} h_{ss}^{(j)}\left(s_t\right)$*, use the "most stable" generalized eigenvalues, those with the smallest modulus.*

4. *Check* $\max\left\{\left\|\mathcal{D}_{1,n_x} h_{ss}^{(j)}\left(s_t\right) - \mathcal{D}_{1,n_x} h_{ss}^{(j-1)}\left(s_t\right)\right\|, \left\|\mathcal{D}_{1,n_x} g_{ss}^{(j)}\left(s_t\right) - \mathcal{D}_{1,n_x} g_{ss}^{(j-1)}\left(s_t\right)\right\|\right\} < crit.$ *If yes, then stop and check for MSS. If no, then set* $j \implies j+1$ *and return to step* 2.

11 Appendix D: Second partial derivatives

This appendix derives, in detail, the six steps described in Section 5.2. Let us first define

$$\mathcal{D}_k \mathcal{D}_j f_{ss}^i\left(s_{t+1}, s_t\right) =$$
$$\mathcal{D}_k \mathcal{D}_j f^i\left(y_{ss}, y_{ss}, x_{ss}, x_{ss}, 0_{n_\varepsilon}, 0_{n_\varepsilon}, \overline{\theta}_1, \widehat{\theta}_2\left(s_{t+1}\right), \overline{\theta}_1, \widehat{\theta}_2\left(s_t\right)\right)$$

for all s_{t+1} and s_t and $1 \leq i \leq n_y + n_x$ and $1 \leq j, k \leq 2\left(n_y + n_x + n_\varepsilon + n_\theta\right)$ and

$$\mathcal{H}_{n_1,n_2;m_1,m_2} f_{ss}^i\left(s_{t+1}, s_t\right) = \left[\begin{array}{ccc} \mathcal{D}_{n_1} \mathcal{D}_{m_1} f_{ss}^i\left(s_{t+1}, s_t\right) & \ldots & \mathcal{D}_{n_1} \mathcal{D}_{m_2} f_{ss}^i\left(s_{t+1}, s_t\right) \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{n_2} \mathcal{D}_{m_1} f_{ss}^i\left(s_{t+1}, s_t\right) & \ldots & \mathcal{D}_{n_2} \mathcal{D}_{m_2} f_{ss}^i\left(s_{t+1}, s_t\right) \end{array}\right]$$

for all s_{t+1} and s_t and $1 \leq n_1, m_1 < n_2, m_2 \leq 2\left(n_y + n_x + n_\varepsilon + n_\theta\right)$ and $1 \leq i \leq n_y + n_x$.

11.1 Step 1: Obtaining the derivatives of x_{t-1} twice

The first equation is the derivative with respect to x_{t-1} twice.

$$\mathcal{H}_{1,n_x;1,n_x}\mathbb{G}^i_{ss}\left(s_t\right) = \sum_{s'=1}^{n_s} p_{s_t,s'} \times$$

$$\left(\begin{array}{c}
\left(\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right)^{\intercal}\left(\begin{array}{c}\left(\begin{array}{c}\mathcal{H}_{1,n_y;1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\ +2\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\\ +2\mathcal{H}_{1,n_y;n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right)\\ +2\mathcal{H}_{1,n_y;2n_y+n_x+1,2(n_y+n_x)}f^i_{ss}\left(s',s_t\right)\end{array}\right)\\
+\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right)\\ +2\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\\ +2\mathcal{H}_{n_y+1,2n_y;2n_y+n_x+1,2(n_y+n_x)}f^i_{ss}\left(s',s_t\right)\end{array}\right)\\
+\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}\left(\begin{array}{c}\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\\ +\mathcal{D}_{1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{H}_{1,n_x;1,n_x}g_{ss}\left(s'\right)\end{array}\right)\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\\ +2\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+n_x+1,2(n_y+n_x)}f^i_{ss}\left(s',s_t\right)\end{array}\right)\\
+\left(\begin{array}{c}\mathcal{D}_{1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\ +\mathcal{D}_{2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{H}_{1,n_x;1,n_x}h_{ss}\left(s_t\right)\\
+\mathcal{D}_{n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{H}_{1,n_x;1,n_x}g_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_y+n_x);2n_y+n_x+1,2(n_y+n_x)}f^i_{ss}\left(s',s_t\right)
\end{array}\right)$$

The condition $\mathcal{H}_{1,n_x;1,n_x}\mathbb{G}^i_{ss}\left(s_t\right) = 0_{n_x\times n_x}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution for $\left\{\left\{\mathcal{H}_{1,n_x;1,n_x}g^i_{ss}\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{1,n_x;1,n_x}h^i_{ss}\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

11.2 Step 2: Obtaining the derivatives of x_{t-1} and ε_t

The second equation is the derivative with respect to x_{t-1} and ε_t.

$$\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}\left(s_t\right)=\sum_{s'=1}^{n_s}p_{s_t,s'}\times$$

$$\left(\begin{array}{c}
\left(\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)\right)^{\intercal}\left(\begin{array}{c}\left(\begin{array}{c}\mathcal{H}_{1,n_y;1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\+2\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\\+\mathcal{H}_{1,n_y;n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right)\\+\mathcal{H}_{1,n_y;2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}\left(s',s_t\right)\end{array}\right)\\
+\mathcal{D}_{1,n_x}g_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}\left(\begin{array}{c}\mathcal{H}_{n_y+1,2n_y;1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\+\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\\+\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right)\\+\mathcal{H}_{n_y+1,2n_y;2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}\left(s',s_t\right)\end{array}\right)\\
+\mathcal{D}_{1,n_x}h_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}\mathcal{H}_{2n_y+1,2n_y+n_x;n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right)\\+\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}\left(s',s_t\right)\\+\left(\begin{array}{c}\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\\+\mathcal{D}_{1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{H}_{1,n_x;1,n_x}g_{ss}\left(s'\right)\end{array}\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\end{array}\right)\\
+\left(\begin{array}{c}\mathcal{H}_{2n_y+n_x+1,2(n_x+n_y);1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\+\mathcal{H}_{2n_y+n_x+1,2(n_y+n_x);2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\\
+\left(\begin{array}{c}\mathcal{D}_{1,n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x}g_{ss}\left(s'\right)\\+\mathcal{D}_{2n_y+1,2n_y+n_x}f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}h_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_y+n_x);n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right)\\
+\mathcal{D}_{n_y+1,2n_y}f^i_{ss}\left(s',s_t\right)\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}g_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_x+n_y);2(n_x+n_y)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}\left(s',s_t\right)
\end{array}\right)$$

Conditional on the solution from Step 1, the condition $\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}\left(s_t\right)=0_{n_x\times n_\varepsilon}$ for $1\leq i\leq n_y+n_x$ implies a linear system of equations that determines the solution $\left\{\left\{\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}g^i_{ss}\left(s_t\right)\right\}_{i=1}^{n_y},\left\{\mathcal{H}_{1,n_x;n_x+1,n_x+n_\varepsilon}h^i_{ss}\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

11.3 Step 3: Obtaining the derivatives of x_{t-1} and χ

The third equation is the derivative with respect to x_{t-1} and χ.

$$\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}\left(s_t\right) = \sum_{s'=1}^{n_s} p_{s_t,s'} \times$$

$$\left(\begin{array}{c}
\left(\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{1,n_x} h_{ss}\left(s_t\right)\right)^{\intercal}\left(\begin{array}{c}
\mathcal{H}_{1,n_y;1,n_y} f^i_{ss}\left(s',s_t\right)\left(\begin{array}{c}\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right)\\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\end{array}\right)\\
+\mathcal{H}_{1,n_y;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\\
+\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\\
+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s'\right)\\
+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s_t\right)
\end{array}\right)\\
+\mathcal{D}_{1,n_x} h_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}
\mathcal{H}_{2n_y+1,2n_y+n_x;1,n_y} f^i_{ss}\left(s',s_t\right)\left(\begin{array}{c}\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right)\\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\end{array}\right)\\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+1,2n_y+n_x;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s'\right)\\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s_t\right)\\
\mathcal{D}_{1,n_y} f^i_{ss}\left(s',s_t\right)\left(\begin{array}{c}\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} g_{ss}\left(s'\right)\\ +\mathcal{H}_{1,n_x;1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\end{array}\right)
\end{array}\right)\\
+\mathcal{D}_{1,n_x} g_{ss}\left(s_t\right)^{\intercal}\left(\begin{array}{c}
\mathcal{H}_{n_y+1,2n_y;1,n_y} f^i_{ss}\left(s',s_t\right)\left(\begin{array}{c}\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right)\\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\end{array}\right)\\
+\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\\
+\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\\
+\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s'\right)\\
+\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s_t\right)
\end{array}\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_x+n_y);1,n_y} f^i_{ss}\left(s',s_t\right)\left(\begin{array}{c}\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right)\\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\end{array}\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_y+n_x);n_y+1,2n_y} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_y+n_x);2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_x+n_y);2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right)\mathcal{D}\theta_{ss}\left(s'\right)\\
+\mathcal{H}_{2n_y+n_x+1,2(n_x+n_y);2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right)\theta_\chi\left(s\right)\\
+\left(\begin{array}{c}\mathcal{D}_{1,n_y} f^i_{ss}\left(s',s_t\right)\mathcal{D}_{1,n_x} g_{ss}\left(s'\right)\\ +\mathcal{D}_{2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right)\end{array}\right)\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} h_{ss}\left(s_t\right)\\
+\mathcal{D}_{n_y+1,2n_y} f^i_{ss}\left(s',s_t\right)\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)
\end{array}\right)$$

Given the solution from Step 2, the condition $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t) = 0_{n_x}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution $\left\{\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}g^i_{ss}(s_t)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}h^i_{ss}(s_t)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

11.4 Step 4: Obtaining the derivatives of ε_{t-1} twice

The fourth equation is the derivative with respect to ε_{t-1} twice.

$$\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}(s_t) = \sum_{s'=1}^{n_s} p_{s_t,s'} \times$$

$$\left(\begin{array}{c}
[\mathcal{D}_{1,n_x}g_{ss}(s')\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}(s_t)]^\intercal \left[\begin{array}{c}\left[\begin{array}{c}\mathcal{H}_{1,n_y;1,n_y}f^i_{ss}(s',s_t)\,\mathcal{D}_{1,n_x}g_{ss}(s') \\ +2\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\end{array}\right]\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}(s_t) \\ +2\mathcal{H}_{1,n_y;n_y+1,2n_y}f^i_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s_t) \\ +2\mathcal{H}_{1,n_y;2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}(s',s_t)\end{array}\right] \\
+\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}(s_t)^\intercal \left[\begin{array}{c}\left[\begin{array}{c}\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t) \\ +\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\,\mathcal{H}_{1,n_x;1,n_x}g_{ss}(s')\end{array}\right]\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}(s_t) \\ +2\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}(s',s_t)\end{array}\right] \\
+\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s_t)^\intercal \left[\begin{array}{c}\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y}f^i_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s_t) \\ +2\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\,\mathcal{D}_{n_x+1,n_x+n_\varepsilon}h_{ss}(s_t) \\ +2\mathcal{H}_{n_y+1,2n_y;2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}(s',s_t)\end{array}\right] \\
+\left[\begin{array}{c}\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\,\mathcal{D}_{1,n_x}g_{ss}(s') \\ +\mathcal{D}_{2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\end{array}\right]\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}h_{ss}(s_t) \\
+\mathcal{D}_{n_y+1,2n_y}f^i_{ss}(s',s_t)\,\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}g_{ss}(s_t) \\
+\mathcal{H}_{2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon)}f^i_{ss}(s',s_t)
\end{array}\right)$$

Given the solution from Step 1, the condition $\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}\mathbb{G}^i_{ss}(s_t) = 0_{n_\varepsilon \times n_\varepsilon}$ for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution $\left\{\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}g^i_{ss}(s_t)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}h^i_{ss}(s_t)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

11.5 Step 5: Obtaining the derivatives of ε_{t-1} and χ

The fifth equation is the derivative with respect to ε_{t-1} and χ.

$$\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}\left(s_t\right) = \sum_{s'=1}^{n_s} p_{s_t,s'} \times$$

$$\left(\begin{array}{c}
\left(\mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}\left(s_t\right)\right)^{\intercal} \left(\begin{array}{c}
\mathcal{H}_{1,n_y;1,n_y} f^i_{ss}\left(s',s_t\right) \left(\begin{array}{c} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right) \\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \end{array}\right) \\
+\mathcal{H}_{1,n_y;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right) \\
+\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \\
+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s'\right) \\
+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s_t\right)
\end{array}\right) \\
+\mathcal{D}_{n_x+1,n_x+n_\varepsilon} h_{ss}\left(s_t\right)^{\intercal} \left(\begin{array}{c}
\mathcal{H}_{2n_y+1,2n_y+n_x;1,n_y} f^i_{ss}\left(s',s_t\right) \left(\begin{array}{c} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right) \\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \end{array}\right) \\
+\mathcal{H}_{2n_y+1,2n_y+n_x;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right) \\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s'\right) \\
+\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s_t\right) \\
\mathcal{D}_{1,n_y} f^i_{ss}\left(s',s_t\right) \left(\begin{array}{c} \mathcal{H}_{1,n_x;n_x+n_\varepsilon+1} g_{ss}\left(s'\right) \\ +\mathcal{H}_{1,n_x;1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \end{array}\right)
\end{array}\right) \\
+\mathcal{D}_{n_x+1,n_x+n_\varepsilon} g_{ss}\left(s_t\right)^{\intercal} \left(\begin{array}{c}
\mathcal{H}_{n_y+1,2n_y;1,n_y} f^i_{ss}\left(s',s_t\right) \left(\begin{array}{c} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right) \\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \end{array}\right) \\
+\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right) \\
+\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \\
+\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s'\right) \\
+\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s_t\right)
\end{array}\right) \\
+\mathcal{H}_{2(n_x+n_y)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);1,n_y} f^i_{ss}\left(s',s_t\right) \left(\begin{array}{c} \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s'\right) \\ +\mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \end{array}\right) \\
+\mathcal{H}_{2(n_x+n_y)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);n_y+1,2n_y} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} g_{ss}\left(s_t\right) \\
+\mathcal{H}_{2(n_x+n_y)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \\
+\mathcal{H}_{2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s'\right) \\
+\mathcal{H}_{2(n_y+n_x)+n_\varepsilon+1,2(n_x+n_y+n_\varepsilon);2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)} f^i_{ss}\left(s',s_t\right) \mathcal{D}\theta_{ss}\left(s_t\right) \\
+\left(\begin{array}{c} \mathcal{D}_{1,n_y} f^i_{ss}\left(s',s_t\right) \mathcal{D}_{1,n_x} g_{ss}\left(s'\right) \\ +\mathcal{D}_{2n_y+1,2n_y+n_x} f^i_{ss}\left(s',s_t\right) \end{array}\right) \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} h_{ss}\left(s_t\right) \\
+\mathcal{D}_{n_y+1,2n_y} f^i_{ss}\left(s',s_t\right) \mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1} g_{ss}\left(s_t\right)
\end{array}\right)$$

Given the solution from Steps 1 and 3, the condition $\mathcal{H}_{n_x+1,n_x+n_\varepsilon,n_x+n_\varepsilon;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t)=0_{n_\varepsilon}$ for $1\leq i\leq n_y+n_x$ implies a linear system of equations that determines the solution $\left\{\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}g^i_{ss}(s_t)\right\}_{i=1}^{n_y},\left\{\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+n_\varepsilon+1}h^i_{ss}(s_t)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.

11.6 Step 6: Obtaining the derivatives of χ twice

The sixth equation is the derivative with respect to χ twice.

$$\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t)=\sum_{s'=1}^{n_s}p_{s_t,s'}\times$$

$$\left(\begin{array}{c}
\begin{pmatrix}\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s')\\+\mathcal{D}_{1,n_x}g_{ss}(s')\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\end{pmatrix}^{\intercal}\mathcal{H}_{1,n_y;1,n_y}f^i_{ss}(s',s_t)\begin{pmatrix}\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s')\\+\mathcal{D}_{1,n_x}g_{ss}(s')\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\end{pmatrix}\\
+2\begin{pmatrix}\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s')\\+\mathcal{D}_{1,n_x}g_{ss}(s')\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\end{pmatrix}^{\intercal}\begin{pmatrix}\mathcal{H}_{1,n_y;n_y+1,2n_y}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s_t)\\+\mathcal{H}_{1,n_y;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\\+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s')\\+\mathcal{H}_{1,n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s_t)\end{pmatrix}\\
+\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s_t)^{\intercal}\begin{pmatrix}\mathcal{H}_{n_y+1,2n_y;n_y+1,2n_y}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+n_\varepsilon+1}g_{ss}(s_t)\\+2\mathcal{H}_{n_y+1,2n_y;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\\+2\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s')\\+2\mathcal{H}_{n_y+1,2n_y;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s_t)\end{pmatrix}\\
+\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)^{\intercal}\begin{pmatrix}\mathcal{H}_{2n_y+1,2n_y+n_x;2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\\+2\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s')\\+2\mathcal{H}_{2n_y+1,2n_y+n_x;2(n_x+n_y+n_\varepsilon+n_\theta)+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s_t)\\+\left[\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\mathcal{H}_{1,n_x;1,n_x}g_{ss}(s')\right]\mathcal{D}_{n_x+n_\varepsilon+1}h_{ss}(s_t)\\+2\left[\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\mathcal{H}_{1,n_x;n_x+n_\varepsilon+1}g_{ss}(s')\right]\end{pmatrix}\\
+(\varepsilon')^{\intercal}\begin{pmatrix}\mathcal{H}_{2(n_x+n_y)+1,2(n_x+n_y)+n_\varepsilon;2(n_x+n_y)+1,2(n_x+n_y)+n_\varepsilon}f^i_{ss}(s',s_t)\\+\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\mathcal{H}_{n_x+1,n_x+n_\varepsilon;n_x+1,n_x+n_\varepsilon}g_{ss}(s')\\+\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s')'\mathcal{H}_{1,n_y;1,n_y}f^i_{ss}(s',s_t)\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s')\\+2\mathcal{D}_{n_x+1,n_x+n_\varepsilon}g_{ss}(s')'\mathcal{H}_{1,n_y;2(n_y+n_x)+1,2(n_y+n_x)+n_\varepsilon}f^i_{ss}(s',s_t)\end{pmatrix}\varepsilon'\\
+\mathcal{D}\theta_{ss}(s')^{\intercal}\begin{pmatrix}\mathcal{H}_{2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta;2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s')\\+2\mathcal{H}_{2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta;2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s_t)\end{pmatrix}\\
+\mathcal{D}\theta_{ss}(s_t)^{\intercal}\mathcal{H}_{2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta);2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{D}\theta_{ss}(s_t)\\
+\begin{pmatrix}\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\mathcal{D}_{1,n_x}g_{ss}(s')\\+\mathcal{D}_{2n_y+1,2n_y+n_x}f^i_{ss}(s',s_t)\end{pmatrix}\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}h_{ss}(s_t)\\
+\mathcal{D}_{1,n_y}f^i_{ss}(s',s_t)\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}g_{ss}(s')\\
+\mathcal{D}_{n_y+1,2n_y}f^i_{ss}(s',s_t)\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}g_{ss}(s_t)\\
+\mathcal{D}_{2(n_x+n_y+n_\varepsilon)+1,2(n_x+n_y+n_\varepsilon)+n_\theta}f^i_{ss}(s',s_t)\mathcal{H}\theta_{ss}(s')\\
+\mathcal{D}_{2(n_x+n_y+n_\varepsilon)+n_\theta+1,2(n_x+n_y+n_\varepsilon+n_\theta)}f^i_{ss}(s',s_t)\mathcal{H}\theta_{ss}(s_t)
\end{array}\right)$$

Given the solutions from Steps 1, 3 and 4, the condition $\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1}\mathbb{G}^i_{ss}(s_t)=0$

for $1 \leq i \leq n_y + n_x$ implies a linear system of equations that determines the solution $\left\{\left\{\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1} g_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_y}, \left\{\mathcal{H}_{n_x+n_\varepsilon+1;n_x+n_\varepsilon+1} h_{ss}^i\left(s_t\right)\right\}_{i=1}^{n_x}\right\}_{s_t=1}^{n_s}$.