# Dynamic Perturbation*

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

February 18, 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. We also demonstrate that our method can solve models with a much larger number of state variables than alternative methods, and it does so faster and with more accuracy[1].

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

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

[1]Link to the latest version:. https://www.southampton.ac.uk/~alexmen/dynamic_perturbation.pdf

# 1 Introduction

It is often interesting to solve large models (hundreds of state variables) over a period of large transition. This makes standard perturbation methods–the typical technique used to solve large scale models–an inaccurate solution method.[2] To address this issue, We develop a technique which consists of applying perturbation methods at many points over the equilibrium path. The solution is much more accurate than that found with standard perturbation methods and can be applied to a broad class of DSGE models. Matlab code is available from www.southampton.ac.uk/∼alexmen/.

Similarly to the Parameterized Expectation Approach, the curse of dimensionality is broken because the model is only solved over the equilibrium path, see Den Haan & Marcet (1990). However, the method proposed here does not rely on a contraction mapping of the policy functions which often leads to convergence issues with large models. **?** propose a substantial improvement that allows them to solve a model with 80 state variables. The proposed method has successfully been adopted by Mennuni (2013) to solve a model with 890 variables of which 365 state variables over the transition, on a laptop.

The method is detailed in Section 2. But to get a visual sense of the accuracy of the method, Figure 4 compares it to a 2nd order perturbation of the steady state in a version of the RBC model with full depreciation, for which the true solution has analytical form.[3] As it can be appreciated by the Figure, the solution computed with the proposed method is much more accurate than that computed with the 2nd order perturbation of the state state, and in fact visually indistinguishable from the true solution.

[2] As Caldara et al. (2012) point out, in practice perturbation methods are the only computationally feasible method for solving medium- and large-scale dynamic stochastic general equilibrium (DSGE) models, but these techniques guarantee accurate solutions only in the proximity of the steady state.

[3] Since the true solution of this model is linear in the logs of the variables, Taylor expanding on the logs of the variables gives the exact solution with both methods. To introduce approximation error, the Taylor expansion is computed on the variables in levels. This way the computed policy functions do not coincide with the true ones, but are only tangent around the point where the Taylor expansion has been taken.

# 2 Dynamic Perturbation

In the next two sections, we first provide an intuitive description of our numerical algorithm, followed by a more technical step-by-step directions.

## 2.1 Outline of the algorithm

Large DSGE models are typically solved using the perturbation methods. One usually constructs a Taylor series approximation to the system of equilibrium conditions of the model around the model's non-stochastic steady state. For the values of the state variables close to the deterministic steady state, this solution method is usually quite accurate even for models that induce large Jensen inequalities (Caldara et al. (2012)). On the other hand, the quality of approximation can deteriorate substantially for the values of the state variables far away from the steady state, or if the model exhibits large non-linearities. Thus, applying the standard perturbation methods can be problematic if the researcher is concerned with 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 with the zero lower bound for the nominal interest rate, often lead to policy functions that exhibit kinks, and thus are not easily amenable to the standard perturbation methods. To resolve this, we propose a new methodology which essentially consists of applying repeated local approximations over the entire transition path, between the initial point and the steady state. The goal of our approach is to achieve a uniform precision along the whole simulated solution path. To explain how our approach works, we need to introduce some notation.

Following Schmitt-Grohe & Uribe (2004) and Gomme & Klein (2011) , the model can be expressed as

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

where $E_t$ is the expectation given information at time $t$, $x_t$ is a vector of current-period realizations of the state variables, $y_t$ is a vector of current-period realizations of the control (or "jump") variables of the model, while $x_{t+1}$ and $y_{t+1}$ are their corresponding next-period realizations.

A recursive representation of the solution takes the form of the policy function for the control variables:

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

and state variables:

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

where, following Schmitt-Grohe & Uribe (2004), $\sigma$ is a scalar that scales the variance of $\eta_t$, while $\{\eta_t\}$ is an i.i.d. sequence of innovations with zero mean and variance matrix $\Sigma_\eta$.

Suppose we know the initial values of the state variables, $x_0$, and the realized sequence of shocks, $\{\eta_t\}_{t=0}^{T}$, and we want to find the corresponding sequence of state and control variables that solve the model. An approximated solution can be found by Taylor expanding the deterministic version of equation 1:

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

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

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

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. Using the steady state as a point of approximation circumvents this problem, since by definition, the values of the variables are not expected to change in a steady state. However, as 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 the current-period control variables $y_t$ and next-period state and control variables, $x_{t+1}$ 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). We will label this as the "finding local dynamic solution"

(FLDS) task. 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 $\eta_{t+1}$ to obtain the values of the state and control variables in the following period.

To achieve FLDS, we construct an auxiliary deterministic path between the current-period state vector $x_t$ and the steady state vector $\bar{x}$, which then we can trace backwards. To construct this auxiliary path, we use the policy functions obtained by approximating the model 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 $\bar{x}$. Since in practice, we stop when we are sufficiently close to the steady state $\bar{x}$, this generates a finite auxiliary path $\{x_t, \widetilde{x}_{t+1}, \ldots, \widetilde{x}_{\overline{T}}\}$ such that $\widetilde{x}_{\overline{T}}$ is close to $\bar{x}$.[4]

Next, we move backwards along this auxiliary path. We start with finding policy functions approximated around the last point in the auxiliary path, $\widetilde{x}_{\overline{T}}$. As mentioned earlier, to Taylor expand around this point, we need to have a local solution to the system given $\widetilde{x}_{\overline{T}}$. In the next section we show how one can find such a solution, using the steady state policy functions to find the next-period control variables. In short, the next-period control variables, $y_{\overline{T}+1}$, can be found for any given values of next-period state variables $x_{\overline{T}+1}$ using the steady-state policy functions, $y = g_{\bar{x}}(x)$.

Since F is a function of $(n_x + n_y)$ equations in $2*(n_x + n_y)$ unknowns, one can solve such system for $x_{\overline{T}+1}$ and $y_{\overline{T}}$, using the steady-state policy functions $y = g_{\bar{x}}(x)$ with $x_{\overline{T}+1}$ as input to find the next-period control variables, $y_{\overline{T}+1}$.

This allows us to find $\widetilde{\widetilde{y}}_{\overline{T}}$ and $\widetilde{\widetilde{x}}_{\overline{T}+1}$ (and $\widetilde{\widetilde{y}}_{\overline{T}+1} = g_{\bar{x}}(\widetilde{\widetilde{x}}_{\overline{T}+1})$) that solve the system (4). At this point we can solve for policy functions approximated around this new point using a method that is similar to standard perturbation methods.[5] After this, we move backwards to the previous point in the auxiliary path, $\widetilde{x}_{\overline{T}-1}$, solving the system after substituting the policy functions found during the previous step, approximated at $\widetilde{x}_{\overline{T}}$, for the future control variables. We repeat this backward step until we reach $x_t$. At that point, our FLDS task is solved: we have policy functions at $x_t$! We use these policy functions and the actual realization of shocks $\eta_{t+1}$ to obtain $y_t$ and $x_{t+1}$. Next, we use $x_{t+1}$ as a new starting point and loop.

[4] 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 $\bar{x}$.

[5] But it has weaker invertibility restrictions.

In summary, starting with $x_t = x_0$ and repeating the above algorithm, we construct a model solution path, $\{x_t, y_t\}_0^T$ of desired length $T$ that correspond the s sequence of realized shocks $\{\eta_1, \ldots, \eta_T\}$.

The precision of the solution based on perturbation around the steady state can deteriorate quickly as one moves further away from the point of approximation. In contrast to this, our algorithm achieves a uniform precision along the whole simulated solution path. To formally state this, let us define:

$$F(x, \sigma, h, g) \equiv E_x\left(f\left[h(x, \sigma) + \sigma\eta, g(h(x, \sigma) + \sigma\eta, \sigma), x, g(x, \sigma)\right]\right), \tag{5}$$

where $E_x$ is the expectation over $\eta$ given the state variables $x$. Pick some desired precision $\hat{\varepsilon} > 0$. The algorithm seeks to find points $\{x_t\}_1^T$ such that, given an initial condition $x_0$ and sequence of shocks $\{\eta_t\}_1^T$, we achieve a uniform precision along the whole simulated path:

$$|F(x_t, 0, h_{x_t}, g_{x_t})| < \hat{\varepsilon} \quad for \quad t = 0, ..., T-1, \tag{6}$$

where $h_{x_t}, g_{x_t}$ are policy functions approximated around $x_t$.[6]

## 2.2 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.

1. Taylor expand 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 $h_{\bar{x}}(x, \sigma)$, $g_{\bar{x}}(x, \sigma)$ for $\sigma = 0$. If these are stable, then go to the next step (for stability, see for instance Blanchard & Kahn (1980)).[7]

[6] Note that in equation (6) $\sigma = 0$. This is because the expectation operator in equation 5 is replaced by the certainty equivalence assumption of zero innovations.

[7] 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.

2. Put $t_0 = 1$ (it indicates the initial condition or the last point solved for).

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

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

4. Set the shocks to 0 and $\sigma = 0$. Start from $x_{t_0}$ and, using the policy function $\widetilde{h}(\cdot)$ (with $\sigma = 0$), generate a sequence $\{\widetilde{x}_t\}_{t_0}^{\bar{T}}$ with $\bar{T} > T$.[8] If this sequence does not converge to the steady state, increase $\bar{T}$ and repeat this step.

   The following steps trace the auxiliary path backwards from the steady state to the current initial point, $x_{t_0}$, and compute the next point in our solution sequence.

5. Set $t = \bar{T}$

6. Set $x = x_t$.

7. Find $x'$ and $y$ such that $f(x', \widetilde{g}(x', 0), x, y) = 0$ (note that we substituted $y'$ with $\widetilde{g}(x', 0)$).[9]

8. Derive linear approximations to the policy functions, $h_{x_t}(\cdot)$, $g_{x_t}(\cdot)$, using the Taylor expansion of $f(x', y', x, y) = 0$, around the local solution point $(x', y', x, y)$ (with $y' = \widetilde{g}(x', 0)$) 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 $\widetilde{h}(\cdot) = g_{x_t}(\cdot)$ and $\widetilde{g}(\cdot) = g_{x_t}(\cdot)$.

10. If $t > t_0$, set $t = t - 1$ and go back to step 6.

11. If $t = t_0$, we have found a local solution to $f(x', y', x, y) = 0$ with $x = x_{t_0}$. This solves our FLDS task. We can use this point to construct a Taylor approximation to the equilibrium system of equations (1), and obtain local

[8] 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.

[9] This step is similar to a step in the policy function iteration algorithm. Here, this step makes sure that the function $f(\hat{x}_1, \hat{y}_1, \hat{x}, \hat{y}) = 0$ holds and hence a Taylor expansion is admissible.

approximation to the policy functions, $h_{x_{t_0}}(x,\sigma)$ and $g_{x_{t_0}}(x,\sigma)$[10]. We use these policy functions and the realizations of the shocks, $\eta_{t_0}$, to obtain and store the next values of state and control variables in our model simulation, $x_{t_0+1} = h_{x_{t_0}}(x_{t_0},\sigma) + \sigma\eta_{t_0}$, $y_{t_0} = g_{x_{t_0}}(x_{t_0},\sigma)$, and then go to the next step.

12. If $t_0 = T$, the whole solution has been found! Otherwise, set $t_0 = t_0 + 1$ 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 nonlinearity 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.

The iteration procedure over the equilibrium path is reminiscent of the Parametrized Expectation Approach (PEA). See Den Haan & Marcet (1990) and Marcet & Lorenzoni (1999)): both algorithms break the curse of dimensionality by only approximating the global policy function over the equilibrium path rather than over the entire state space. In practice however, the PEA may show convergence problems that make its implementation hard for high-dimensional applications.[11] An important advantage of the proposed method is that, unlike the PEA, it does not iterate on the equilibrium path and hence does not rely on a contraction mapping, which explains why convergence problems do not arise. On the other hand, the invertibility conditions necessary to derive policy functions through perturbation methods need to be satisfied at all the points where the perturbation is applied; these invertibility conditions have not been violated in the models solved so far.[12]

[10] 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.

[11] See in this respect the improvements made by Judd et al. (2009) (typically, this approach is also less accurate because it interpolates across the points).

[12] In addition, this method is only valid on the specific equilibrium path generated by the initial conditions and the time series of innovations. However, it is possible to simulate for a long time horizon and then interpolate over the solution to obtain policy functions valid over a larger portion of the ergodic set.

# 3 Comparison to other solution methods

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

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:

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

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 $\rho$ 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:

$$\max_{\{c_{i,t},k_{i,t+1}\}_{t=0,\dots,\infty}^{i=1,\dots,N}} E_0 \sum_{i=1}^{N} \lambda_i \left[ \sum_{t=}^{\infty} \beta^t u(c_{i,t}) \right]$$
$$\text{s.t. } \sum_{i=1}^{N} c_{i,t} + \sum_{i=1}^{N} k_{i,t+1} = \sum_{i=1}^{N} k_{i,t}(1-\delta) + \sum_{i=1}^{N} A a_{i,t} f(k_{i,t})$$

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

The solution must satisfy $N$ Euler equations:

$$1 = \beta E_t \left[ \frac{u'(c_{i,t+1}}{u'(c_{i,t})} (1 - \delta + A a_{i,t} f'(k_{i,t+1}) \right] \tag{7}$$

## 3.2 One Country and Full Capital Depreciation

To evaluate this algorithm, I test it on the model in note **??**, equations **??**-**??**, and with full depreciation, for which the analytical solution is known. I 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

$$\max_t [\max(|x_t^* - x_t^{***}|, |y_t^* - y_t^{***}|)] \tag{8}$$

[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

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

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 4. I 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. Solving the main model of the paper in section **??** takes about 10 hours and 20 minutes. As a measure of accuracy, I compute the error

$$|f(E_t(x_{t+1}), E_t(y_{t+1}), x_t, y_t)| \tag{9}$$

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 (9) 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 $1.710^{-12}$.[15]

## 3.3 Two Countries

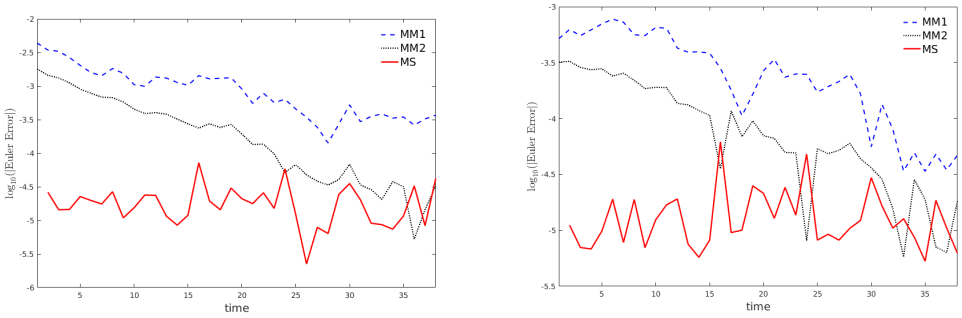
We start our comparison with the case of $N = 2$ countries. Obtaining a numerical solution in this case is a fairly simple task, as the total number of state variables, $2N = 4$, is low. However, even this simple setup allows us to demonstrate some advantages of our solution method, both in terms of accuracy and speed.

[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.

[15]Although this result is quite reassuring, it should be noticed that an explosive solution, or a solution which alternates explosive to implosive patterns, can be consistent with this result. However, from a graphical inspection, the solution does not present an oscillating path and all variables converge to a steady state.

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 & Maliar (2015) using both first-order and second-order polynomials as the basis functions (MM2)[16]. 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[17].

Figure 1: Euler equation errors, 2-country model




(a) $k_0 = 0.5k_{ss}$ (b) $k_0 = 1.5k_{ss}$

[16]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.

[17]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)

Table 1: Solution Time

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

Figure 1 compares Euler equation errors along a simulated path for consumers in country 1[18]. In the left panel, we show the Euler errors from the simulated path where we start the simulation with capital in both countries 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)[19].

## 3.4 Changing volatility of shocks

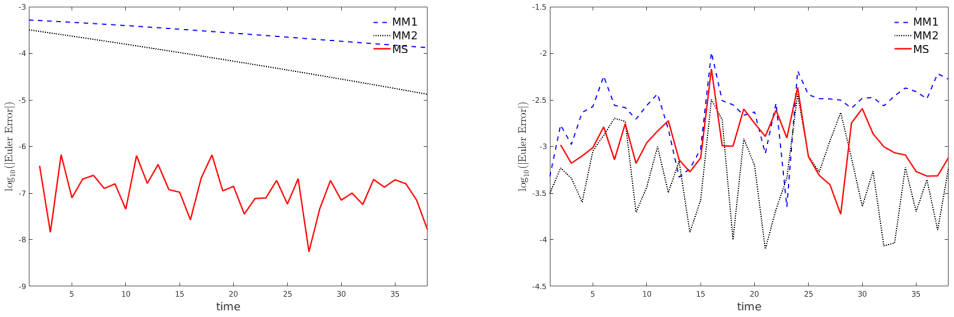
Our numerical method is based on local first-order approximations to the equilibrium system of equations, and thus implies certainty equivalence (see Schmitt-Grohe & Uribe (2004) for the details). This means that our method does not capture the impact of the size of the shocks on the policy

[18]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.

[19]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

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.

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




(a) $\sigma = 1^{-6}$ (b) $\sigma = 0.1$

Figure 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 the size of the shocks. Subplot (a) demonstrates the results for the case when $\sigma = 1^{-6}$. 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[20].

## 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 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; $L_\infty$ 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 ($\sigma$=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:

[20]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 | $L_1$ | $L_\infty$ | CPU | $L_1$ | $L_\infty$ | CPU | $L_1$ | $L_\infty$ | 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 |

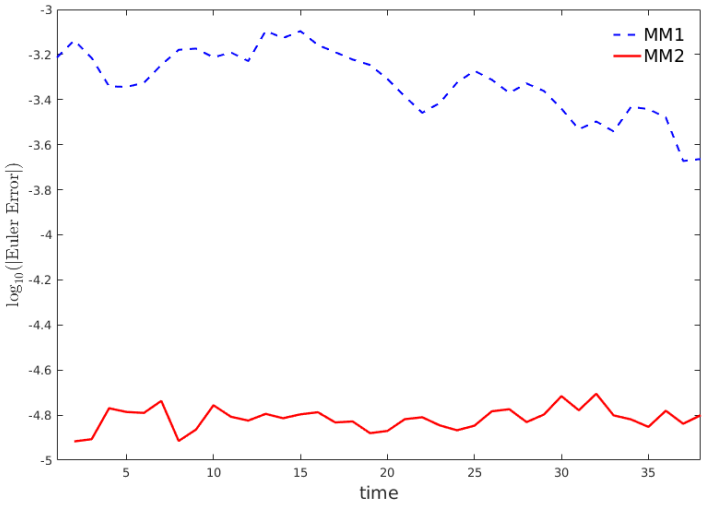



Figure 3

# 4 Figures

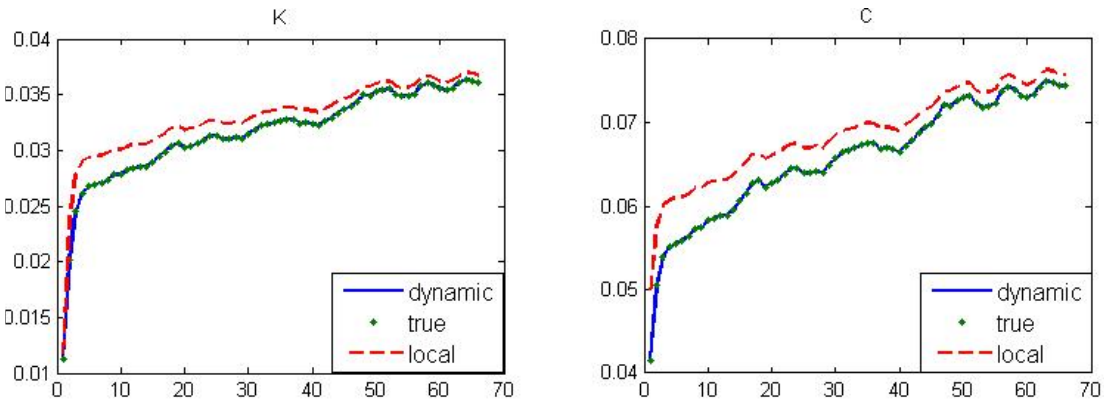



Figure 4: Capital and consumption from the analytical example of Appendix 2 compared with the solution computed with local and dynamic perturbation.

# References

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

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.

Den Haan, W. J. & Marcet, A. (1990), 'Solving the stochastic growth model by parameterizing expectations', *Journal of Business & Economic Statistics* **8**(1), 31–34.

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.

Judd, K., Maliar, L. & Maliar, S. (2009), Numerically stable stochastic simulation approaches for solving dynamic economic models, NBER Working Papers 15296, National Bureau of Economic Research, Inc.

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.

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.

Marcet, A. & Lorenzoni, G. (1999), The parameterized expectation approach: some practical issues, *in* R. Marimon & A. Scott, eds, 'Computational Methods for the Study of Dynamic Economies', pp. 143–171.

Mennuni, A. (2013), 'Labor force composition and aggregate fluctuations', *Discussion Paper Series In Economics And Econometrics 1302, Economics Division, School of Social Sciences* .

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:

$$y - y_0^i = F^i(x - x_0^i) \tag{10}$$

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:

$$A \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ y_{\tau+1} - y_1^{i+1} \end{bmatrix} = B \begin{bmatrix} x_\tau - x_0^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} \tag{11}$$

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 (10) as:

$$y_{\tau+1} - y_1^{i+1} = F^i \left( x_{\tau+1} - x_1^{i+1} \right) + F^i \left( x_1^{i+1} - x_0^i \right) + \left( y_0^i - y_1^{i+1} \right) \tag{12}$$

Plugging equation (12) into equation (11), we get:

$$A \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ F^i \left( x_{t+1} - x_1^{i+1} \right) + \tilde{F} \end{bmatrix} = B \begin{bmatrix} x_\tau - x_0^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} \tag{13}$$

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:

$$\underbrace{\left[A_x + A_y F^i, \quad -B_y\right]}_{\widetilde{A}} \begin{bmatrix} x_{\tau+1} - x_1^{i+1} \\ y_\tau - y_0^{i+1} \end{bmatrix} = B_x \left(x_\tau - x_0^{i+1}\right) - A_y \widetilde{F} \tag{14}$$

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.[21]

[21] 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 $t$.