Dynamic Perturbation[*]

Alessandro Mennuni, Serhiy Stepanchuk
University of Southampton
alessandro.mennuni@soton.ac.uk, sstepanchuk@gmail.com

December 15, 2018

Abstract

We develop a new algorithm to solve large scale dynamic stochastic general equilibrium models over a large transition. The method consists of Taylor expanding the equilibrium conditions of the model not just around the steady state, but along the entire equilibrium path. The method can be applied to a broad class of models and is orders of magnitudes more accurate than solutions based on local perturbation of the steady state. The method is also able to solve models with kinks and strong nonlinearities. Finally, because our policies are locally linear, we can make use of a version of the Kalman filter with time varying coefficients to identify shocks from data. With this tool in hand we are able to evaluate the likelihood and use the algorithm for estimation of nonlinear models.

JEL Classification Codes: E32, C63
Keywords: Transition, Business Cycle, Computational Economics

[*] We thank Martin Gervais, Paul Klein, Yang Lu.

1 Introduction

This paper develops an algorithm that can be used to compute and estimate large scale and highly nonlinear dynamic stochastic general equilibrium models (DSGE).

The most common algorithm to solve and estimate DSGE models is a perturbation method based on approximating the model around the steady state. This method is so popular because it is fast, can handle models with many state variables, and produces linear policies which can be combined with the Kalman filter (KF) for model estimation. The Achilles heel of this method, however, is accuracy! On the other hand, global methods can deliver accurate approximation to the model solution, since they approximate the model not only around a steady state, but over a large part of the whole state space. However, this accuracy comes at a cost, as global methods tend to be computationally intensive. In particular:

i. they face the curse of dimensionality i.e. they can only solve models with a limited number of state variables;

ii. they are slow which makes them less useful for estimation, where a model has to be solved thousands of times.

The reason why global methods are accurate but slow is that they solve the model all over the sate space. Instead perturbation only approximates around one point. These are two extremes. The method that we develop lies in between. In particular, we extend perturbation methods and apply local approximations repeatedly along an equilibrium path. As a result, our solution is much more accurate than that found with standard perturbation methods and can be applied to a broad class of DSGE models. A second major advantage is that the solution is approximated by a sequence of linear functions. This enables us to use the solution produced by our algorithm as input to model estimation based on maximum likelihood and bayesian methods. In particular, that the policy functions are approximated by locally linear functions allow us to use Kalman filtering techniques in the estimation rather than non linear filtering techniques such as the particle filter, which are computationally more demanding and less developed.

So the three key advantages are the ability to accurately solve models that are large, non linear, and that we are able to use this method to estimate models. To highlight these key advantages we choose a number of applications. To deal with he curse of dimensionality we sole a multicountry model and compare to Maliar & Maliar (2015). To deal with nonlinearities we solve a model with kinks. We also use the latter model int the estimation.

Our first application is to apply to a multicountry model. This is a standard model to test algorithmans that can handle large models because it is easily scalable by choosing the number of countries.[1] For instance Krueger et al. (2011) and Brumm & Scheidegger (2017) use multi-country models to illustrate sparse grid methods with which the number of grid points increases much more slowly in the number of state variables than with standard tensor product grids. This way it is possible to handle much larger models than what possible with more standard global methods: Krueger et al. (2011) solve for up to 50 countries (100 state variables). An alternative approach that is more related to ours uses simulation methods to restrict attention to regions of the state space visited in equilibrium. This way Maliar & Maliar (2015) solve a model with 200 countries. We compare our method with the latter. First, we can easily handle 200 countries. Second, with a standard model calibration we are more accurate when far away from the steady state and comparable to their high order case when close to the steady state. In fact our error does not increase when we move far away from the steady state. One issue, however, is that our algorithm, abstracts from the Jensen inequality; as a result our accuracy is a negative function of the size of the shocks.[2] To isolate the impact of the Jensen inequality we consider two extreme examples. One with very small shocks —essentially a deterministic model— and one with very large ones (about 14 times larger than typical model calibration). In the small shocks scenario our solution always produces substantially lower numerical errors. In the large shocks case, MM can achieve slightly better results with high order polinomial basis. However, this approach is only feasible with a smaller number of countries. Finally, an important difference with simulation methods is that our algorithm does not involve a fix point problem so we do not have convergence issues.

The model above is fairly log linear, as a result it does not serve as a useful framework to appreciate the ability to handle strong nonlinearities. To this aim, we consider a model with financial frictions and occasionally binding constraints as in Mendoza (2010). This is an especially challenging framework because the policy functions have kinks in the region of the state space where the constraint is activated.[3] We get fairly accurate solutions.

[1] see JEDC special issue.

[2] This is because we use locally linear function, however, it would be possible to extend the method to include higher order local approximations.

[3] Currently this is one of the 2 nonlinear setups that are most popular in the DSGE

In fact, our errors are smaller than the errors made in a smooth neoclassical model with standard perturbation methods.

The model above is also an ideal framework to illustrate the estimation techniques because it cannot be dealt with standard linear estimation techniques, and it is very interesting given the interest in financial crises.

The problem with estimating nonlinear models is how to assess the likelihood given some parameter values. There are 2 problems. One is to solve the model which is time consuming. The second one is to then back out shocks that make the model simumalted data consistent with the true ones. This latter is usually done with the KF which however is a linear filter, so it cannot be combined with non linear methods and more elaborate and time consuming methods must be used, making the problem intractable. However, because our policies are locally linear, we can make use of a version of the KF with time varying coefficients found in the engineering literature. With this tool in hand we are able to evaluate the likelihood. Finally, since our algorithm is fast, we can use standard methods to maximize the likelihood or draw from the posterior. To make sure that the filter actually works in practice we test it by backing out shocks from artificially simulated data. We find that we recover them perfectly.

2 Dynamic perturbation: the algorithm

In this section, we describe our numerical algorithm. We start with an intuitive outline of the main ideas behind the algorithm in section 2.1, and then provide its more detailed step-by-step description in section 2.2.

2.1 The algorithm: the outline of the main ideas

DSGE models, especially those with a large number of state variables, are typically solved using perturbation methods. These methods are based on

literature. The other one is the New Keynesian model with a zero lower bound. We chose this application because the zero lower bound one has an additional challenge — the existence of multiple steady states. While it is possible to extend our algorithm to the case of multiple steady states, the description of the algorithm would become more cumbersome. Instead, the occasionally binding constraint application allows us to streamline the presentation of the key elements of our method. In addition, the typical macro model has a unique steady state so writing the algorithm for the case of multiple steady states would have made it less readily applicable.

(log-)linear, or higher-order Taylor series approximation to the system of equilibrium conditions of the model around a fixed approximation point, usually its deterministic steady state. Close to the point of approximation, this solution method is usually quite accurate (see Caldara et al. (2012)). However, the quality of approximation can deteriorate substantially for the values of the state variables far away from the fixed approximation point, especially if the model exhibits large non-linearities. As a result, the standard perturbation methods can provide inaccurate solutions when the researcher wants to study the transitional dynamics after policy, demographic, or technological changes, or in response to large shocks. Similarly, a wide range of models that have recently become of interest to the economists, such as the models with occasionally binding constraints or the models with the zero lower bound for the nominal interest rate, may lead to policy functions that exhibit kinks, and thus are not easily amenable to the standard perturbation methods. We propose a numerical method that aims to deliver an accurate solution in these challenging settings. In essence, it repeatedly applies local approximations over the entire transition path, between the initial point and the steady state (the long-run solution) of the model. Local approximation of the model around a given point on a transition path allows us to obtain an accurate approximation to the model's policy and transition functions around that point. Combining these functions with a particular realization of the shocks, we obtain the values of the state variables at the next point along the transition, where the whole process is repeated. As a result, we obtain an approximate solution to the transition path, with a uniform degree of precision along the whole path. In addition, we obtain a sequence of local linear approximations to the policy and transition functions, which we can use as inputs into a modified Kalman filter that allows for time-varying coefficients, and use it to evaluate the model-implied likelihood function[4].

Before we provide more details, we need to introduce some notation. As in Schmitt-Grohe & Uribe (2004), we consider a dynamic general equilibrium macroeconomic model that can be formulated as a system of equilibrium conditions:

$$E_t[f\left(x_{t+1}, y_{t+1}, x_t, y_t\right)] = 0, \tag{1}$$

where E_t is the expectation conditional on the information at time t, x_t is a vector of size n_x of the "current-period" realizations of the predetermined (or

[4] More on this in section 5

state) variables, y_t is a vector of size n_y of the "current-period" realizations of the non-predetermined (or *control*) variables of the model, while x_{t+1} and y_{t+1} are the corresponding "next-period" realizations of these variables. The state vector x_t can be partitioned into $x_t = [x_t^1, x_t^2]$, where x_t^1 consists of endogenous state variables, while x_t^2 follows some exogenous stochastic process. In all our applications, we will assume that x_t^2 follows a stationary VAR(1) process:

$$x_{t+1}^2 = \Lambda x_t^2 + \sigma \tilde{\eta} \varepsilon_{t+1}$$

where ε_t is a vector of shocks (of size $n_\varepsilon = n_{x^2}$) that have zero mean and variance matrix I, and $\tilde{\eta}$ is an $n_\varepsilon \times n_\varepsilon$ matrix[5].

The vector-valued function f typically combines first-order static and dynamic optimality conditions that characterize optimal choices of economic agents populating the model, market-clearing conditions and the equations that characterize the laws of motion for the endogenous and exogenous state variables. It consists of $n = n_x + n_y$ possibly non-linear equations.

We assume that the true model solution can be represented recursively, as a policy function that maps state variables into the control variables:

$$y_t = g(x_t, \sigma) \tag{2}$$

and the transition function that maps current values of the state variables (and possibly realizations of the shocks) into the next-period values of the state variables:

$$x_{t+1} = h(x_t, \sigma) + \sigma \eta \varepsilon_{t+1} \tag{3}$$

where $\eta = \begin{bmatrix} 0 \\ \tilde{\eta} \end{bmatrix}$.

In the exposition of our method, the following notation will be useful. We will use $\hat{g}_x$ to denote a linear approximation of g around x, and similarly use $\hat{h}_x$ to denote a linear approximation of h around x. Since in this paper, we restrict our attention to locally linear approximations, they will have the certainty equivalence property (see Schmitt-Grohe & Uribe (2004)). Our approach can be extended to using higher-order local approximations to policy functions, but we live this to future work.

Suppose we know the initial values of the state variables, x_0, and the sequence of realized shocks, $\{\varepsilon_t\}_{t=1}^T$, and want to find the corresponding path

[5] In all our applications, we consider models with stationary fundamentals. However, our algorithm can be easily applied in environments with non-stationary elements.

for state and control variables that solve the model. A starting point of the standard perturbation method is finding a (log-)linear, or higher order Taylor approximation of the deterministic version of the equilibrium system of equations 1:

$$f\left(x_{t+1}, y_{t+1}, x_t, y_t\right) = 0. \tag{4}$$

around the steady state, where $x_t = x_{t+1} = \bar{x}$ and $y_t = y_{t+1} = \bar{y}$ and

$$f(\bar{x}, \bar{y}, \bar{x}, \bar{y}) = 0.$$

The key insight in our method is that one can construct a Taylor series approximation to the system of equilibrium conditions at any "dynamic" point that satisfies this system, not just the steady state. Unfortunately, finding such a point where $f\left(x_{t+1}, y_{t+1}, x_t, y_t\right) = 0$ can be a challenging task. Because of the dynamic links in the model, one needs to take into account the whole future path of the model's variables in order to pin down their current values. One way to see the nature of the problem is to note that the $n = n_x + n_y$ equations in f do not allow us to solve for the $n_x + n_y + n_y$ values of (x_{t+1}, y_t, y_{t+1}) (the n_x values of x_t are predetermined and known in period t). Intuitively, the choice of y_{t+1} has to be consistent with equilibrium behavior starting with time $t + 1$, and so on.

Using the steady state as a point of approximation circumvents this problem, since by definition, the values of the variables will remain constant over time. However, as we have explained above, using the steady state as a point approximation may lead to poor approximation quality in many cases of interest.

Therefore, the key step in our algorithm is to be able to start with any "current-period" values of the state vector x_t (potentially far from the steady state), and find the corresponding values of y_t, x_{t+1} and y_{t+1} such that the vector $(x_{t+1}, y_{t+1}, x_t, y_t)$ satisfies the system of equilibrium conditions in (1) (and is consistent with the rest of some equilibrium path for state and control variables). We will call this problem "finding local dynamic solution" (FLDS). Once this point is found, we can use it to derive an approximation to the policy functions around it, and use them together with the realized values of innovations ε_{t+1} to obtain the values of the state and control variables in the following period.

To solve FLDS, we start by constructing an auxiliary deterministic path between the "current-period" state x_t and the steady state $\bar{x}$, and then trace it backwards. To construct this auxiliary path, we use the policy functions approximated around the steady state. Setting all the innovation values to 0, we apply the steady-state policy functions and build a deterministic path from x_t to $\bar{x}$. We stop when this path gets sufficiently close to the steady state $\bar{x}$, so this procedure generates a finite auxiliary path $\{x_t, \widetilde{x}_{t+1}, \ldots, \widetilde{x}_T\}$ such that $\widetilde{x}_T$ is close to $\bar{x}$.[6]

Next, we move backwards along this auxiliary path until we reach x_t, at which point, FLDS will be solved. We start at $\widetilde{x}_T$, and assume that in period $T+1$, the values of control variables can be determined using the policy functions approximated around the steady state, $y_{T+1} = \hat{g}_{\bar{x}}(x_{T+1})$. Now one can use $n_x + n_y$ equations in f to solve for $\widetilde{y}_T$ and $\widetilde{x}_{T+1}$ (and thus also find $\widetilde{y}_{T+1} = \hat{g}_{\bar{x}}(\widetilde{x}_{T+1})$) such that:

$$f\left(\widetilde{x}_{T+1}, \widetilde{y}_{T+1}, \widetilde{x}_T, \widetilde{y}_T\right) = 0.$$

Next, we can construct a Taylor approximation of the equilibrium system f around this solution, and use it to obtain the approximation to the policy around $\widetilde{x}_T$, $\hat{g}_{\widetilde{x}_T}$[7].

After this, we move backwards to the previous point in the auxiliary path, $\widetilde{x}_{\overline{T}-1}$, replace $\hat{g}_{\bar{x}}$ with $\hat{g}_{\widetilde{x}_T}$ (in other words, we assume that y_T can be found from $\underline{\ }y_T = \hat{g}_{\widetilde{x}_T}(x_T)$), and repeat the previous steps: (i) first, we find $\widetilde{y}_{T-1}$ and $\widetilde{\widetilde{x}}_T$ (and the implied $\widetilde{\widetilde{y}}_T = g_{\widetilde{x}_T}(\widetilde{\widetilde{x}}_T)$) such that[8]:

$$f\left(\widetilde{\widetilde{x}}_T, \widetilde{\widetilde{y}}_T, \widetilde{x}_{T-1}, \widetilde{y}_{T-1}\right) = 0.$$

and then we use a Taylor approximation of f around $(\widetilde{\widetilde{x}}_T, \widetilde{\widetilde{y}}_T, \widetilde{x}_{T-1}, \widetilde{y}_{T-1})$ to find an approximation to the policy function around $\widetilde{x}_{T-1}$.

These steps can be repeated until we reach x_t. At that point, FLDS problem is solved: we have $(\widetilde{\widetilde{x}}_{t+1}, \widetilde{\widetilde{y}}_{t+1}, x_t, \widetilde{y}_t)$ such that:

$$f\left(\widetilde{\widetilde{x}}_{t+1}, \widetilde{\widetilde{y}}_{t+1}, x_t, \widetilde{y}_t\right) = 0.$$

We can then obtain an approximation to the policy and transition functions at x_t. Combining them with the realization of the shocks ε_{t+1}, we get y_t and x_{t+1}, and repeat the above procedure starting with x_{t+1} as the initial point. This can be repeated to produce a simulated equilibrium path of any desired length.

[6] As it will be clear later, this path need not be a true equilibrium path. In principle, one could also just take a linear path between x_t to $\bar{x}$.

[7] The implementation of step is somewhat different from the standard approach as in Blanchard & Kahn (1980), and we provide more details in the appendix.

[8] Note that $\widetilde{\widetilde{x}}_T$ need not coincide with $\widetilde{x}_T$, and similarly $\widetilde{\widetilde{x}}_T$ need not coincide with $\widetilde{y}_T$

2.2 The algorithm details

In this section, we provide a more detailed step-by-step description of our numerical algorithm. Some of the more technical steps are described in detail in appendix.

Suppose we have the initial values of the state variables, x_0, and the sequence of realized shocks, $\{\varepsilon_t\}_0^{\bar{T}}$. We want to obtain the equilibrium path (of length $\bar{T}+1$) for state and control variables of the model.

1. Apply the Taylor expansion to the system of equations in (1) around the deterministic steady state, $(\bar{x}, \bar{y}, \bar{x}, \bar{y})$, where $\bar{x}$ and $\bar{y}$ are the deterministic steady state values of the state and control variables respectively, and obtain a linear approximation to the policy functions $\hat{h}_{\bar{x}}(x)$, $\hat{g}_{\bar{x}}(x)$. If these are stable, then go to the next step (for stability, see for instance Blanchard & Kahn (1980)).[9]

2. Put $t = 0$ (index t is used to denote the element in our simulated equilibrium path).

   The next 2 steps draw the auxiliary path from the initial condition to the deterministic steady state

3. Set $\widetilde{h}(\cdot) = \hat{h}_{\bar{x}}(\cdot)$ and $\widetilde{g}(\cdot) = \hat{g}_{\bar{x}}(\cdot)$

4. Set the shocks to 0. Start from x_t and, using the steady state policy function $\hat{h}_{\bar{x}}$, generate a sequence $\{\widetilde{x}_\tau\}_t^T$ with $T > \bar{T}$.[10] If this sequence does not converge to the steady state, increase T and repeat this step.

   The following steps trace the auxiliary path backwards from the steady state to the current initial point, x_t, and compute the next point in our solution sequence.

5. Set $\tau = T$

[9] This algorithm is described for models that are stable around the steady state. In fact, it could be extended to models that do not have a steady state provided that a point (x', y', x, y) such that $f(x', y', x, y) = 0$ is known. Indeed, it has worked for models that are locally unstable in some regions of the state space.

[10] Since $\sigma = 0$, this simulation is independent of any time series for the innovations to the shocks. It provides a path along which to move backward from the steady state.

6. Set $x = \widetilde{x}_\tau$.

7. Find x' and y such that $f\left(x', \widetilde{g}(x'), x, y\right) = 0$ (note that we substituted y' with $\widetilde{g}(x')$).[11]

8. Derive linear approximations to the policy functions, $\hat{h}_x$, $\hat{g}_x$, using the Taylor expansion of $f\left(x', y', x, y\right) = 0$, around the local solution point (x', y', x, y) (with $y' = \widetilde{g}(x')$) found in the previous step. Our implementation of this step has some novel features that are detailed in appendix.

9. Update $\widetilde{h}(.)$ and $\widetilde{g}(.)$ to $\widetilde{h}(\cdot) = \hat{g}_x(\cdot)$ and $\widetilde{g}(\cdot) = \hat{g}_x(\cdot)$.

10. If $\tau > t$, set $\tau = \tau$ and go back to step 6.

11. If $\tau = t$, we have found a local solution to $f(x', y', x, y) = 0$ with $x = x_t$. This solves our FLDS problem. We can use this point to construct a Taylor approximation to the equilibrium system of equations (1), and obtain local approximation to the policy functions, $\hat{h}_{x_t}(x, \sigma)$ and $\hat{g}_{x_t}(x, \sigma)$[12]. We use these policy functions and the realizations of the shocks, ε_{t+1}, to obtain and store the next values of state and control variables in our model simulation, $x_{t+1} = \hat{h}_{x_t}(x_t, \sigma) + \sigma\eta\varepsilon_{t+1}$, $y_t = \hat{g}_{x_t}(x_t, \sigma)$, and then go to the next step.

12. If $t = \bar{T}$, the whole solution has been found! Otherwise, set $t = t + 1$ and go back to step 3.

Variations of this algorithm can be conceived; for instance, to increase speed one could avoid going backward through all the points on the equilibrium path, but make larger jumps from the steady state until x_0. On the other hand, if one is concerned with capturing high degree of non-linearity of policy functions, one can break the backward step from x_t to x_{t-1} into several substeps by linearly interpolating between the 2 points.

[11] This step is similar to a step in the policy function iteration algorithm. Here, this step makes sure that the function $f\left(x', y', x, y\right) = 0$ holds and hence a Taylor expansion is admissible.

[12] In this step, one can obtain either a linear or a higher order approximation. In this version of the paper, we limit ourselves to linear approximations.

3 Dealing with high-dimensinal models: multi-country RBC model

In this section, we evaluate our solution method by comparing its performance to other popular numerical approaches. As our test laboratory for these comparisons, we use a multi-country real business cycle model. This model has been widely used for comparing the performance of different solution methods (see, for instance, Kollman et al. (2011)).

In the next section, we provide a brief description of the model. Next, we use a special case of this model with only 1 country and full capital depreciation, and compare our solution results to those obtained from perturbation solution around the deterministic steady state, and to the analytical solution (which is available in this special case). After that, we use a more general model setup, and compare our results to those obtained using the method developed in Maliar & Maliar (2015), which is specifically designed to provide a globally accurate solution to problems with a large number of state variables.

3.1 Model setup

In this section we describe the model that we use in our comparison exercise. It is similar to one of the models analysed in Maliar & Maliar (2015). The advantage of this model is that one can easily increase the dimensionality of the problem. This will allow us to study how a given numerical algorithm performs in a problem with large number of state variables.

Model Setup: there are N countries, each populated by an infinitely-lived representative agent. They consume a single consumption good produced simultaneously in each of the N countries. The representative agent in each country has the same time separable expected utility function. In particular, we assume that the per-period utility function is logarithmic:

$$u_i(c_i) = u(c_i) = \log(c_i), \quad i = 1, \dots N.$$

Output in each country is produced using only capital, which is the only factor of production. All countries have the same Cobb-Douglas production function, and differ only in the quantity of capital employed and realized value of the multiplicative productivity shock:

$$f_i(k_i) = Aa_i k_i^{\alpha}, \quad i = 1, \dots N$$

where a_i is the value of the productivity shock in country i and A is a normalizing constant chosen so that $k_i = 1$ in a deterministic steady state. The logarithm of a_i follows an AR(1) process:

$$\log a_{i,t+1} = \rho \log a_{i,t} + \varepsilon_{i,t}$$

where ρ is the autocorrelation coefficient, and $\varepsilon_{i,t} \sim N(0, \sigma^2)$. We assume that $\varepsilon_{i,t}$ are uncorrelated across countries.

In this frictionless economy, one can obtained the solution by solving the corresponding social planner's problem:

$$\max_{\{c_{i,t},k_{i,t+1}\}_{t=0,\dots,\infty}^{i=1,\dots,N}} E_0 \sum_{i=1}^{N} \lambda_i \left[\sum_{t=}^{\infty} \beta^t u(c_{i,t}) \right]$$

$$\text{s.t.} \quad \sum_{i=1}^{N} c_{i,t} + \sum_{i=1}^{N} k_{i,t+1} = \sum_{i=1}^{N} k_{i,t}(1-\delta) + \sum_{i=1}^{N} A a_{i,t} f(k_{i,t})$$

for some given $\{k_{i,0}, a_{i,0}\}_{i=1,\dots N}$. λ_i is the planner's weight assigned to each country i.

The solution must satisfy N Euler equations:

$$1 = \beta E_t \left[\frac{u'(c_{i,t+1}}{u'(c_{i,t})} (1 - \delta + A a_{i,t} f'(k_{i,t+1}) \right] \tag{5}$$

3.2 One Country and Full Capital Depreciation

To evaluate this algorithm, we test it on the one-country version of model with full depreciation, for which the analytical solution is known. We then compare the true equilibrium path $\{x_t^*, y_t^*\}_0^T$ with the one generated by this algorithm, $\{x_t^{**}, y_t^{**}\}_0^T$, and with the one generated by a second-order expansion around the steady state $\{x_t^{***}, y_t^{***}\}_0^T$. For an initial condition quite far from the steady state, $x_0 = [.2k_{ss}, -.5]$, with variance of the shock equal to 0.007, [13] the maximum error

$$\max_t [\max\left(|x_t^* - x_t^{***}|, |y_t^* - y_t^{***}|\right)] \tag{6}$$

[13] This is the typical calibration of a TFP shock in the RBC model. The other parameters are $\theta = .33$, $\rho = .99$ and $\beta = .99$.

using second-order approximation around the steady state is 0.0077. Using the proposed algorithm, the maximum error

$$\max_t[\max\left(|x_t^* - x_t^{**}|, |y_t^* - y_t^{**}|\right)]$$

is 2.2610^{-5}, which is 340 times smaller than taking the expansion only around the steady state. The simulation computed with the two methods is compared with the true solution in Figure 7. We conclude that this method makes a notable improvement in terms of accuracy versus perturbation around the steady state.[14]

In this example the code takes 27 seconds to run a simulation of 60 periods on a laptop. As a measure of accuracy, we compute the error

$$|f\left(E_t(x_{t+1}), E_t(y_{t+1}), x_t, y_t\right)| \tag{7}$$

for all t, i.e. the residuals from the equilibrium conditions when $t+1$ variables are at their "expected" levels (abstracting from Jensen's inequality). Abstracting from Jensen's inequality, a solution to the model is such that the error (7) is equal to 0 for all t. Hence, the size of this error gives a sense of the accuracy of the approximated solution. The maximum error with local approximation of the steady state is 0.29; with this algorithm it is 1.710^{-12}.

3.3 Two Countries

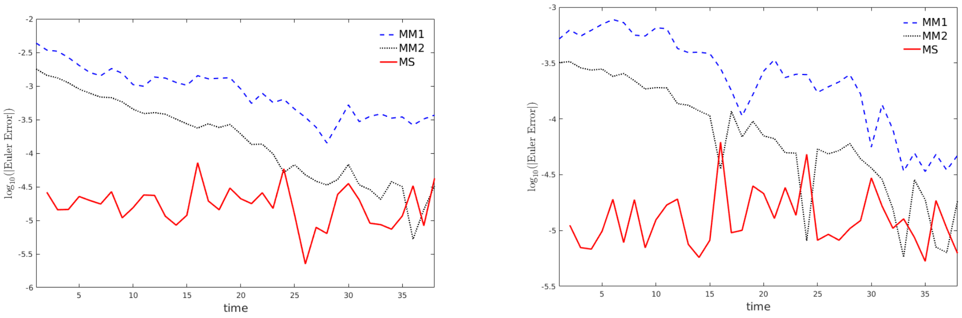
Next, we move to the case with $N = 2$ countries (and $\delta < 1$). Obtaining a numerical solution in this case is a fairly simple task, as the total number of state variables, $2N = 4$, is low. However, even this simple setup allows us to demonstrate some advantages of our solution method, both in terms of accuracy and speed.

To obtain the approximations to the capital and consumption policy functions, we solve the model with 3 numerical solution methods: (1) our solution method described above in section ... (we label the results that correspond to using our solution method as MS in the graphs and tables below); (3) the method of Maliar & Maliar (2015) using only the first-order polynomials as basis functions (which we label MM1); (3) the method of Maliar &

[14] Furthermore, the accuracy seems robust to initial errors. In fact, using a small $\bar{T}$ such that the initial path does not converge to the steady state and the backward procedure starts with an initial error, has a negligible effect on accuracy. Intuitively, step 7 corrects for such errors.

Maliar (2015) using both first-order and second-order polynomials as the basis functions (MM2)[15]. Using these approximations, we obtain the simulated paths of length $T = 40$ for capital and consumption in both countries. To assess the accuracy of the solution, we look at the implied errors in the Euler equations expressed in consumption units. To highlight the advantages of our numerical method, we start the simulation with the capital in both countries far away from the corresponding steady state values. To generate the sequence of productivity shocks in both countries for the simulated path, we set their standard deviation to $\sigma_i = 0.01$, which is similar to the values usually adopted by the literature in this type of models[16].

Figure 1: Euler equation errors, 2-country model



(a) $k_0 = 0.5k_{ss}$

(b) $k_0 = 1.5k_{ss}$

Figure 1 compares Euler equation errors along a simulated path for consumers in country 1[17]. In the left panel, we show the Euler errors from the simulated path where we start the simulation with capital in both countries

[15] We do not use higher order polynomials with the Maliar method, since in our experience this becomes impractical as we increase the number of countries: either the time to convergence becomes prohibitively long, or the method fails to converge altogether.

[16] To obtain the realized values of capital and consumption along the simulated path, we use the same realizations of productivity shocks that we use to obtain the numerical solution with our method. To approximate the expectational terms in the Euler equations, we use the monomial integration rules with $2N^2 + 1$ integration nodes, as described in Maliar & Maliar (2015)

[17] Recall that we assume that all countries are identical except for the actual realizations of productivity shocks, which may be different in each country. However, in a Pareto efficient solution, all consumers will have identical consumption that does not depend on the country-specific productivity realizations.

Table 1: Solution Time

Solution Method	CPU time
MM1	20.5 sec
MM2	58.4 sec
MS	10.7 sec

50 percent below the steady state value, while in the right panel, we start the simulated path with capital in both countries 50 percent above the steady state value. In both cases, the errors from the MS solution are uniformly lower than those from the MM1 solution along the whole simulated path. The errors from the MM2 solution are higher than those from the MS solution during the initial part of the simulated path, when the capital levels are still far away from their respective steady state values. The two set of errors become similar as the capital levels approach their steady states. It is also worth noting that the quality of the approximation of the MS solution stays uniform along the whole simulated path, independently of whether the values of the state variables are close to the steady state or not, while for the solutions obtained using Maliar method, the quality deteriorates further away from the steady state.

Table 1 shows that our algorithm is also noticeably faster than that of Maliar & Maliar (2015)[18].

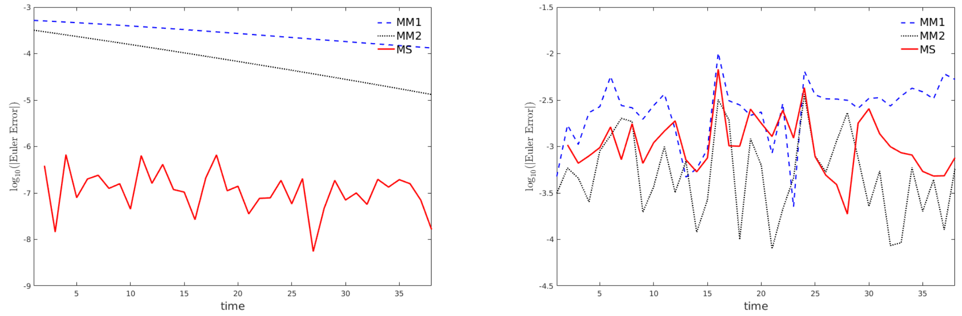
3.4 Changing volatility of shocks

Our numerical method is based on local first-order approximations to the equilibrium system of equations, and thus implies certainty equivalence (see Schmitt-Grohe & Uribe (2004) for the details). This means that our method does not capture the impact of the size of the shocks on the policy functions. This is in contrast to the Maliar & Maliar (2015) global solution method. As a result, we can expect that the performance of our method, compared to that of Maliar & Maliar (2015), will improve when we reduce the size of the shocks, and will deteriorate as we increase the size of the shocks.

2 confirms our expectations. It shows the size of the Euler equation errors for the 2-country version of the model, assuming 2 different magnitudes of

[18] For Maliar algorithm in the model with $N = 2$ countries, we approximated the expectations using the monomial integration rules with $2N^2 + 1$ integration nodes

Figure 2: Euler equation errors, changing the volatility of shocks



(a) $\sigma = 10^{-6}$

(b) $\sigma = 0.1$

the size of the shocks. Subplot (a) demonstrates the results for the case when $\sigma = 10^{-6}$. The graph shows that our algorithm delivers a much more precise solution compared to Maliar & Maliar (2015) algorithm. In this case, the error due to the certainty equivalence assumption is essentially eliminated. This allows us to highlight the advantages of our algorithm in term of the accuracy along the transition path. It is worth to note again that, unlike the results from Maliars algorithm, the errors generated by our solution method do not become larger further away from the steady state.

Subplot (b) shows the case with $\sigma = 0.1$. Similarly to the previous case, this is a rather extreme parametrization (with the size of the shocks about 10 times larger than usual calibrations in the literature), which we use only to illustrate certain advantages and drawbacks of the two numerical algorithms. In this case, the solution from our algorithm is still slightly more precise than Maliar & Maliar (2015) solution that uses only the first-order polynomials, and slightly less precise than Maliar & Maliar (2015) solution that uses the second-order polynomials[19].

3.5 Changing the number of countries

The advantages of our algorithm really come to light when we increase the number of the countries, and thus the dimensionality of the problem (recall

[19] We expect that the ability of our algorithm to capture the impact of the size of the shocks on the solution would improve substantially after extending it to utilizing the second-order approximations. However, we leave this extension for the future.

Table 2: Accuracy and speed in multi-country model

	N=20			N=40			N=200		
Soln Method	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
MM1	-4.27	-3.01	188.18	-4.28	-2.98	244.70	-4.29	-2.94	792.27
MM2	-4.85	-3.41	1399.21	-4.94	-3.48	12105.21			
MS	-5.43	-4.43	24.31	-5.42	-4.53	58.01	-5.42	-4.70	390.92

that the number of state variables in our model is equal to $2N$, where N is the number of countries). It is worth to note that the algorithm developed by Maliar & Maliar (2015) is considered to be at the cutting edge of the numerical algorithms intended to solve the problems with the large number of state variables.

Table 2 shows the approximation errors (L_1 denotes the average errors across the Euler equations in all N countries and the resource constraint; L_∞ denotes the corresponding maximum errors) and computing time in seconds (CPU) from the different numerical solutions. We followed Maliar & Maliar (2015) and set the following parameters of their algorithm to be the same as those reported in their Table 3: for the case of $N = 20$ countries, we set the target number of points in the EDS grid to $\bar{M} = 1000$, and used monomial integration rules with $2N$ integration nodes; for the case of $N = 40$ countries, we set $\bar{M} = 4000$ and used a one-node Gauss-Hermite integration rule; finally, for $N = 200$ case, we used $\bar{M} = 1000$ and one-node Gauss-Hermite integration rule. We found the choice of the integration rule to be critical for the running time of Maliars algorithm. For a large number of countries ($N \geq 40$), the one-node Gauss-Hermite integration rule appears to be the only viable option. Similarly, for a large N using polynomials of order higher than 1 slows the algorithm down substantially. However, intuitively, using a one-node integration node eliminates the ability of Maliars algorithm to capture the impact of the size of the shocks on the solution – the advantage of their algorithm over ours that we have discussed above. Unfortunately, we did not manage to achieve convergence with their algorithm using one-integration node rule for the case of large shocks (σ=0.1) to illustrate this point.

Figure **??** shows the Euler errors along the whole simulation paths from MM1 and MS solutions for the case of $N = 200$ countries:

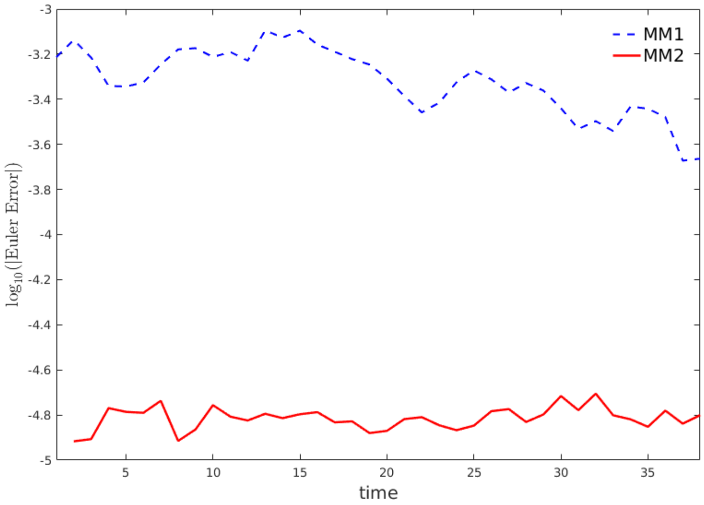


Figure 3

4 Dealing with models with large non-linearities: a model of sudden stops

The model that we use in this section is based on Mendoza (2010). One of its main characteristic is the presence of the occasionally binding borrowing constraint. This feature makes it similar to many recent papers that analyze the impact of financial imperfections on the economy. At the same time, it can introduce large non-linearities to the model's policy functions, which makes it an ideal testing ground for our algorithm.

4.1 Model setup

A representative consumer in a small open economy maximizes:

$$E_0\left(\sum_{t=0}^{\infty}\exp\left(-\sum_{\tau=0}^{t}\rho\left(\bar{c}_\tau - N(\bar{L}_\tau)\right)\right)u\left(c_t - N(L_t)\right)\right)$$

subject to a sequence of the following period budget constraints:

$$(1+\tau_c)c_t + i_t + q_t^b b_{t+1} = \exp(\varepsilon_t^A)F(k_t, L_t) - \phi(R_t - 1)w_t L_t + b_t$$

where

$$i_t = \delta k_t + (k_{t+1} - k_t)\left(1 + \Psi\left(\frac{k_{t+1} - k_t}{k_t}\right)\right)$$

Note that $\rho(.)$ is an increasing and concave "endogenous discount rate" function. It solves the problem of continuum of deterministic steady states in a small open economy model. $\bar{c}$ and $\bar{L}$ denote the aggregate consumption and labor supply which consumer takes as given (in equilibrium, $\bar{c} = c$ and $\bar{L} = L$).

Output is produced using a constant-returns-to-scale technology that requires capital (k_t) and labor (l_t) as inputs. ε_t^A is a TFP shock. ϕ is a fraction of the cost of labor that is paid in advance of sales with an intra-period working capital loan. International lenders charge the world interest rate $R_t = R\exp(\varepsilon_t^R)$ on both the intra- and inter-period loans, where ε_t^R is the interest rate shock.

Additionally, domestic consumer faces the following collateral constraint:

$$q_t^b b_{t+1} - \phi R_t w_t L_t \geq -\kappa q_t k_{t+1}$$

where $q_t^b = 1/R_t$.

The functional forms of preferences and technology are as follows:

$$\begin{aligned}
u(c_t - N(L_t)) &= \frac{\left(c_t - \frac{L_t^\omega}{\omega}\right)^{1-\sigma} - 1}{1-\sigma}, \quad \sigma, \omega > 1,\\
\rho(c_t - N(L_t)) &= \gamma \log\left(1 + c_t - \frac{L_t^\omega}{\omega}\right), \quad 0 < \gamma \leq \sigma,\\
F(k_t, L_t) &= Ak_t^{1-\alpha}L_t^\alpha, \quad 0 < \alpha < 1, A > 0,\\
\Psi\left(\frac{z_t}{k_t}\right) &= \frac{a}{2}\left(\frac{z_t}{k_t}\right), \quad a \geq 0.
\end{aligned}$$

The equilibrium system of equations is provided in the appendix.

4.2 Solution

To incorporate the occasionally binding constraints in our numerical algorithm, we follow Judd et al. (2000) and introduce the penalty function of the form:

$$K \cdot \max\left(-(\kappa q_t k_{t+1} + q_t^b b_{t+1} - \phi R_t w_t L_t)^d, 0\right)$$

This penalty is activated only when the borrowing limit is violated. By choosing a large $K > 0$ (and $d \in \{2, 4\}$), we discourage the representative consumer from violating the borrowing limit. Alternatively, the occasionally binding constraint can be incorporated into our analysis using the methods from Garcia & Zangwill (1981).

We set the parameter values similar to those in Mendoza (2010), draw a sequence of shock realizations for ϵ_t^A and ϵ_t^R, and use our algorithm to generate an equilibrium path for the state and control variables. Figure 4 shows the result. As one can see from the "Borrowing limit" part of the figure, our solution generates 3 "sudden stop" episodes when the borrowing limit is binding.

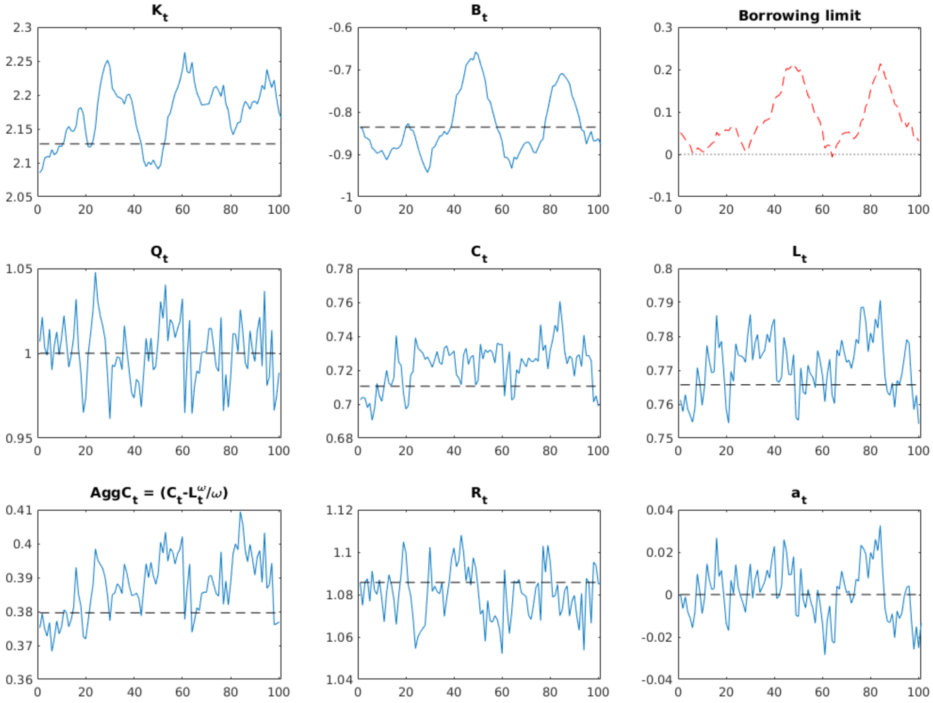


Figure 4: Simulated Equilibrium Path from the Sudden Stops Model

5 Model estimation with Kalman filter

5.1 Idea

For a given set of parameter values and shock realizations, our algorithm produces a path of equilibrium state and control variables, and a corresponding sequence of local linear approximations to the policy and transition functions. We can use these local linear approximations to the policy and transition functions as inputs for the generalized version of the Kalman filter where the Kalman filter model coefficients are allowed to vary over time. Since the true policy and transition functions are non-linear, we will use the *extended* Kalman filter, where these functions are replaced with their linear approximations (Jacobians).

Using the recursive representation of the true solution of the model in (2) and (3) and expanding the vector of states x_t and controls y_t as needed, we can without any loss of generality assume that the vector of observables, z_t, is a subset of y_t, and formulate the non-linear state-space model as follows[20]:

$$\begin{aligned} x_{t+1} &= f^{kf}(x_t) + \eta^{kf}_{t+1} \\ z_t &= h^{kf}(x_t) + \epsilon^{kf}_t \end{aligned}$$

The first equation above is the non-linear state transition equation, while the second one is the non-linear measurement equation. ϵ^{kf}_t is the vector of measurement errors. One can replace f^{kf} and h^{kf} with their linear approximations (Jacobians) evaluated at the most recent predicted values of the state vector to formulate the extended Kalman filter model:

$$\begin{aligned} x_{t+1} &= F^{kf}_{\hat{x}_{t|t}}(x_t) + \eta^{kf}_{t+1} \\ z_t &= H^{kf}_{\hat{x}_{t|t-1}}(x_t) + \epsilon^{kf}_t \end{aligned}$$

and use $F^{kf}_{\hat{x}_{t|t}}$ together with $H^{kf}_{\hat{x}_{t|t-1}}$ in the usual Kalman gain and covariance-updating formulas.

5.2 Applying Kalman filter in the Sudden Stops model

We assume that we can observe the following 5 variables: (1) output growth (measured as $\log(Y_t) - \log(Y_{t-1})$), (2) investment to output ratio $\left(\frac{I_t}{Y_t}\right)$, (3)

[20] The superscript "kf" is used to denote that we are working towards formulating a Kalman filter model

current account to output ratio $\left(\frac{CA_t}{Y_t}\right)$, (4) real interest rate (R_t), (5) hours worked (L_t). Since we only have 2 structural shocks in our version of the Sudden Stops model (ε_t^A and ε_t^R), we need to introduce measurement errors. We assume that the first 3 observables are measured with error, while the real interest rate and hours worked are measured without an error[21]. As our exercise, we keep the model parameters the same as in section 4.2, draw a particular sequence of shock realizations and use our algorithm to generate an equilibrium path of length T[22]. We use this as artificial data, and then apply the extended Kalman filter described above to recover the realizations of the shocks. We have discovered that a straightforward application of the extended Kalman filter does not produce very accurate results. The intuition is that in a highly non-linear model, the values of the Jacobians $F_{\hat{x}}$ and $H_{\hat{x}}$ that approximate the true transition and measurement equations could be sensitive to the point of approximation. To improve the accuracy of the Kalman filter, we apply the *iterative extended* Kalman filter. The idea here is to iterate on the measurement equation, updating both H and the best state estimate. The details of the algorithm are provided in Havlik & Straka (2015), who argue that iterated Kalman filter can be viewed as an application of the Gauss-Newton method that generates the *maximum a posteriory* (MAP) estimate of the state vector. The resulting match between the artificial data and the Kalman filter-generated sequences of observables is shown in figure 5, and for some other variables (including the two structural shocks) in figure 6.

In figures 8 and 9 in the appendix we show the results from the straightforward application of the extended Kalman filter, without iterating on the measurement equation.

[21] We found that as long as the size of these measurement errors are kept small, it makes little difference in which of the equations they appear.

[22] For this exercise, we use $T = 100$

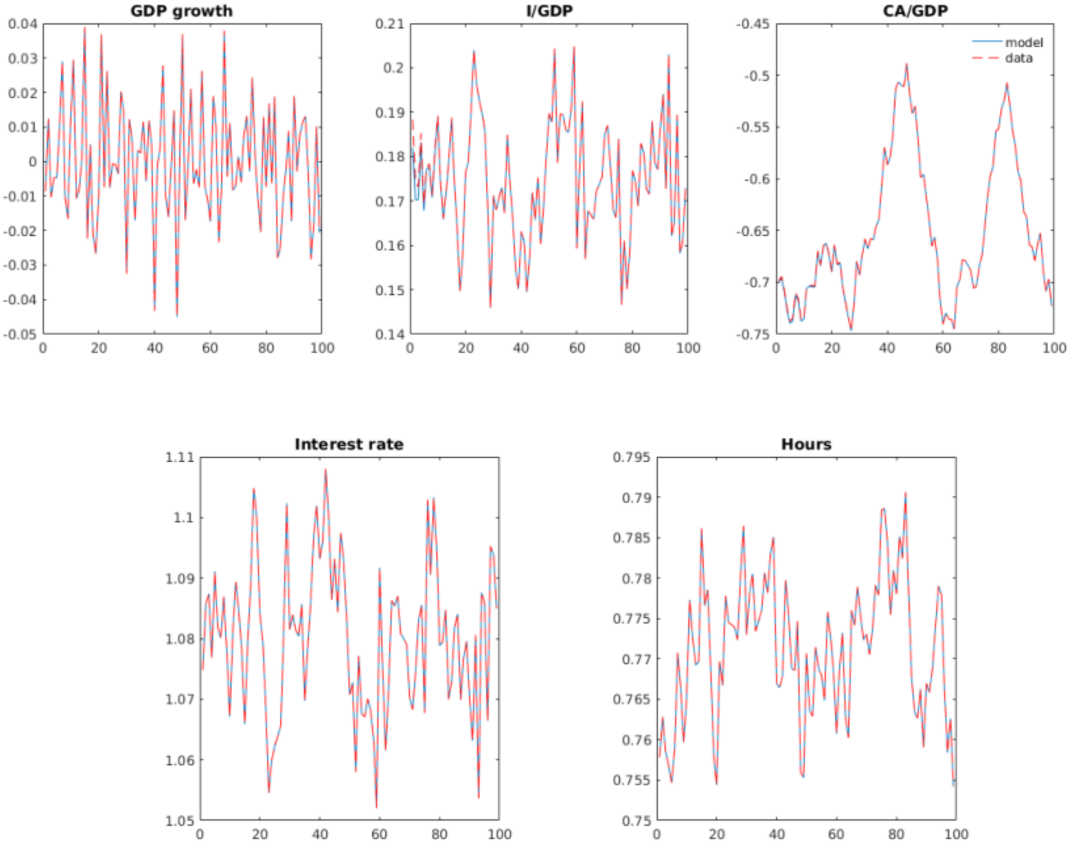


Figure 5: Comparison between artificial data and Kalman filter-generated measurement variables

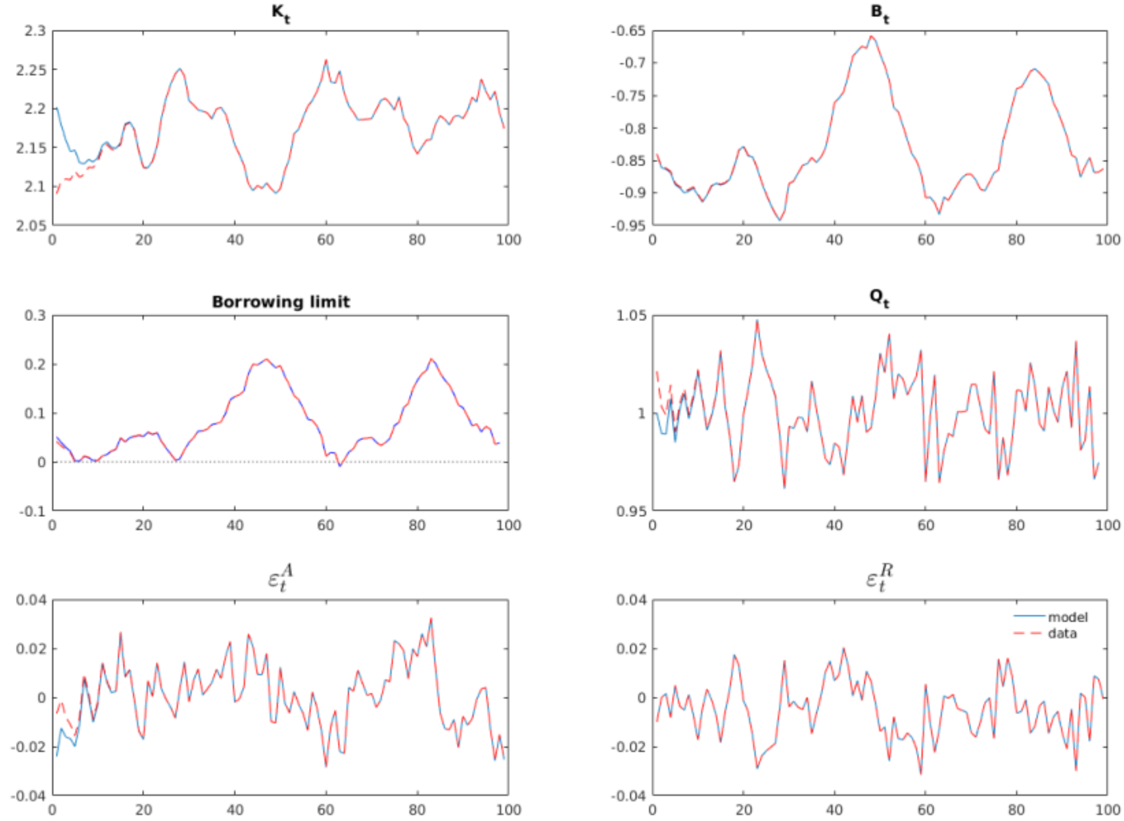


Figure 6: Comparison between other artificial data and Kalman filter-generated variables

6 Figures

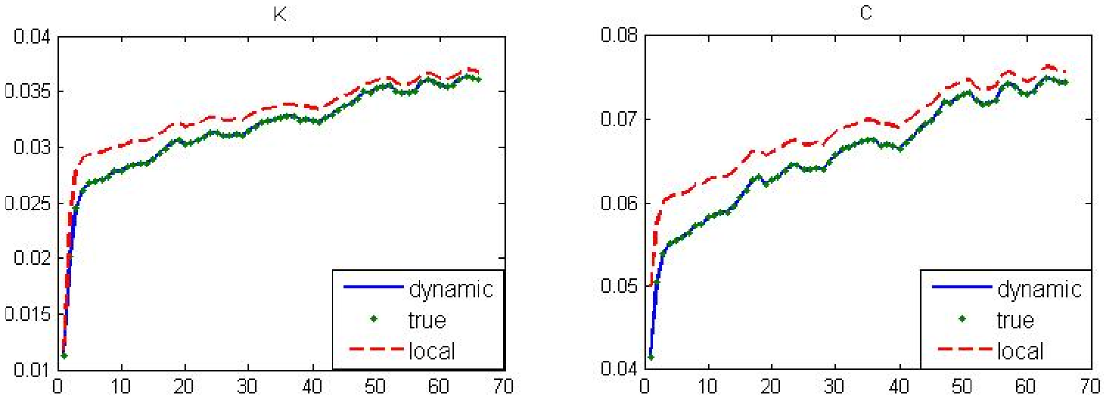


Figure 7: Capital and consumption from the analytical example of Appendix 2 compared with the solution computed with local and dynamic perturbation.

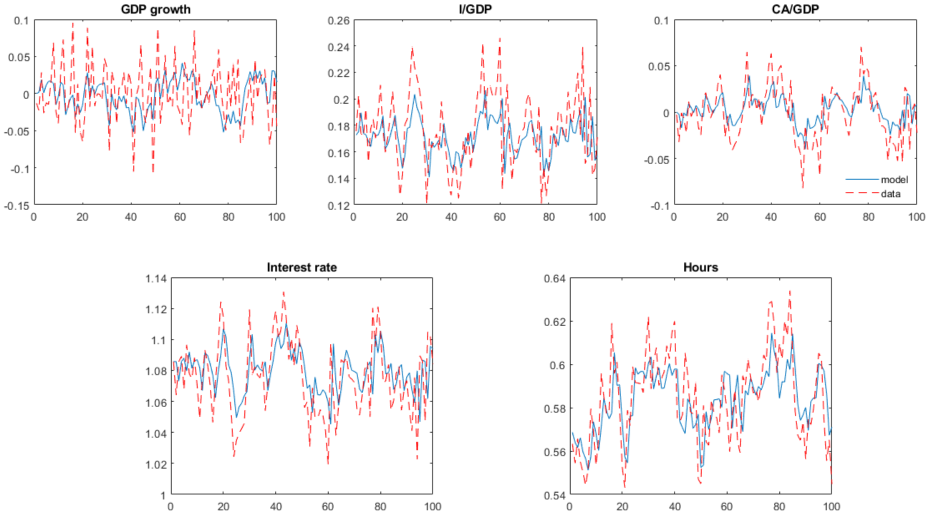


Figure 8: Comparison between artificial data and Kalman filter-generated measurement variables (without iteration on the measurement equation)

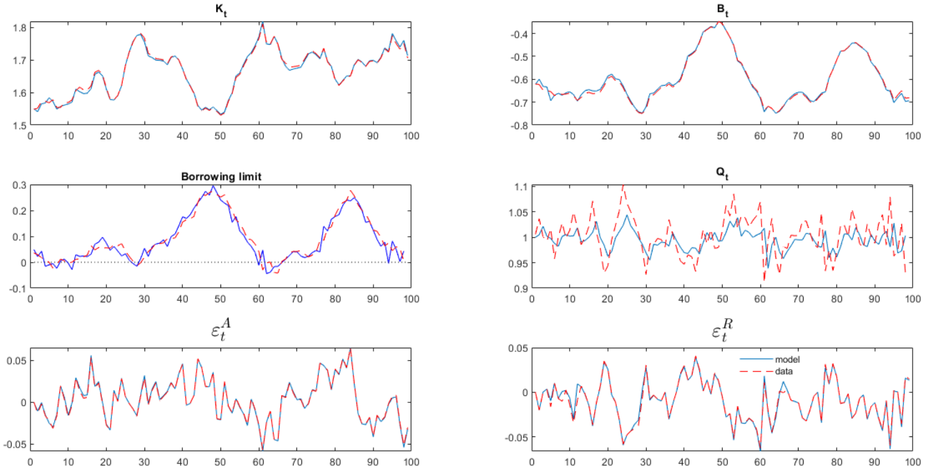


Figure 9: Comparison between other artificial data and Kalman filter-generated variables (without iteration on the measurement equation)

References

Blanchard, O. & Kahn, C. (1980), 'The solution of linear difference models under rational expectations', *Econometrica* **48**(5), 1305–11.

Brumm, J. & Scheidegger, S. (2017), 'Using adaptive sparse grids to solve highdimensional dynamic models', *Econometrica* **85**.

Caldara, D., Fernandez-Villaverde, J., Rubio-Ramirez, J. & Yao, W. (2012), 'Computing DSGE models with recursive preferences and stochastic volatility', *Review of Economic Dynamics* **15**(2), 188–206.

Garcia, C. & Zangwill, W. (1981), *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ.

Gomme, P. & Klein, P. (2011), 'Second-order approximation of dynamic models without the use of tensors', *Journal of Economic Dynamics and Control* **35**(4), 604–615.

Havlik, J. & Straka, O. (2015), 'Performance evaluation of iterated extended kalman filter with variable step-length', *Journal of Physics: Conference Series* **659**.

Judd, K., Kubler, F. & Schmedders, K. (2000), '"computing equilibria in infinite-horizon finance economies: The case of one asset"', *Journal of Economic Dynamics and Control* **24**.

Klein, P. (2000), 'Using the generalized Schur form to solve a multivariate linear rational expectations model', *Journal of Economic Dynamics and Control* **24**(10), 1405–1423.
URL: *https://ideas.repec.org/a/eee/dyncon/v24y2000i10p1405-1423.html*

Kollman, R., Maliar, S., Malin, B. & P.Pichler (2011), '"comparison of solutions to the multi-country real business cycle model"', *Journal of Economic Dynamics and Control* **35**, 186–202.

Krueger, D., Kubler, F. & Malin, B. (2011), 'Solving the multi-country real business cycle model using a smolyak-collocation method', *Journal of Economic Dynamics and Control* **35**.

Maliar, L. & Maliar, S. (2015), 'Merging simulation and projection approaches to solve high-dimensional problems with an application to a new keynesian model', *Quantitative Economics* **6**(1), 1–47.

Mendoza, E. (2010), '"sudden stops, financial crises and leverage"', *American Economic Review* .

Schmitt-Grohe, S. & Uribe, M. (2004), 'Solving dynamic general equilibrium models using a second-order approximation to the policy function', *Journal of Economic Dynamics and Control* **28**(4), 755–775.

A Deriving iterative linear approximations to policy functions (step 8 of the algorithm)

Suppose we have obtained the linear approximation to the policy function for the control variables from the previous steps of the algorithm:

$$y = \widetilde{g}(x) = y_0^i + F^i(x - x_0^i)$$

or in deviation form:

$$y - y_0^i = F^i(x - x_0^i) \tag{8}$$

If step 8 of the algorithm has not been previously reached yet, then $\widetilde{g}(.)$ is the steady state policy function, $\widetilde{g}(.) = g_{\bar{x}}(.)$, and $y_0 = \bar{y}$ and $x_0 = \bar{x}$ are the steady state values of the control and state variables respectively. Otherwise, $\widetilde{g}(.)$ is the linear approximation to the policy functions obtained during step 8 of the previous iteration i of the algorithm, $\widetilde{g}(.) = g_{x_t}(.)$, while x_0^i and y_0^i are the "current-period" values of the state and control variables, used to construct the Taylor approximation to the equilibrium system of equations in that step.

Suppose that on the next iteration $i+1$ of the algorithm, we find that point $(x_1^{i+1}, y_1^{i+1}, x_0^{i+1}, y_0^{i+1})$ solves the deterministic version of our equilibrium system of equations, so that $f(x_1^{i+1}, y_1^{i+1}, x_0^{i+1}, y_0^{i+1}) = 0$ (here, we use x_1^{i+1} and y_1^{i+1} to denote the "next-period" values, and x_0^{i+1} and y_0^{i+1} to denote the "current-period" values). Using the notation similar to that in Gomme & Klein (2011), we can write the first-order Taylor approximation to the system of equations in (4) as:

$$A \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ y_{\tau+1} - y_1^{i+1} \end{bmatrix} = B \begin{bmatrix} x_\tau - x_0^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} \tag{9}$$

Note that in a standard application of the perturbation methods, one usually has $x_1^{i+1} = x_0^{i+1} = \bar{x}$ and $y_1^{i+1} = y_0^{i+1} = \bar{y}$.

It is convenient to re-write equation (8) as:

$$y_{\tau+1} - y_1^{i+1} = F^i \left(x_{\tau+1} - x_1^{i+1} \right) + F^i \left(x_1^{i+1} - x_0^i \right) + \left(y_0^i - y_1^{i+1} \right) \tag{10}$$

Plugging equation (10) into equation (9), we get:

$$A \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ F^i \left(x_{t+1} - x_1^{i+1} \right) + \tilde{F} \end{bmatrix} = B \begin{bmatrix} x_\tau - x_0^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} \tag{11}$$

where $\widetilde{F} = F^i \left(x_1^{i+1} - x_0^i \right) + \left(y_0^i - y_1^{i+1} \right)$.

Partition A into A_x and A_y, where the number of columns in A_x is the same as the number of state variables, and the number of columns in A_y is the same as the number of jump variables, and similarly partition B into B_x and B_y, so that:

$$\begin{bmatrix} A_x & A_y \end{bmatrix} \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ F^i \left(x_{\tau+1} - x_1^{i+1} \right) + \widetilde{F} \end{bmatrix} = \begin{bmatrix} B_x & B_y \end{bmatrix} \begin{bmatrix} x_\tau - x_0^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix}$$

or

$$A_x \left(x_{\tau+1} - x_1^{i+1} \right) + A_y F^i \left(x_{\tau+1} - x_1^{i+1} \right) + A_y \widetilde{F} = B_x \left(x_\tau - x_0^{i+1} \right) + B_y \left(y_\tau - y_0^{i+1} \right)$$

This can be re-written as:

$$\underbrace{\begin{bmatrix} A_x + A_y F^i, & -B_y \end{bmatrix}}_{\widetilde{A}} \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} = B_x \left(x_\tau - x_0^{i+1} \right) - A_y \widetilde{F} \qquad (12)$$

If $\widetilde{A}$ is invertible, we get the new solution for the state and control variables:

$$\begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} = \widetilde{A}^{-1} B_x \left(x_\tau - x_0^{i+1} \right) - \widetilde{A}^{-1} A_y \widetilde{F}$$

We have never encountered issues with inverting $\widetilde{A}$: unlike matrix A, $\widetilde{A}$ does not have rows filled with zeros which gives rise to singularity.[23]

B Equilibrium equations for the Sudden stop model of section 4.1

$$\begin{aligned}
q_t &= 1 + \Psi\left(\frac{z_t}{k_t}\right) + \frac{z_t}{k_t}\Psi'\left(\frac{z_t}{k_t}\right), \\
d_t &= \exp(\varepsilon_t^A) F_k(k_t, L_t), \\
\lambda_t(1+\tau_c) &= \left(c_t - \frac{L_t^\omega}{\omega}\right)^{-\sigma}, \\
-\lambda_t q_t^b + E_t \lambda_{t+1} &\left(1 + c_{t+1} - \frac{L_{t+1}^\omega}{\omega}\right)^{-\gamma} + \mu_t q_t^b = 0, \\
-\lambda_t q_t + E_t \lambda_{t+1} &\left(1 + c_{t+1} - \frac{L_{t+1}^\omega}{\omega}\right)^{-\gamma} (d_{t+1} + q_{t+1}) + \mu_t \kappa q_t = 0,
\end{aligned}$$

[23] As Klein (2000) points out, A is not invertible when static (intratemporal) equilibrium conditions are included in f. These static equations show up as rows entirely filled with zeros in matrix A because for the equations associated to such rows, all derivatives to variables in $t+1$ are zero. Instead $\widetilde{A}$ includes B_y, the derivatives to the jump variables at time t.