Comparing Accuracy of Second Order Approximation and Dynamic Programming*

Stephanie Becker†, Lars Grüne‡and Willi Semmler§

June 2006, revised version January 2007

Abstract

The accuracy of the solution of dynamic general equilibrium models has become a major issue. Recent papers, in which second order approximations have been substituted for first order, indicate that this change may yield a significant improvement in accuracy. Second order approximations have been used with considerable success when solving for the decision variables in both small and large-scale models. Additionally, the issue of accuracy is relevant for the approximate solution of value functions. In numerous dynamic decision problems, welfare is usually computed via this same approximation procedure. However, Kim and Kim (2003) have found a reversal of welfare ordering when they moved from first to second order approximations. Other researchers, studying the impact of monetary and fiscal policy on welfare, have faced similar challenges with respect to the accuracy of approximations of the value function. Employing a base-line stochastic growth model, this paper compares the accuracy of second order approximations and dynamic programming solutions for both the decision variable and the value function as well. We find that, in a neighborhood of the equilibrium, the second order approximation method performs satisfactorily; however, on larger regions, dynamic programming performs significantly better with respect to both the decision variable and the value function.

*We want to thank Jinill Kim and Harald Uhlig for discussions and a referee of the Journal for extensive comments on a previous version of our paper

†c/o Lars Grüne, Mathematical Institute, University of Bayreuth

‡Mathematical Institute, University of Bayreuth

§Schwartz Center for Economic Policy Analysis, New School, New York, and CEM, Bielefeld

1 Introduction

Much effort has been expended studying the accuracy of solutions to dynamic general equilibrium models. There are already many papers that focus on the policy function while comparing different solution methods. Such studies can be found, for example, in Taylor and Uhlig (1990), Den Haan and Marcet (1994), Christiano and Fisher (2000), and Aruoba, et al (2004). The current paper can be seen as an extension of this earlier work. However, within the context of a dynamic programming approach, we study not only the accuracy of the policy function, but the accuracy of the value function as well.

Recent work on the solving of dynamic general equilibrium models and on the accuracy of estimates of the policy function has generally followed the perturbation method suggested in Judd (1996, 1998) and Jin and Judd (2005). Researchers such as Collard and Juillard (2001) and Schmitt-Grohe and Uribe (2004a, 2005a, 2005b) have had much success using this technique by substituting second order for first order Taylor approximations in the decision rule. As a consequence, second order approximations are now widely applied to small and large-scale models.[1]

Despite all of this research, the extent to which a second order approximation can be used to proxy the value function is still under-explored. The importance of the value function is revealed in, for example, the context of asset-pricing models.[2] The computation of the value function is also essential in welfare analysis, particularly whenever one wishes to undertake a welfare comparison of policy rules in dynamic macro models. Kim and Kim (2003) have presented an important result in which a reversal of welfare ordering appears as a consequence of the approximation method. Specifically, for a first order approximation of the value function, the welfare of a closed economy is higher than for an open economy with complete markets. However, when a second order approximation is used, the ordering is completely reversed.

Researchers who study the impact of fiscal policy on welfare also have to deal not only with the accuracy of the decision variable, but with the accuracy of the value function as well.[3] These accuracy issues are particularly important in the study of monetary policy models, which also use the dynamic stochastic general equilibrium framework, and have been employed in the analysis of welfare implications for economies with market friction.[4] Benigno and Woodford (2006b) have addressed the issue of accuracy more generally in their study of policy models. They examine linear-quadratic (LQ) models, their variants, and investigate the degree to which LQ-models approximate the exact objectives and constraints of the corresponding dynamic macro-model.

Numerical work on second order approximations of policy functions, e.g., the decision rule, has already been done. Collard and Juillard (2001) investigate the problem by exploring the impact of the curvature of the value function along with other parameters, e.g., the discount rate, on the decision rule. They also derive a rough bound on the size of the errors

[1] For examples, see Schmidt-Grohe and Uribe (2004b) and Smets and Wouters (2004).

[2] Collard and Juillard (2001) for an illustration.

[3] Benigno and Woodford (2006a,b).

[4] Smets and Wouters (2004)

generated in second order approximations of the value function. Additionally, Schmitt-Grohe and Uribe (2004a) have put forward MATLAB-implemented routines for computing second order approximations of the decision variable. They do not concern themselves with the welfare function; thus, they do not investigate the approximation error in the value function itself.

As aforementioned, the accuracy issue was brought to our attention by the work of Kim and Kim (2004, 2006), wherein the accuracy of linear and second order approximations of the value function are explored in a simplified variant of a dynamic equilibrium model. The accuracy of the value function was addressed in the work of Benigno and Woodford (2006a,b). As in Judd (1998, ch. 13), they pursue the problem as to how a "naive" linear quadratic approximation, one that uses quadratic objective functions and linear state equations, can be misleading. They show that it is more accurate to use a proper second order approximation using the first order optimality conditions as a starting point. Further, they show that using the second order approximation of the first order conditions and the state equation gives a more accurate solution of both the decision variable as well as the welfare function. While Benigno and Woodford (2006a,b) use the second order information in order to obtain an approximative linear quadratic problem, which in particular yields an approximately optimal policy in linear feedback form, Schmidt-Grohe and Uribe (2004a, 2005a, 2005b) use a quadratic approximation of the optimal policy. However, the main finding remains the same in both methods, i.e., that second order approximations are satisfactory and appear to be superior "in the case of small enough disturbances" of the steady state.[5]

Our paper goes a step further by arguing that second order approximations may not, in general, deliver sufficiently accurate results. By employing a base-line stochastic growth model, we compare the second order approximation with the dynamic programming solution for the decision variable and for the value function. We pay less attention to the impact of the parameters on the accuracy of the solution, but rather on the size of the shocks; specifically, when measured by the distance of the state variable from its steady state, we investigate the impact of shocks on the accuracy of both the decision variable and the value function. We find that, in a neighborhood of the equilibrium, the second order approximation method performs satisfactorily; however, on larger regions, dynamic programming performs significantly better with respect to both the decision variable and the value function.

The paper is organized as follows: Section 2 introduces a base-line stochastic growth model and tests the accuracy of the second order approximation. Here, we also discuss the extent to which the stochastic dynamics affect the accuracy of the solution. Section 3 solves the base-line model with dynamic programming and also studies the accuracy of the solution with respect to the size of the shocks. Section 4 concludes the paper. The appendix gives a summary of the MATLAB files and the dynamic programming method.

[5]Benigno and Woodford (2006b)

2 Second Order Approximation

This section deals with the application of the second order approximation to solve the stochastic version of the Ramsey growth model. Section 2.1 lays out the problem and adapts and applies the optimality conditions. The model is also solved for its equilibrium. Section 2.2 presents and discusses the approximation results. In section 2.3 we present an extension of the simple model where the stochastic dynamics is now more pronounced. We show of how a change in the objective function affects the approximation coefficients.

2.1 The Base Line Model

The Ramsey model is an economic growth model, which is taken in its presented, stochastic setup from Santos and Vigo-Aguiar (1998) and Grüne and Semmler (2004).The model[6] is two dimensional in its state variables, x, and one dimensional in its control variable, u. The functional form of the objective function is:

$$\max_{u_t} E\left[\sum_{t=0}^{\infty} \beta^t \ \log u_t\right] \tag{1}$$

with the control system $x_{t+1} = \varphi(x_t, u_t, z_t)$ where φ is given by:

$$\varphi(x, u, z) = \begin{pmatrix} A\, e^{x_2}\, x_1^{\alpha} - u \\ \rho x_2 + z \end{pmatrix}. \tag{2}$$

Here $x_{t,1} > 0$, $x_{2,t} \in \mathbb{R}$ and z_t being i.i.d. random variables which are Gauss-normal distributed with mean 0 and standard deviation $\sigma = 0.008$.

This problem is well suited to test both the second order approximation and the dynamic programming method, because

(i) The solution is sufficiently smooth for $x_1 \neq 0$.

(ii) The exact solution for the value function is known and defined by:

$$V(x) = B \ + \ C \ \ln x_1 \ + \ D\, x_2 \tag{3}$$

with

$$\begin{aligned} B &= \frac{\ln((1-\beta\alpha)\, A) + \frac{\beta\alpha}{1-\beta\alpha} \ln(\beta\alpha A)}{(1-\beta)} \\ C &= \frac{\alpha}{1-\alpha\beta} \\ D &= \frac{1}{(1-\alpha\beta)(1-\rho\beta)}. \end{aligned}$$

[6] Variations of the baseline model are solved and estimated in Gong and Semmler (2006).

The exact solution for u is defined by

$$u(x) = (1 - \alpha\beta)\, A\, e^{x_2}\, x_1^{\alpha}. \tag{4}$$

In our computations we use $A = 5$, $\alpha = 0.34$, $\beta = 0.95$ and $\rho = 0.9$. For convenience, the Gauss-normal distributed random variable z can be normalized with the variance σ^2 such that $\frac{z}{\sigma} \sim N(0,1)$. This yields $z = \sigma \cdot \varepsilon$, where ε is an $N(0,1)$-random variable independent of x.[7]

This transformation provides us with the form of the model as proposed of Schmitt-Grohe and Uribe (2004). The variable x_1 follows a deterministic dynamic and the variable x_2 a stochastic dynamic. Note that variables of period $(t+1)$ are denoted by a' and variables of period t have no time index:

$$\begin{aligned} x_1' &= A\, e^{x_2} x_1^{\alpha} - u \\ x_2' &= \rho x_2 + \sigma\varepsilon. \end{aligned}$$

The exact calculations of the value function and the optimal control also allow us to determine the accuracy of the procedures, see figure 1.

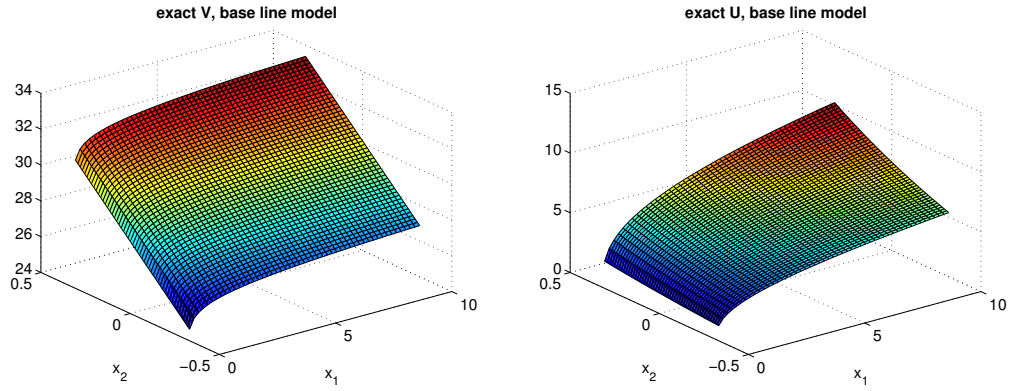

Figure 1: The exact optimal value function $V(x)$ and optimal control $u(x)$

The idea of the second order approximation, which is based on the perturbation method,[8] and which we apply here consists in assuming that the optimal decisions and the optimal dynamics can be expressed as

$$u(x) = g(x,\sigma) \quad \text{and} \quad x' = h(x,\sigma)$$

for smooth functions g and h. Then, these functions are approximated via their second order Taylor approximation in $x = \bar{x}$ and $\sigma = 0$, where $\bar{x}$ is an optimal equilibrium point of the unperturbed system, i.e., for $\sigma = 0$. This parameter σ is called the perturbation parameter.

[7] See Aström, (1970) for a detailed discussion of such a transformation.

[8] see Judd (1996) and (1998, ch. 13)

In order to apply the method, we need to determine the necessary optimality conditions derived from the Hamiltonian function and the optimal equilibrium $(\bar{x}, \bar{u})$ for the unperturbed system, which can be computed from these conditions. These conditions are given by[9]:

$$\begin{aligned} E_t\left[x_{t+1}\right] &= \left(\frac{\partial H^t}{\partial \lambda_{t+1}}\right)^T && \text{optimality condition for the state variable} \\ \lambda_t &= \left(\frac{\partial H^t}{\partial x_t}\right)^T && \text{optimality condition for the co-state variable} \\ 0 &= \left(\frac{\partial H^t}{\partial u_t}\right)^T && \text{maximum condition} \end{aligned}$$

where the Hamiltonian reads:

$$H^t(x_t, u_t, \lambda_t) = E_t\left[\beta^{t+1} \log u_t + \lambda_{t+1,1} \cdot (A\, e^{x_{t,2}} x_{t,1}^{\alpha} - u_t) + \lambda_{t+1,2} \cdot (\rho x_{t,2} + \sigma \varepsilon_t)\right].$$

This yields the following conditions:

- optimality condition for the state variable:

$$\begin{aligned} E_t\left[x_{t+1,1}\right] &= E_t\left[A\, e^{x_{t,2}}\, x_{t,1}^{\alpha} - u_t\right] \\ \Longrightarrow 0 &= E_t\left[x_{t+1,1} - A\, e^{x_{t,2}}\, x_{t,1}^{\alpha} + u_t\right] \qquad (5) \\ \text{and} & \\ E_t\left[x_{t+1,2}\right] &= E_t\left[\rho x_{t,2} - \sigma \varepsilon_t\right] \\ \Longrightarrow 0 &= E_t\left[x_{t+1,2} - \rho x_{t,2} + \sigma \varepsilon_t\right] \qquad (6) \end{aligned}$$

- optimality condition for the co-state variable:

$$\begin{aligned} \lambda_{t,1} &= E_t\left[\lambda_{t+1,1} \alpha A\, e^{x_{t,2}}\, x_{t,1}^{\alpha-1}\right] \qquad (7) \\ \lambda_{t,2} &= E_t\left[\lambda_{t+1,2}\rho + \lambda_{t+1,1}(A e^{x_{t,2}} x_{t,1}^{\alpha})\right] \qquad (8) \end{aligned}$$

- maximum condition:

$$\begin{aligned} E_t\left[\beta^{t+1}\, \frac{1}{u_t} - \lambda_{t+1,1}\right] &= 0 \\ \Longrightarrow \frac{\beta^{t+1}}{u_t} &= E_t\left[\lambda_{t+1,1}\right] \qquad (9) \end{aligned}$$

We can simplify the conditions as follows: Substituting the maximum condition (9) in the co-state equation (7) (noting that in the expectation E_t the variable $\lambda_{t+1,1}$ is the only random variable in (7)) yields

$$\lambda_{t,1} = \frac{\beta^{t+1}}{u_t} \alpha A\, e^{x_{t,2}}\, x_{t,1}^{\alpha-1}.$$

[9] For details see Becker (2005).

Using this equation for $t+1$ instead of t, taking expectations E_t and substituting (9) on the left hand side yields the condition:

$$0 = E_t \left[\frac{\beta}{u_{t+1}} \, \alpha \, A \, e^{x_{t+1,2}} \, x_{t+1,1}^{(\alpha-1)} - \frac{1}{u_t} \right] \tag{10}$$

Written in a compact way (using again $x = x_t$ and $x' = x_{t+1}$ etc.), the optimality conditions are thus:

$$\boxed{E_t(f(x, x', u, u')) = 0 \quad \text{for} \quad f(x, x', u, u') := \begin{pmatrix} x_1' - A \, e^{x_2} \, x_1^{\alpha} + u \\ x_2' - \rho \, x_2 \\ \frac{\beta}{u'} \, \alpha \, A \, e^{x_2'} \, x_1'^{(\alpha-1)} - \frac{1}{u} \end{pmatrix}} \tag{11}$$

Remark 2.1 *The function f, defined in equ.* (11)*, does not include the second co-state optimality condition* (8) *because this equation does not add any information to the equilibrium conditions since $\lambda_{t,2}$ is not contained in the maximum condition* (9)*.*

Note that (11) is already in the form needed for the second order approach, because the MATLAB–programs from Schmitt-Grohe, S. and M. Uribe (2005b) compute the second order approximation of g and h directly from f. Hence, we do not need to solve for the steady state of (11) for the general model.

We do, however, need to solve for the optimal equilibrium $(\bar{x}, \bar{u})$ of (11) for perturbation parameter $\sigma = 0$, because $(\bar{x}, \sigma)$ is the point the second order Taylor approximation expanded around and $\bar{u}$ is needed as the zero order term in g. This is much easier than the general case because for $\sigma = 0$ the model becomes deterministic, i.e., in the steady state we have:

$$\begin{aligned} x_1' &= x_1, \\ x_2' &= x_2, \\ u' &= u \end{aligned}$$

Now, from the second row of equ. (11)

$$x_2' - \rho \, x_2 = 0$$

the equilibrium value for $\bar{x}_2$ is

$$\bar{x}_2 = 0.$$

This result in combination with the first row of (11) yields

$$u = A \cdot 1 \cdot x_1^{\alpha} - x_1. \tag{12}$$

Using this in the third row of (11) one is able to express x_1 in known variables:

$$\begin{aligned} & \beta\,\alpha\,A\,x_1^{(\alpha-1)}\,\frac{1}{A\,x_1^{\alpha}-x_1} = \frac{1}{A\,x_1^{\alpha}-x_1} \\ \Longleftrightarrow \quad & \beta\,\alpha\,A\,x_1^{(\alpha-1)} = 1 \\ \Longleftrightarrow \quad & \beta\,\alpha\,A = x_1^{(1-\alpha)} \\ \Longleftrightarrow \quad & \left(\ \beta\,\alpha\,A\ \right)^{\frac{1}{1-\alpha}} = x_1 \end{aligned}$$

Using the preset parameter values, x_1 equals

$$\begin{aligned} \left(\ 0.95\cdot 0.34\cdot 5\ \right)^{\frac{1}{0.66}} &= x_1 \\ x_1 &\approx 2.067344815 \end{aligned}$$

¿From equ. (12) we know:

$$\begin{aligned} 5\cdot(2.067344815)^{0.34} - 2.067344815 &= u \\ u &\approx 4.333103529 \end{aligned}$$

which yields the equilibrium values

$$\bar{x} = (2.067344815, 0), \quad \bar{u} = 4.333103529$$

These values serve as the starting point for testing the second order approximation as a solution procedure. In the Ramsey model the control variable, u, denotes consumption, the first state variable, x_1, the capital stock and the transformed second state variable, e^{x_2}, the technology shock. This means that with our parameter specifications an equilibrium exists for the consumption at 4.33, for the capital stock at 2.07 and for the technology shock at 1.

2.2 Numerical Results

We employ the numerical procedure and the numerical subroutines of Appendix A, respectively, to obtain the second order approximations. We present the numerical results of our computations and compare those to the exact solution.

In order to compute the second order Taylor approximations of g and h about the equilibrium

$$\left[\ \bar{x};\ \ \bar{u}\ \right] \ = \ \left[\ (2.067344815, 0);\ \ 4.333103529\ \right],$$

we need the first and second order derivatives of these functions with respect to x and σ. The MATLAB programs, as reported in Appendix A, compute for the approximation the following coefficients:

- Coefficients of the linear terms:

$$g_{x_1} = 0.7126, \quad g_{x_2} = 4.3331$$

$$h_{x_1} = \begin{pmatrix} 0.3400 \\ 0 \end{pmatrix}, \quad h_{x_2} = \begin{pmatrix} 2.0673 \\ 0.9 \end{pmatrix}.$$

- Coefficients of the quadratic terms for x:

$$g_{x_1x_1} = -0.2275, \quad g_{x_1x_2} = g_{x_2x_1} = 0.7126, \quad g_{x_2x_2} = 4.3331$$

$$h_{x_1x_1} = \begin{pmatrix} -0.1085 \\ 0 \end{pmatrix}, \quad h_{x_1x_2} = h_{x_2x_1} = \begin{pmatrix} 0.3400 \\ 0 \end{pmatrix}, \quad h_{x_2x_2} = \begin{pmatrix} 2.0673 \\ 0 \end{pmatrix}.$$

- Coefficients of the quadratic terms for σ:

$$g_{\sigma\sigma} = 1.5677 \cdot 10^{-10}, \quad h_{\sigma\sigma} = \begin{pmatrix} -0.1568 \cdot 10^{-9} \\ 0 \end{pmatrix}.$$

Note: since the last two terms are approximately zero, the stochastic part plays a negligible role in the approximation. This is also immediately clear from the expressions for the exact solution in equs. (3) and (4), because these formulae reveal that neither the optimal control policy nor the optimal value function depend on σ.

The above values yield for the approximations of the decision variable $u = g(x, \sigma)$ and the state variables x_1 and x_2, following Appendix A, the following:

$$\begin{aligned}
\widehat{u} \;=\; & 4.333103529 + 0.7126 \cdot (x_1 - 2.067344815) + 4.3331 \cdot (x_2 - 0) \\
& + \frac{1}{2} \cdot (-0.2275) \cdot (x_1 - 2.067344815) \cdot (x_1 - 2.067344815) \\
& + \frac{1}{2} \cdot 0.7126 \cdot (x_2 - 0) \cdot (x_1 - 2.067344815) \\
& + \frac{1}{2} \cdot 0.7126 \cdot (x_1 - 2.067344815) \cdot (x_2 - 0) \\
& + \frac{1}{2} \cdot 4.3331 \cdot (x_2 - 0) \cdot (x_2 - 0) \\
& + \frac{1}{2} \cdot 0 \cdot 0.008 \cdot 0.008.
\end{aligned}$$

$$
\begin{aligned}
\widehat{x}_1^{'} &= 2.067344815 + 0.3400 \cdot (x_1 - 2.067344815) + 2.0673 \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot (-0.1085) \cdot (x_1 - 2.067344815) \cdot (x_1 - 2.067344815) \\
&\quad + \frac{1}{2} \cdot 0.34 \cdot (x_2 - 0) \cdot (x_1 - 2.067344815) \\
&\quad + \frac{1}{2} \cdot 0.34 \cdot (x_1 - 2.067344815) \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot 2.0673 \cdot (x_2 - 0) \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot 0 \cdot 0.008 \cdot 0.008,
\end{aligned}
$$

$$
\begin{aligned}
\widehat{x}_2^{'} &= 0 + 0 \cdot (x_1 - 2.067344815) + 0.9 \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot 0 \cdot (x_1 - 2.067344815) \cdot (x_1 - 2.067344815) \\
&\quad + \frac{1}{2} \cdot 0 \cdot (x_2 - 0) \cdot (x_1 - 2.067344815) \\
&\quad + \frac{1}{2} \cdot 0 \cdot (x_1 - 2.067344815) \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot 0 \cdot (x_2 - 0) \cdot (x_2 - 0) \\
&\quad + \frac{1}{2} \cdot 0 \cdot 0.008 \cdot 0.008 \\
&\quad + \sigma \varepsilon^{'}.
\end{aligned}
$$

In order to discuss the result of the approximation, we want to compare both the approximated optimal policy $\widehat{u}$ and the approximated optimal value function $\widehat{V}$. Knowing the approximated optimal control $\widehat{u}(x)$ and the approximate optimal dynamic $\widehat{h}(x) = (\widehat{x}_1', \widehat{x}_2')$ from the approximation above, we have two ways of approximating V.

The first method we use is a second order approximation to V: Similar to the procedure described in Collard and Juillard (2001), we use second order Taylor approximations of both sides of the dynamic programming principle

$$
V(x) = E\left[\log \widehat{u}(x) + \beta V(\widehat{h}(x))\right] \tag{13}
$$

in order to obtain a linear system of equations for the coefficients of the second order Taylor approximation of V, which can be solved using an appropriate MAPLE procedure (cf. Appendix A). For the values given above we obtain the coefficients

$$
V_0 = 29.32566, \;\; V_{x_1} = 0.24291, \;\; V_{x_2} = 10.18671
$$

$$V_{x_1x_1} = -0.11749,\ \ V_{x_1x_2} = V_{x_2x_1} = 0.14047 \cdot 10^{-4},\ \ V_{x_2x_2} = 0.47826 \cdot 10^{-3},\ \ V_{\sigma\sigma} = 0.$$

The resulting second order approximation for the optimal value function is denoted by $\widetilde{V}$. Note that using a second order approximation of the optimal value function is analogous to Kim and Kim (2003) and Benigno and Woodford (2006b).

An alternative way of computing an approximation to V is obtained by evaluating the functional (1) using the approximately optimal control $\widehat{u}$, i.e., by computing

$$V_{\widehat{u}}(x) = E\left[\sum_{t=0}^{\infty} \beta^t \log \widehat{u}(x_t)\right] \tag{14}$$

with $x_0 = x$ and $x_{t+1} = \varphi(x_t, \widehat{u}(x_t), z_t)$ and φ from (2). A simple calculation yields

$$V_{\widehat{u}}(x) = E\left[\log \widehat{u}(x) + \beta V_{\widehat{u}}(\varphi(x, \widehat{u}(x), z))\right] \tag{15}$$

We evaluate the expectation using a numerical quadrature rule. We use the rectangle rule with 11 subintervals for $z \in [-0.032, 0.032]$) approximating $V_{\widehat{u}}$ by a piecewise multilinear function $\widehat{V}$ on a rectangular grid. We employ equidistant grids with 51×51 nodes. The computational domain is given below. Equ. (15) becomes a system of linear equations which can be solved by standard solvers; we used a Jacobi–iteration starting from $\widetilde{V}$ which is reasonably fast and easy to implement in MATLAB. Note that the system of equations which needs to be solved for computing $\widehat{V}$ contains $51 \cdot 51 = 2601$ unknowns while the system of equations to be solved for $\widetilde{V}$ contains just 7 unknowns. In particular, the computation of $\widehat{V}$, which essentially employs a dynamic programming technique, is subject to the curse of dimensionality while the computation of $\widetilde{V}$ is not.

Next we provide a graphical analysis of the errors. The figures have been created with the MATLAB file `comparison.m` which is described in Appendix A. Numerical values for the errors can be found in Table 1, below. For the graphical analysis of the error we consider three domains.

2.2.1 The domain $[2.0, 2.1] \times [-0.025, 0.025]$

At first, the estimates are analyzed in a rather small region around the equilibrium $\hat{x} \approx (2.067,\ 0)$. Figure 2 depicts the approximation errors $|V - \widetilde{V}|$, $|V - \widehat{V}|$ and $|u - \widehat{u}|$ on this domain.

All errors are very small around the equilibrium and increase slightly the larger the deviation. Note that for the computation of the coefficients of $\widetilde{V}$, the coefficients of $\widehat{u}$ and $\widehat{h}$ were truncated to five digits, thus the obtained accuracy in the order of 10^{-5} is the best one can expect.

The approximated value function $\widehat{V}$ in Figure 2 approximation were computed on $[1, 4] \times [-0.32, 0.32]$ with 51×51 nodes; for simplicity, the same grid was used in the evaluation of $\widehat{V}$ and $\widehat{u}$. Note that we can only solve (15) on an invariant domain Ω for $\varphi(x, \widehat{u}(x), z)$ (i.e., we need $\varphi(x, \widehat{u}(x), z) \in \Omega$ for all $x \in \Omega$ and all values of the random variable z used in the

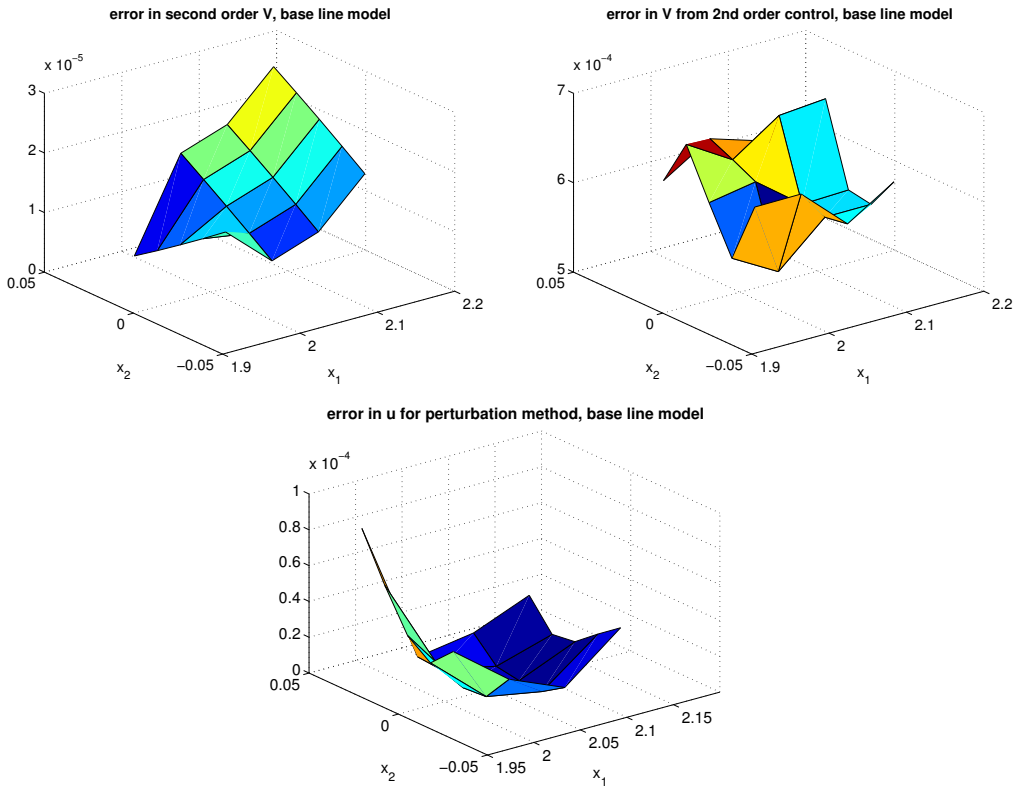

Figure 2: Approximation error $|V - \widetilde{V}|$ (top left), $|V - \widehat{V}|$ (top right) and $|u - \widehat{u}|$ (bottom) on $[2.0, 2.1] \times [-0.025, 0.025]$

numerical quadrature for evaluating the expectation). Thus, only a few nodes of the grid lie in our small domain $[2.0, 2.1] \times [-0.025, 0.025]$, which explains the "ragged" appearance of the figures. Nevertheless, even with this rather coarse interpolation the approximation error is below $7 \cdot 10^{-4}$.

2.2.2 The domain $[1, 4] \times [-0.32, 0.32]$

Using the same numerical approximation as before we now show the error on the domain $[1, 4] \times [-0.32, 0.32]$.

Figure 3 depicts the approximation errors $|V - \widetilde{V}|$, $|V - \widehat{V}|$ and $|u - \widehat{u}|$.

Here one can observe that the approximation error in both $\widehat{u}$ and $\widetilde{V}$ considerably increases away from the equilibrium. On this domain, however, this error in $\widehat{u}$ still has only a small effect on $\widehat{V}$, because the regions where $|u - \widehat{u}|$ is large are visited only with small probabilities in (14) for x in this domain. This situation changes if we further increase the domain.

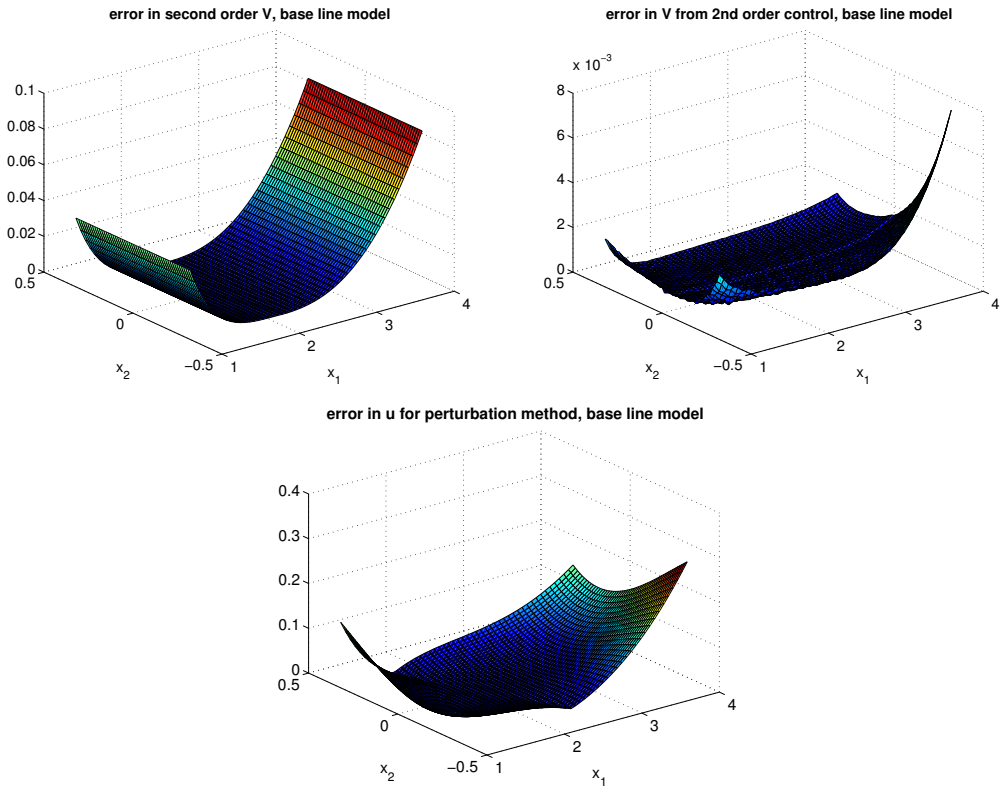

Figure 3: Approximation error $|V - \widetilde{V}|$ (top left), $|V - \widehat{V}|$ (top right) and $|u - \widehat{u}|$ (bottom) on $[1, 4] \times [-0.32, 0.32]$

2.2.3 The domain $[0.1, 10] \times [-0.32, 0.32]$

For the following pictures we have evaluated (15) on $[0.1, 10] \times [-0.32, 0.32]$, again with 51×51 grid nodes.

Figure 4 depicts the approximation errors $|V - \widetilde{V}|$, $|V - \widehat{V}|$ and $|u - \widehat{u}|$. In addition, Figure 5 shows a comparison between the approximated and exact optimal value functions and control.

Now the error is very pronounced not only in the optimal control variable $\widehat{u}$ and the second order approximation $\widetilde{V}$ but also in the optimal value function $\widehat{V}$ obtained from evaluating $\widehat{u}$ according to (14), increasing up to a value of 6 for $x = (10, -0.32)$.

Note that the error does not evolve symmetrically in all directions. Also, while there is an obvious correlation between the errors in $\widehat{u}$ and $\widetilde{V}$, the relation between the errors in $\widehat{u}$ and $\widehat{V}$ is not as direct as one might expect. Rather, a large error in $\widehat{V}$ is caused only if the trajectory in (14) stays, with high probability and for a long time, in a region with a large error in $\widehat{u}$. Thus, the dynamics of the control system in (14) (which are, in turn, influenced by $\widehat{u}$) play a prominent role here.

In conclusion, the approximation error for the value functions $\widetilde{V}$, $\widehat{V}$ and the control variable $\widehat{u}$ increases with the distance to the equilibrium. Hence, for this procedure the neighbor-

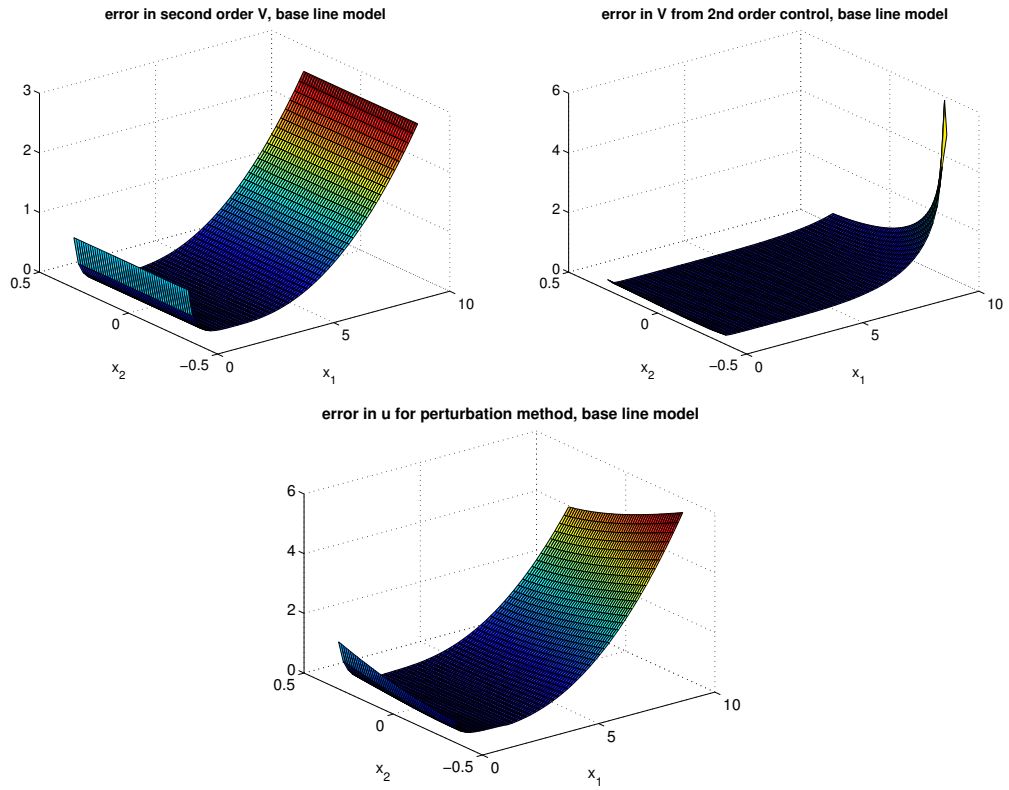

Figure 4: Approximation error $|V - \widetilde{V}|$ (top left), $|V - \widehat{V}|$ (top right) and $|u - \widehat{u}|$ (bottom) on $[0.1, 10] \times [-0.32, 0.32]$

hood to the equilibrium is a necessary condition for accuracy. As shown above, both the approximation for $\widehat{u}$ and for $\widehat{V}$ generate very small errors around the equilibrium, but they are increasing away from it. Thus, the computation of the optimal value function based on second order approximations, as done in Kim and Kim (2003), Benigno and Woodford (2006a,b) and Collard and Juillard (2001), is justified only in a small enough neighborhood of the equilibrium. Furthermore, the more involved dynamic programming type computation of $\widehat{V}$ from $\widehat{u}$, according to (15), shows considerably better results than the straightforward second order approximation $\widetilde{V}$. On the other hand, one should note that while the loss in precision as the system deviates from the equilibrium is an obvious disadvantage of the second order approximation, its big advantage is the high speed of calculation for $\widehat{u}$ and $\widetilde{V}$.

2.3 The Extended Model

Since the size of the stochastic influence σ does not affect the exact solution (3), (4) of our base line model discussed in Section 2.1, the values g_{ss} and h_{ss} of the approximation are almost identical to 0. To increase the role of stochastic dynamics in the model, in this section we investigate an extended model where the objective function will be extended by an additional term including x_2.

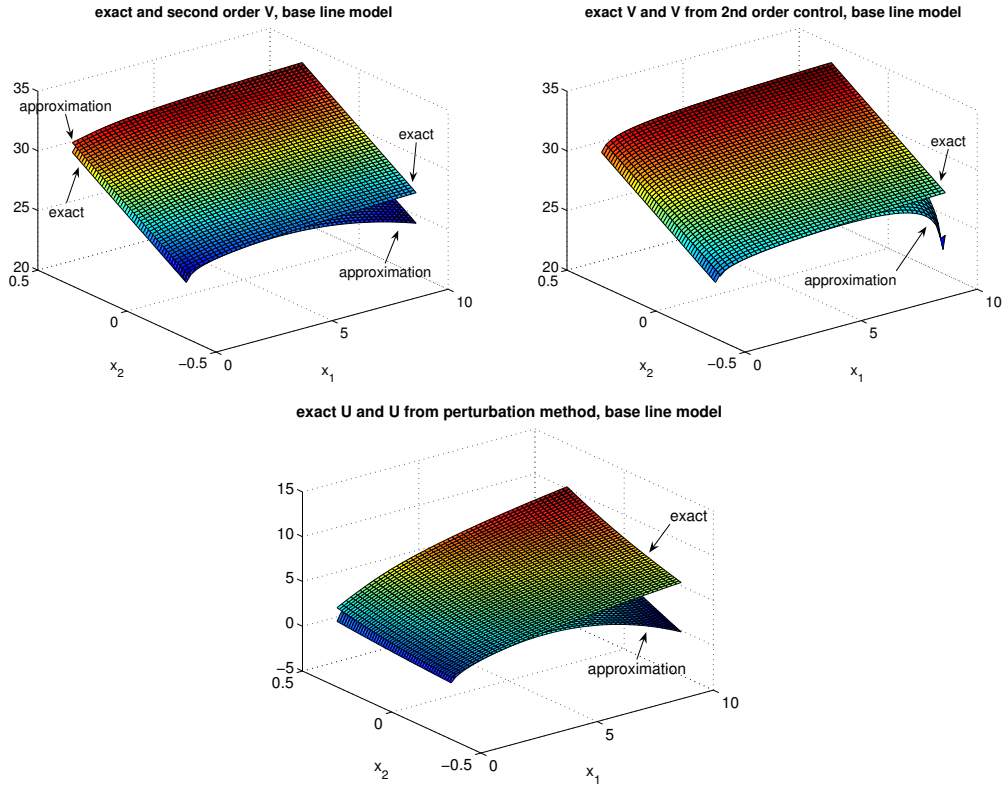

Figure 5: Comparison between V and $\widetilde{V}$ (top left), V and $\widehat{V}$ (top right) and u and $\widehat{u}$ (bottom) on $[0.1, 10] \times [-0.32, 0.32]$

In the extended model setup the objective function is defined as:

$$\max_{u_t} E\left[\sum_{t=0}^{\infty} \beta^t \, \log u_t \cdot e^{\kappa \cdot x_2}\right] \tag{16}$$

and the dynamics remain unchanged:

$$\varphi(x, u, z) \;\; = \;\; \begin{pmatrix} A\, e^{x_2}\, x_1^{\alpha} - u \\ \rho x_2 + z \end{pmatrix}. \tag{17}$$

The parametric specification stays the same as in the simple case, that is: $A = 5$, $\alpha = 0.34$, $\beta = 0.95$, $\rho = 0.9$. Figure 6 shows the optimal value functions, depending on κ, computed by the dynamic programming algorithm as presented in Appendix B.

In what follows we specify the new parameter to $\kappa = 2$. To find the equilibrium condition, again, the optimality conditions have to be defined and solved. Proceeding as in Section 2.1 yields

$$E_t(f(x, x', u, u')) = 0$$

$$\text{for} \quad f(x, x', u, u') := \begin{pmatrix} x_1' \; - A\, e^{x_2}\, x_1^{\alpha} + \; u \\ x_2' - \rho \, x_2 \\ \frac{\beta}{u'} \cdot e^{\kappa x_2'} \cdot \alpha \cdot A \cdot e^{x_2'} \cdot {x_1'}^{(\alpha-1)} - \frac{1}{u} \cdot e^{\kappa x_2} \end{pmatrix}. \tag{18}$$

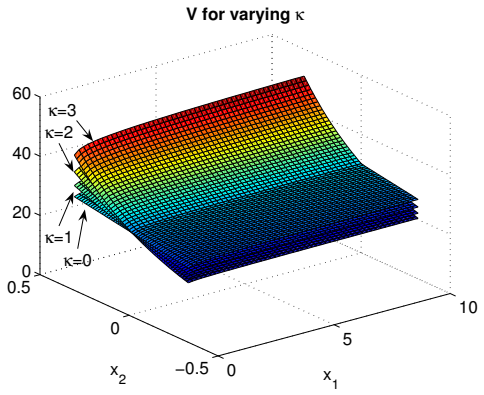

Figure 6: Value function of extended model for $\kappa = 0, 1, 2, 3$

Although the extension of the objective function affects the last row in equ. (18), the equilibrium values for x_1, x_2 and u for perturbation parameter $\sigma = 0$ stay the same. The reason is for this is that in steady state still $\bar{x}_2 = 0$ needs to hold and thus the additional $e^{\kappa x_2}$-terms vanish. Substituting this in the first row of the equilibrium equ. (18) yields the same as equ. (12):

$$u = A \cdot 1 \cdot x_1^{\alpha} - x_1. \tag{19}$$

Like in the previous case, the expression for u can be used to solve for x_1:

$$\begin{aligned} & \beta\, \alpha\, A\, x_1^{(\alpha-1)}\, \frac{e^{\kappa x_2}}{A\, x_1^{\alpha} - x_1} = \frac{e^{\kappa x_2}}{A\, x_1^{\alpha} - x_1} \\ \iff \qquad & \beta\, \alpha\, A\, x_1^{(\alpha-1)} = 1. \end{aligned}$$

This yields the following equilibrium values:

$$\bar{x} = (2.067344815, 0), \quad \bar{u} = 4.333103529$$

Setting $\kappa = 2$, the perturbation methods yields the following coefficients:

- Coefficients of the linear terms:

$$g_{x_1} = 0.7126, \quad g_{x_2} = 4.7277$$

$$h_{x_1} = \begin{pmatrix} 0.3400 \\ 0 \end{pmatrix}, \; h_{x_2} = \begin{pmatrix} 1.6727 \\ 0.9000 \end{pmatrix}$$

- Coefficients of the quadratic x-terms:

$$g_{x_1x_1} = -0.2275, \quad g_{x_1x_2} = g_{x_2x_1} = 0.7775, \quad g_{x_2x_2} = 5.0563$$

$$h_{x_1x_1} = \begin{pmatrix} -0.1085 \\ 0 \end{pmatrix}, \quad h_{x_1x_2} = h_{x_2x_1} = \begin{pmatrix} 0.2751 \\ 0 \end{pmatrix}, \quad h_{x_2x_2} = \begin{pmatrix} 1.3441 \\ 0 \end{pmatrix}$$

- Coefficients of the quadratic σ-terms:

$$g_{\sigma\sigma} = -7.5821, \quad h_{\sigma\sigma} = \begin{pmatrix} 7.5821 \\ 0 \end{pmatrix}.$$

Note that the terms $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ increased considerably compared to the base line model. As expected, this implies that the stochastic dynamics are now more important.

The coefficients for the second order approximation $\widetilde{V}$ of the optimal value function evaluate to

$$V_0 = 29.32566, \;\; V_{x_1} = 0.24291, \;\; V_{x_2} = 28.38888$$

$$V_{x_1x_1} = -0.11749, \;\; V_{x_1x_2} = V_{x_2x_1} = 0.41735, \;\; Vx_2x_2 = 42.71896, \;\; V_{\sigma\sigma} = -0.15508 \cdot 10^{-2}.$$

Figure 7 depicts the approximations $\widetilde{V}$, $\widehat{V}$ and $\widehat{u}$ of the optimal value function and the optimal control on the domain $[1, 4] \times [-0.32, 0.32]$.

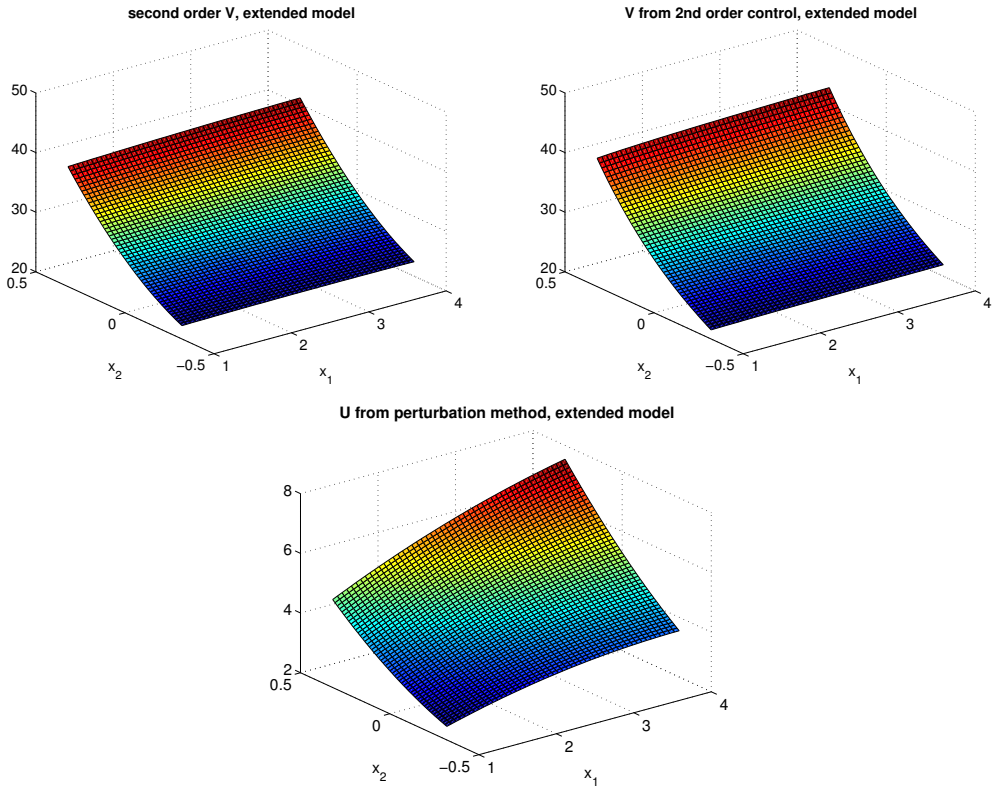

Figure 7: Approximations $\widetilde{V}$ (top left), $\widehat{V}$ (top right) and $\widehat{u}$ (bottom) on $[1, 4] \times [-0.32, 0.32]$

Due to the complexity of the extended model, it is not possible to solve the model analytically. Hence, it is not possible to check for the accuracy of the approximation results. From the previous analysis it can, however, be inferred that the approximation is close to the exact values in a neighborhood of the equilibrium. This is confirmed by the comparison with a dynamic programming method which we will perform in the following section.

3 Comparison with Dynamic Programming

Next we evaluate the results of the previous section by comparing them with the Bellman or Dynamic Programming method. Here we use a dynamic programming method with adaptive grid generation[10], a brief description of the algorithm is given in Appendix B.

3.1 The Base Line Model

Here we will compare the results of the second order approximation with the results using the dynamic programming algorithm. The numerical approximations from the dynamic programming method will be denoted by $\widetilde{V}'$ and $\widetilde{u}'$.

The following Figures 8–10 are the counterparts to the Figures 2–4 for the second order approximation method.

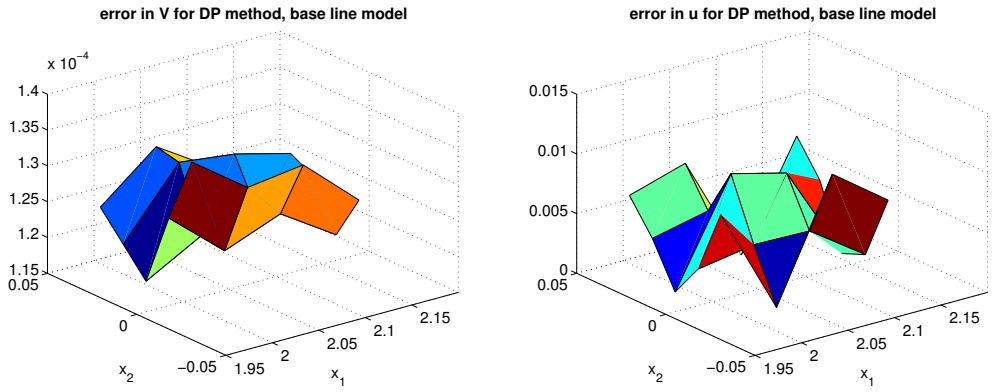

Figure 8: Approximation error $|V-\widehat{V}'|$ (left) and $|u-\widehat{u}'|$ (right) on $[2.0, 2.1]\times[-0.025, 0.025]$

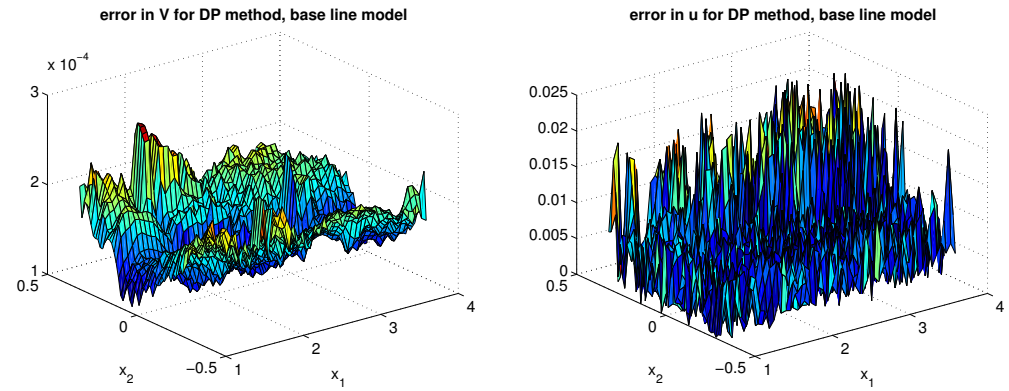

Figure 9: Approximation error $|V - \widehat{V}'|$ (left) and $|u - \widehat{u}'|$ (right) on $[1, 4] \times [-0.32, 0.32]$

[10]See Grüne (2004a,b,c) for a detailed discussion of the algorithm and Grüne and Semmler (2004) for economic applications.

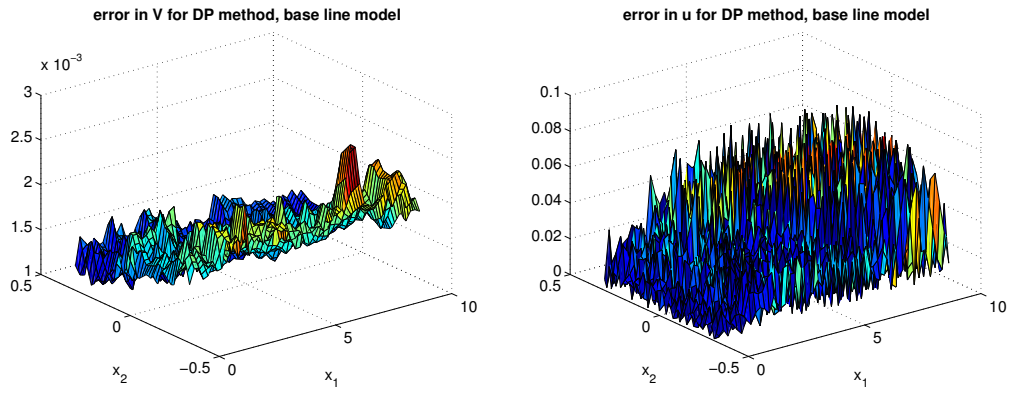

Figure 10: Approximation error $|V - \widehat{V}'|$ (left) and $|u - \widehat{u}'|$ (right) on $[0.1, 10] \times [-0.32, 0.32]$

Being a global method, one would expect the quality of the solution does not vary much with the distance from the equilibrium. On all domains (even on the smallest one), the approximation accuracy for the optimal value function is better than for the second order approximation method. The error in u is considerably larger in a neighborhood of the equilibrium but considerably smaller far away from it.

In order to illustrate the differences between the two approximation methods more accurately, in the following Table 1 the maximum errors of V and u for all methods are compared on different regions. Here the computations for both $\widehat{V}$ and $\widehat{V}'$ were performed on $[1, 4] \times [-0.32, 0.32]$, except for the last line of the table, for which the computations were performed on $[0.1, 10] \times [-0.32, 0.32]$. The error was evaluated on the grid points used for the computation of $\widehat{V}$ by (15), the values for $\widehat{V}'$, which were computed on an adaptive grid, were interpolated onto this grid for the evaluation of the error. In order to remove scaling effects from the results, the table shows the relative errors, i.e., $|(V - \widetilde{V})/V|$, $|(V - \widehat{V})/V|$, $|(u - \widehat{u})/u|$ etc.

The evolution of the maximum approximation error for these ranges confirms the above discussed trends for the approximations: Using the second order perturbation method, the approximation error in $\widetilde{V}$, $\widehat{V}$ and $\widehat{u}$ increases with the domain range. About the equilibrium, the error is almost zero, which can be explained by construction of the method: the equilibrium point $(\bar{x}, \bar{u})$ is given *ex ante* and serves as the development point. The further away from the equilibrium, however, the higher the approximation error. On the intermediate domains our method of computing the value function $\widehat{V}$ from $\widehat{u}$ produces better results than the second order approximation $\widetilde{V}$ while on the two largest domains $\widetilde{V}$ gives better results. On the larger domains both $\widetilde{V}$ and $\widehat{V}$ exhibit errors which are two to three orders of magnitude lower that those in $\widehat{u}$, which in turn reach a maximum relative error > 1 on the largest domain.
Using dynamic programming, the approximation also grows with the domain, however, it starts from a larger value that the perturbation method and increases much more slowly. In particular, it always stays on the order of 10^{-6} for $\widehat{V}'$ and 10^{-3} for $\widehat{u}'$ except for the largest region, this is due to the fact that for this region the solution was computed on a larger computational domain.

relative errors	**second order perturbation**			**dynamic programming**	
Domain	Error of $\widetilde{V}$	Error of $\widehat{V}$	Error of $\widehat{u}$	Error of $\widehat{V}'$	Error of $\widehat{u}'$
in equilibrium	$5.56 \cdot 10^{-7}$	$1.19 \cdot 10^{-4}$	$6.53 \cdot 10^{-11}$	$5.46 \cdot 10^{-6}$	$1.14 \cdot 10^{-3}$
$[2, 2.1] \times [-0.01, 0.01]$	$5.61 \cdot 10^{-7}$	$2.11 \cdot 10^{-5}$	$1.11 \cdot 10^{-6}$	$4.25 \cdot 10^{-6}$	$2.40 \cdot 10^{-3}$
$[1.8, 2.5] \times [-0.05, 0.05]$	$4.61 \cdot 10^{-5}$	$2.28 \cdot 10^{-5}$	$6.93 \cdot 10^{-4}$	$4.87 \cdot 10^{-6}$	$2.63 \cdot 10^{-3}$
$[1.5, 3.5] \times [-0.1, 0.1]$	$1.20 \cdot 10^{-3}$	$3.63 \cdot 10^{-5}$	$1.58 \cdot 10^{-2}$	$7.63 \cdot 10^{-6}$	$4.20 \cdot 10^{-3}$
$[1.5, 3.5] \times [-0.3, 0.3]$	$1.29 \cdot 10^{-3}$	$9.40 \cdot 10^{-5}$	$3.15 \cdot 10^{-2}$	$9.09 \cdot 10^{-6}$	$4.35 \cdot 10^{-3}$
$[1, 4] \times [-0.32, 0.32]$	$3.08 \cdot 10^{-3}$	$2.82 \cdot 10^{-4}$	$6.55 \cdot 10^{-2}$	$9.09 \cdot 10^{-6}$	$4.54 \cdot 10^{-3}$
$[0.1, 10] \times [-0.32, 0.32]$	$9.54 \cdot 10^{-2}$	$2.18 \cdot 10^{-1}$	1.03	$1.01 \cdot 10^{-4}$	$1.55 \cdot 10^{-2}$

Table 1: Comparison of the approximation errors of both methods on increasing domains

Interestingly, except for very small neighborhoods of the origin, we obtain

$$\frac{\text{error in } \widehat{V}'}{\text{error in } \widehat{u}'} \approx \frac{\text{error in } \widehat{V}}{\text{error in } \widehat{u}}$$

Note: the order of computation is different, because in the dynamic programming method $\widehat{u}'$ is computed from the approximation $\widehat{V}'$, while in the perturbation method using (14), $\widehat{V}$ is computed from $\widehat{u}$. Hence, one might expect the latter ratio to be larger due to accumulation of errors. The reason this does not happen lies in the dynamics of the system: those regions where $\widehat{u}$ carries large errors do not contribute much in the formula (14) for $\widehat{V}$ because they are visited with low probability.
Table 1 suggests that the approximation of $\widehat{u}$ using the second order perturbation method is suitable for values in small enough vicinity of the equilibrium, but loses accuracy the further away it departs from the equilibrium. The situation for the optimal value functions $\widetilde{V}$ and $\widehat{V}$ is better but qualitatively the same. In contrast to this, the dynamic programming method yields rather uniformly distributed errors. This method outperforms the perturbation method for the value function on all domains but the smallest and for the control variable on all domains except the two smallest.

3.2 The Extended Model

The model presented in section 2.3 was not analytically solvable. Hence, the results of the perturbation method were not comparable with the true values. In this section the results

of the perturbation and the dynamic programming method will be evaluated.

Figures 11 and 12 compare the respective solutions on two different domains.

Figures 13 and 14 show the respective differences.

Qualitatively, we observe almost the same effect as in the base line model: the difference between the two methods increases as the domain gets larger, which is most likely due to a loss in accuracy in the perturbation method. Quantitatively, the differences in $\widehat{u}$ and $\widehat{u}'$ are comparable to the errors in u in the base line model. The differences in the value functions, however, are larger on the smaller domain and smaller on the larger domain, at least for $\widehat{V}$. A possible explanation for the former could be that the perturbation method is less accurate for the extended model also close to the equilibrium because the solution does now depend on σ, thus an additional approximation error in σ appears. The latter effect might again be due to the fact that the regions where $\hat{u}$ carries a large error are visited with low probability by the trajectory in (14) and hence do not significantly contribute to the value of $\widehat{V}$.

In sum, the extended model strengthens the finding of the simple case, which is that the perturbation method is rather accurate near the equilibrium but less so on larger domains.

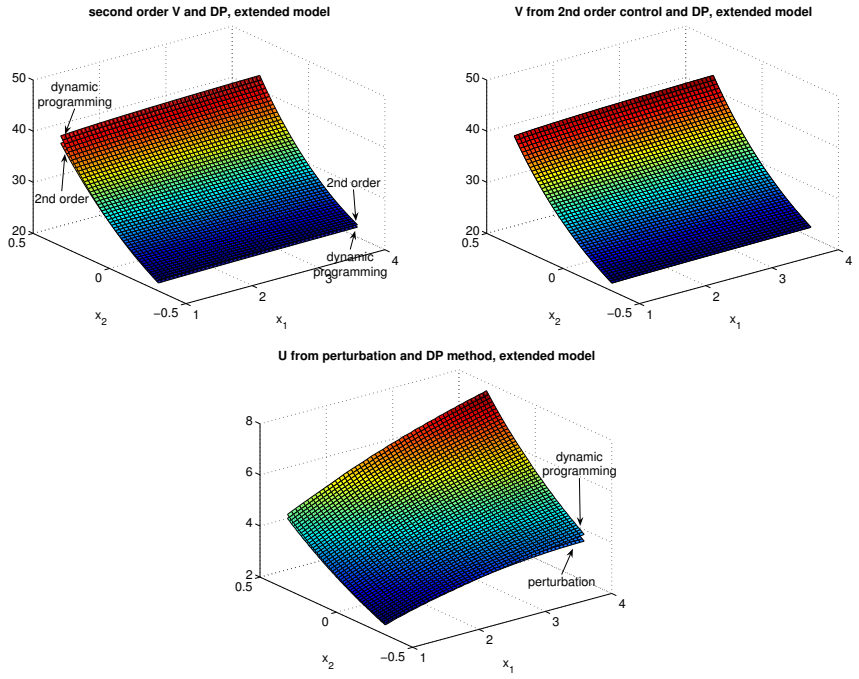

Figure 11: Comparison between V and $\widetilde{V}$ (top left), V and $\widehat{V}$ (top right) and u and $\widehat{u}$ (bottom) on $[1,4] \times [-0.32, 0.32]$

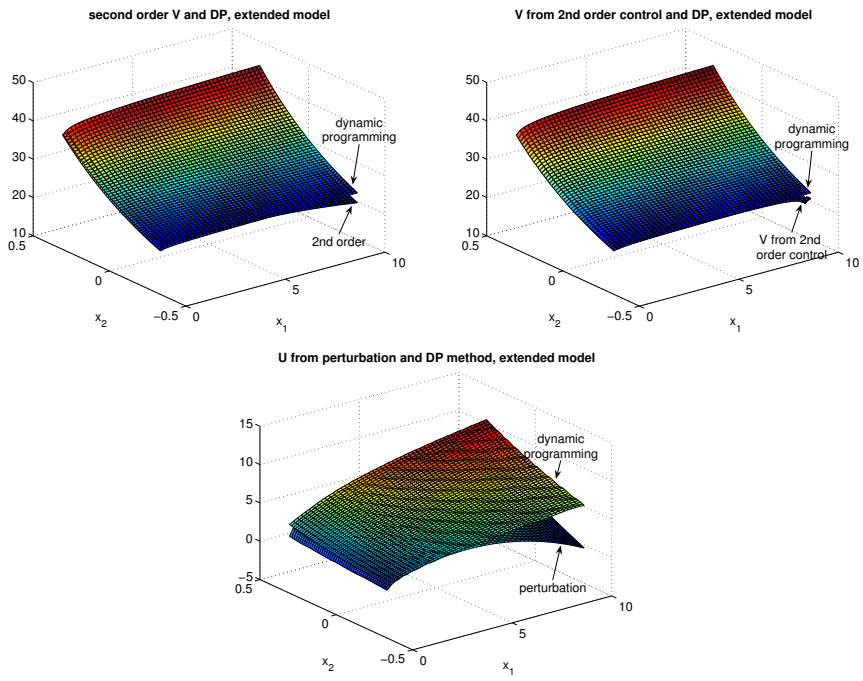

Figure 12: Comparison between V and $\widetilde{V}$ (top left), V and $\widehat{V}$ (top right) and u and $\widehat{u}$ (bottom) on $[0.1,10] \times [-0.32, 0.32]$

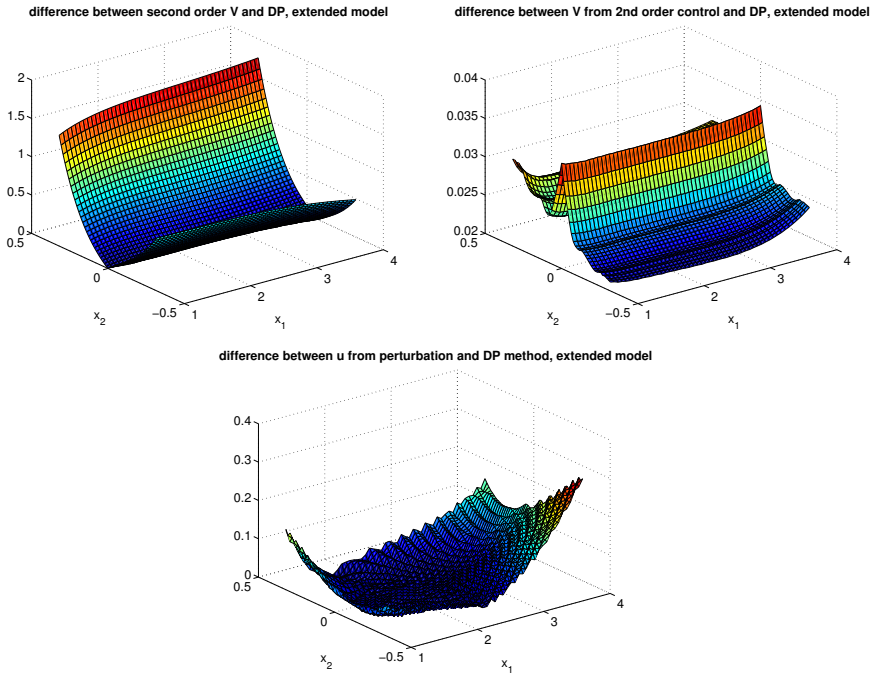

Figure 13: Difference $|\widetilde{V} - \widehat{V}'|$ (top left), $|\widehat{V} - \widehat{V}'|$ (top right) and $|\widehat{u} - \widehat{u}'|$ (bottom) on $[1, 4] \times [-0.32, 0.32]$

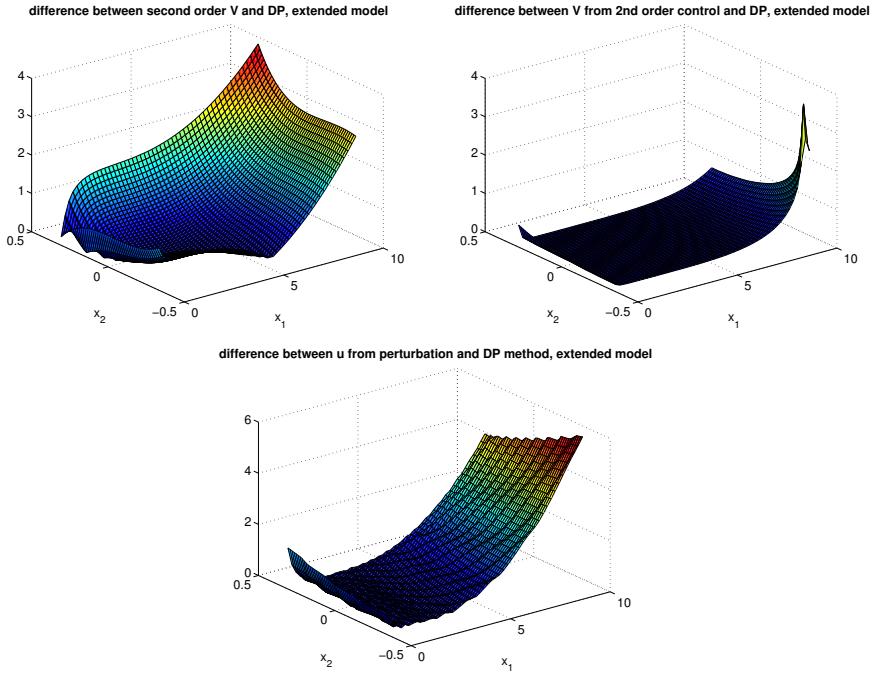

Figure 14: Difference $|\widetilde{V} - \widehat{V}'|$ (top left), $|\widehat{V} - \widehat{V}'|$ (top right) and $|\widehat{u} - \widehat{u}'|$ (bottom) on $[0.1, 10] \times [-0.32, 0.32]$

4 Conclusion

The evaluation of the second order approximation-based perturbation method in sections 2 and 3 showed that the errors in the approximated value functions $\widetilde{V}$ and $\widehat{V}$ and the approximated control variable $\widehat{u}$ increase with the distance to the equilibrium. This can be explained by the truncation error resulting from the Taylor approximation of $\widehat{u}$, or, in other words, by the fact that the approximation $\widehat{u}$ of u is a quadratic function which can only provide a faithful approximation in a region which is small enough such that u itself is approximately quadratic. Thus, as concerning computing welfare through the computation of the optimal value function based on second order approximations as in equ. (13), as done in Kim and Kim (2003), Benigno and Woodford (2006b) and Collard and Juillard (2001), appears to be reasonable only in a small enough neighborhood of the equilibrium.

On the other hand, calculating $\widehat{V}$ from $\widehat{u}$ by (14), our examples reveal that on medium sized domains one can expect a significantly smaller error in $\widehat{V}$ than in the second order approximation $\widetilde{V}$ which only yields useful results in rather small neighborhoods of the origin, however, at the cost that $\widehat{V}$ has to be computed by a dynamic programming type iteration which shares the computational and memory requirements of this method. Thus, from a computational complexity point of view, the computation of $\widehat{V}$ via formula (15) using the second order approximation of the optimal control provides little advantage over the full dynamic programming method which we applied to our problem in Section 3.

A comparison of the two methods in Section 3 showed that the full dynamic programming method inheres a constant approximation error of $\widehat{V}$ which is independent of the equilibrium distance. Although the calculation time of this method is higher, the perturbation method involves — at times complicated — symbolic calculations of the equilibrium equation. These calculations show a potential for improvement. An automatization of this step with programs like MAPLE or MATLAB's symbolic toolbox — which are already successfully used in computing the coefficients from the equilibrium equation — would decrease the time necessary to solve the problem. Yet, the requirement of a solvable equilibrium is a necessary condition and, hence, a further disadvantage of the perturbation method. While the examples presented here were chosen suitably, in general however, solving for the equilibrium might be more complicated or simply impossible.

In summary, for low order problems on large domains, dynamic programming yields, as concerning errors, more stable results for the approximation of the value function $V(x)$ and the control variable u with the errors staying below a certain threshold on the whole domain. While the procedure cannot be applied to higher dimensional problems due to the increasing complexity of the grid of points, it is flexible and well suited for problems of moderate dimension.

The perturbation method does not require the dynamic programming's long computation time. Conditional on the knowledge of the equilibrium conditions, the method yields very accurate values for the approximated control variable $\widehat{u}$ near the equilibrium. The question of what "near the equilibrium" and thus "small enough disturbances" exactly mean in economic terms should be addressed in future research.

Appendices

A MATLAB and MAPLE files

In order to compute the coefficients for the second order approximations $\widehat{u}$, $\widehat{h}$ and $\widetilde{V}$ as well as for the computation of the optimal value function $\widetilde{V}$ based on $\widehat{u}$ we have used MATLAB and MAPLE files which are described in what follows. All files described in the sequel are available from the web page `www.math.uni-bayreuth.de/~lgruene/secondorder.html`.

The computation of the coefficients for $\widehat{u}$ and $\widehat{h}$ was done using the MATLAB files `baseline.m` and `extended.m`, which are based on the routines from Schmitt-Grohe and M. Uribe (2005b). The computation of the coefficients for $\widetilde{V}$ were done in MAPLE using the worksheet `compute_Vcoeff.mw`. All these files do not need any parameters as they only compute the coefficients for the two examples.

All figures as well as the error values in Table 1 were computed with the MATLAB file `comparison.m`. This file computes the exact values u and V from the formulas (4) and (3), the approximations $\widehat{u}$ and $\widetilde{V}$ from the coefficients and the approximation $\widehat{V}$ from $\widehat{u}$ solving (15) by a Jacobi–iteration. The values for $\widehat{V}'$ and $\widehat{u}'$ from the DP method are automatically read from the files `kappa*.asc` where '*' is a file name depending on the domain and model.

The call of this routine is `comparison(kappa,domain,idelta,plotdomain)`. A description of the meaning of the parameters can be found in the comments in the file.

For the figures in this paper we have used the following parameters:

Figure 1:	`comparison(0,1,1e-6,0)`
Figure 2:	`comparison(0,0,1e-6,1)`
Figure 3:	`comparison(0,0,1e-6,0)`
Figure 4:	`comparison(0,1,1e-6,0)`
Figure 5:	`comparison(0,1,1e-6,0)`
Figure 7:	`comparison(2,0,1e-6,0)`
Figure 8:	`comparison(0,0,1e-6,1)`
Figure 9:	`comparison(0,0,1e-6,0)`
Figure 10:	`comparison(0,1,1e-6,0)`
Figure 11:	`comparison(2,0,1e-6,0)`
Figure 12:	`comparison(2,1,1e-6,0)`
Figure 13:	`comparison(2,0,1e-6,0)`
Figure 14:	`comparison(2,1,1e-6,0)`

B Dynamic Programming

In this section we describe a dynamic programming algorithm that has been used in sect. 3 to compute the control variable as well as value function. The basic discretization

procedure has been worked out in Grüne (2004a,b,c) and applied to economic models in Grüne and Semmler (2004). We consider discounted optimal control problems in discrete time $t \in \mathbb{N}_0$ given by

$$V(x) = \max_{u \in \mathcal{U}} \sum_{t=0}^{\infty} \beta^t g(x(t), u(t)) \tag{20}$$

where

$$x_{t+1} = f(x(t), u(t)), \;\; x(0) = x_0 \in \mathbb{R}^n$$

For the discretization in space we consider a grid Γ covering the computational domain of interest. Denoting the nodes of the grid Γ by x^i, $i = 1, \ldots, P$, we are now looking for an approximation V_h^Γ satisfying

$$V_h^\Gamma(x^i) = T_h(V_h^\Gamma)(x^i) \tag{21}$$

for all nodes x^i of the grid, where the value of V_h^Γ for points x which are not grid points (these are needed for the evaluation of T_h) is determined by linear interpolation. For a description of several iterative methods for the solution of (21) we refer the reader to Grüne and Semmler (2004).

For the estimation of the gridding error we estimate the residual of the operator T_h with respect to V_h^Γ, i.e., the difference between $V_h^\Gamma(x)$ and $T_h(V_h^\Gamma)(x)$ for points x which are not nodes of the grid. Thus, for each cell C_l of the grid Γ we compute

$$\eta_l := \max_{x \in C_l} \left| T_h(V_h^\Gamma)(x) - V_h^\Gamma(x) \right|$$

Using these estimates we can iteratively construct adaptive grids as follows:

(0) Pick an initial grid Γ_0 and set $i = 0$. Fix a refinement parameter $\theta \in (0, 1)$ and a tolerance $tol > 0$.

(1) Compute the solution $V_h^{\Gamma_i}$ on Γ_i

(2) Evaluate the error estimates η_l. If $\eta_l < tol$ for all l then stop

(3) Refine all cells C_j with $\eta_j \geq \theta \max_l \eta_l$, set $i = i + 1$ and go to (1).

For more information about this adaptive gridding procedure and a comparison with other adaptive dynamic programming approaches we refer to Grüne and Semmler (2004) and Bauer, Grüne and Semmler (2006).

In order to determine equilibria and approximately optimal trajectories we need an approximately optimal policy, which in our discretization can be obtained in feedback form $u^*(x)$ for the discrete time approximation using the following procedure:

For each x in the grid-domain we choose $u^*(x)$ such that the equality

$$\max_{u \in U} \{hg(x,u) + \beta V_h(x_h(1))\} = \{hg(x,u^*(x)) + \beta V_h(x_h^*(1))\}$$

holds, where $x_h^*(1) = x + hf(x,u^*(x))$. Then the resulting sequence $u_i^* = u^*(x_h(i))$ is an approximately optimal policy and the related piecewise constant control function is approximately optimal.

References

Aruoba, B., J. Fernandez-Villaverde, and J.F. Rubio-Ramirez (2004), "Comparing solution methods for dynamic equilibrium economies", mimeo, University of Pennsylvania.

Aström, K.J. (1970) "Introduction to Stochastic Control Theory", in R. Bellmann, Hrsg., Mathematics in Science and Engineering, Volume 70, Academic Press.

Bauer,F. L. Grüne and W. Semmler (2006) "Adaptive Spline Interpolations for Hamilton-Jacobi-Bellman Equations", Applied Numerical Method, 56:1196-1210.

Becker, S. (2005) "Numerische Lösungen Dynamischer Gleichgewichtsmodelle", Diploma Thesis, University of Bayreuth.

Bellman, R. (1967) "Introduction to the Mathematical Theory of Control Processes", Volume I, in R. Bellman, Hrsg., Mathematics in Science and Engineering, Volume 40, Academic Press.

Benigno, P. and M. Woodford (2006a) "Optimal taxation in an RBC model: A linear quadratic approach", Journal of Economic Dynamics and Control, 30 (9-10): 1445-1485.

Benigno, P. and M. Woodford (2006b) "Linear quadratic approximation of optimal policy problems", mimeo, Colombia University.

Christiano, L. and J. Fisher (2000), "Algorithms for solving models with occasionally binding constraints", Journal of Econmic Dynamics and Control, 24:1179-1231.

Collard, F. and M. Juillard (2001) "Accuracy of Stochastic Perturbation Method: The Case of Asset Pricing Models", Journal of Economic Dynamics and Control, 25: 979-999.

Den Haan, W.J. and A. Marcet (1994), Accuracy in simulations", The Review of Economics Studies, 61:3-17.

Gong, G. and W. Semmler (2006), "Stochastic Dynamic Macroeconomics: Theory and Empirical Evidence", Oxford, New York, Oxford University Press.

Grüne, L. (2004a) "Numerik dynamischer Systeme", Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Wintersemester 2003/2004.

GRÜNE, L. (2004b) "Numerische Dynamik von Kontrollsystemen", Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Sommersemester.

GRÜNE, L. (2004c) "Error estimation and adaptive discretization for the discrete stochastic Hamilton-Jacobi-Bellmann equation", Numerische Mathematik 99, pp.85-112, Springer-Verlag.

GRÜNE, L. AND W. SEMMLER (2004) "Using dynamic programming with adaptive grid scheme for optimal control problems in economics", Journal of Economic Dynamics and Control 28, pp. 2427-2456, 2004.

JIN, H. AND K. L. JUDD (2005) "Perturbation methods for general dynamic stochastic models", Working Paper, Stanford University, `http://bucky.stanford.edu/defaultmain.htm`.

JUDD, K.L. (1996) "Approximation, perturbation, and projection methods in economic analyis", in H. Amman, D. Kendrick, and J. Rust, Hrsg., Handbook of Computational Economics, Amsterdam: Elsevier.

JUDD, K. L. (1998) "Numerical Methods in Economics", Cambridge, The MIT Press.

KIM, J. AND S.H. KIM (2006), "Two pitfalls of linearization methods, mimeo, Tufts University.

KIM, J. AND S.H. KIM (2003), "Spurious Welfare Reversals in International Business Cycle Models", Journal of International Economics, 60: 471-500.

KYDLAND, F.E. AND E. C. PRESCOTT (1982) "Time to build and aggregate fluctations", Econometrica 50, pp. 1345-1370.

MURDOCK, J.A. (1991) "Perturbations, Theory and Methods", John Wiley & Sons.

SCHMITT-GROHE, S. AND M. URIBE (2004a) "Solving dynamic general equilibrium models using a second-order approximation to the policy function", Journal of Economic Dynamics and Control 28, pp. 755-775.

SCHMITT-GROHE, S. AND M. URIBE (2004b), "Optimal fiscal and monetary policy under sticky prices", Journal of Economic Theory, 114: 198-230.

SCHMITT-GROHE, S. (2005a) "Perturbation Methods for the Numerical Analysis of DSGE Models", Lecture Notes, preliminary draft, Duke University, `www.econ.duke.edu/~grohe/teaching/perturbationmethods. htm`.

Schmitt-Grohe, S. and M. Uribe (2005b) "Matlab Codes for Second Order Approximation",
http://www.econ.duke.edu/~uribe/2nd_order.htm.

Smetts, F. and R. Wouters (2004), "Comparing shocks and frictions in US and EURO-Area business cycles", ECB Working paper series, no. 391.

Taylor, J. and H. Uhlig (1990), "Solving nonliear stochastic growth models: A comparison of alternative solution methods", Journal of Business and Economic Statistics 8: 1-18.