WIF - Institute of Economic Research

Multi-Dimensional Transitional Dynamics: A Simple Numerical Procedure

Timo Trimborn, University of Hamburg
Karl-Josef Koch, University of Siegen
Thomas M. Steger, ETH Zurich

Working Paper 04/35
December 2004

Economics Working Paper Series

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Multi-Dimensional Transitional Dynamics: A Simple Numerical Procedure[*]

Timo Trimborn, University of Hamburg[†]
Karl-Josef Koch, University of Siegen
Thomas M. Steger, ETH Zurich

This Version: March 2005

Abstract

We propose the relaxation algorithm as a simple and powerful method for simulating the transition process in growth models. This method has a number of important advantages: (1) It can easily deal with a wide range of dynamic systems including multi-dimensional systems with stable eigenvalues that differ drastically in magnitude. (2) The application of the procedure is fairly user friendly. The only input required consists of the dynamic system. (3) The variant of the relaxation algorithm we propose exploits in a natural manner the infinite time horizon, which usually underlies optimal control problems in economics. Overall, it seems that the relaxation procedure can easily cope with a large number of problems which arise frequently in the context of macroeconomic dynamic models. As an illustrative application, we simulate the transition process of the well-known Jones (1995) model.

JEL classification: C61; C63; O40

Keywords: Saddlepoint problems; Transitional dynamics; Economic growth; Multidimensional stable manifolds

[*] For helpful comments and suggestions we would like to thank Beatriz Gaitan Soto, Christian Groth, Chad Jones, Bernd Lucke and Holger Strulik.

[†] Timo Trimborn: University of Hamburg, Department of Economics, IWK, Von-Melle-Park 5, 20146 Hamburg, Germany, trimborn@econ.uni-hamburg.de, Phone: + 49.40.42838 - 2082, Fax: + 49.40.42838 - 6314.

1 Introduction

Dynamic macroeconomic theory nowadays relies heavily on infinite horizon optimization models which usually give rise to a system of nonlinear differential equations. This dynamic system is then interpreted to describe the evolution of the economy under consideration. Many studies in the field of growth theory have confined their analysis to the balanced growth path (BGP). A comprehensive understanding of the respective model under study requires, however, that we investigate in addition the transition process. At least two important arguments support this view: First, the positive and normative implications might differ dramatically depending on whether an economy converges towards its BGP or grows along the BGP (e.g. Jones, 1995). Second, dynamic macroeconomic models are often employed to conduct comparative welfare investigations of different policy regimes or instruments. In this context, the transition process needs to be taken into account. Linearizing the dynamic system might be appropriate in many cases but can be potentially misleading especially when the analysis aims at a Pareto-ranking of different policy instruments. This overall perspective is best summarized by the following statement due to Jonathan Temple (2003, p. 509): *Ultimately, all that a long-run equilibrium of a model denotes is its final resting point, perhaps very distant in the future. We know very little about this destination, and should be paying more attention to the journey.*

Especially in growth theory the models under study are very often multi-dimensional in the sense that there is more than one (predetermined) state variable. Usual stability properties then imply that the stable manifold is also multi-dimensional.[1] Examples for models which fit into this class comprise R&D-based growth models (e.g. Romer, 1990; Jones, 1995; Eicher and Turnovsky, 1999) as well as human-capital based growth models (e.g.

[1] In the case of saddle-point stability, the dimension of the stable manifold equals the dimension of the state space, while indeterminacy implies that the dimension of the stable manifold exceeds the dimension of the state space.

Lucas, 1988; Mulligan and Sala-i-Martin, 1993; Benhabib and Perli, 1994). Moreover, if the dynamic system is characterized by (stable) eigenvalues which differ substantially in magnitude, then usual procedures are either not applicable or highly inefficient.[2] It is important to notice that this characteristic property is not at all a special (or even pathological) case but instead occurs quite frequently.

The paper at hand contributes to the literature on dynamic macroeconomic theory by proposing the relaxation algorithm as a powerful method to simulate the transition process in growth models. We will argue that this procedure is in general well-suited and highly efficient. This will be demonstrated by simulating the transition process of the well-known Jones (1995) model, which is characterized by a two-dimensional stable manifold with the potential of non-monotonic adjustments.[3]

Turning to the related literature, there are, of course, a number of procedures to simulate the transition process of dynamic macroeconomic models. In the context of growth theory, the most prominent approaches comprise shooting (e.g. Judd, 1998, Chapter 10), time elimination (Mulligan and Sala-i-Martin, 1991), backward integration (Brunner and Strulik, 2002), the projection method (Judd, 1992) as well as the discretization method of Mercenier and Michel (1994). The similarities and differences of the relaxation procedure and the methods mentioned above will be discussed concisely below. This enumeration shows that there are already some procedures which have been used in economics to solve dynamic systems. Nonetheless, we think that there are a number of good reasons to include additionally the relaxation procedure into the toolbox of dynamic macroeconomic theory:

First, our experiences with the relaxation algorithm are positive through-

[2] In the mathematical literature, differential equations exhibiting this structural characteristic are labelled “stiff differential equations”.

[3] In addition, for usual calibrations the Jones (1995) exhibits stable eigenvalues which differ drastically in magnitude. To the best of our knowledge, there is no study simulating the transition process of the Jones (1995) or the Eicher and Turnovsky (1999) model.

out. We have applied the procedure to a wide range of dynamic systems, including multi-dimensional systems (showing non-monotonic adjustments) with stable eigenvalues that differ drastically in magnitude as well as highly dimensional computable general equilibrium models. The algorithm performed amazingly well. It is remarkable that an increase in the dimension of the model under study does not cause any conceptual problems. The researcher need not take restrictions with respect to the model dimension into account. In addition, the procedure seems to be efficient with respect to computer time.

Second, the application of the procedure is fairly user friendly. Specifically, the only input which must be provided by the researcher consists in the dynamic system and the set of underlying parameters. No preliminary manipulations of the dynamic system under study must be conducted before the procedure can be applied; this is different from most other procedures as described in Section 3.

Third, the variant of the relaxation algorithm we propose exploits in a natural manner the infinite time horizon which usually underlies standard optimal control problems. This is achieved by a simple transformation of real calendar time into a transformed time scale (as explained in Section 2.1). For most other procedures, this issue must explicitly be dealt with (explained in Section 3).

Overall, it seems that the relaxation algorithm can easily cope with a large number of problems which arise frequently in the context of multi-dimensional, infinite-time horizon optimal control problems.

The paper is structured as follows: In Section 2, the relaxation procedure is described concisely and then evaluated numerically employing the Ramsey-Cass-Koopmans model as a basic example. Section 3 provides a short comparison to alternative methods. In Section 4, we apply the procedure to simulate the transition process of the well-known Jones (1995) model. Section 5 summarizes and concludes. The appendix (Section 6) provides a more formal description of the relaxation procedure. Finally, the relaxation algorithm has been programmed in MatLab. This program together with a concise instruction manual is available for free download at: www.rrz.uni-hamburg.de/IWK/trimborn/relaxate.htm.

2 The relaxation procedure

2.1 Description of the relaxation procedure

The principle of relaxation can be applied to various numerical problems. Here we use it to solve a differential equation numerically. Relaxation type algorithms applied to differential equations have two very useful properties. First of all, they can easily cope with boundary conditions, such as initial conditions for state variables and transversality conditions of optimal growth. Second, additional equations, e.g. equilibrium conditions or feasibility constraints, can be incorporated straight away. Beyond, by transformation of the (independent) time variable one can solve infinite horizon problems, as they arise from many dynamic optimization problems in economics.

Suppose we want to compute a numerical solution of a differential equation in terms of a large (finite) sequence of points representing the desired path. To start with, we take an arbitrary trial solution, typically not satisfying the slope conditions implied by the differential equation nor the boundary conditions. We measure the deviation from the true path by a multidimensional error function and use the derivative of the error function to improve the trial solution in a Newton type iteration. Hence, at each point of the path the correction is related to the particular inaccuracy in slope and in solving the static equation. The crucial difference to the various shooting methods is the simultaneous adjustment along the path as a whole.

Figure 1 illustrates the adjustment by relaxation of a linear initial guess towards the saddle path in the Ramsey-Cass-Koopmans model. The initial guess starts with a fixed initial value of the state variable k and an arbitrary initial value of the control variable c. It consists of 30 mesh points lined up equidistantly between the starting point and the known steady state of the model. Evaluating the multidimensional error function the algorithm realizes that the fit to the differential equation can be improved by an upward shift of the curve without jeopardizing the boundary conditions. After a few steps the error is sufficiently small and the algorithm stops.

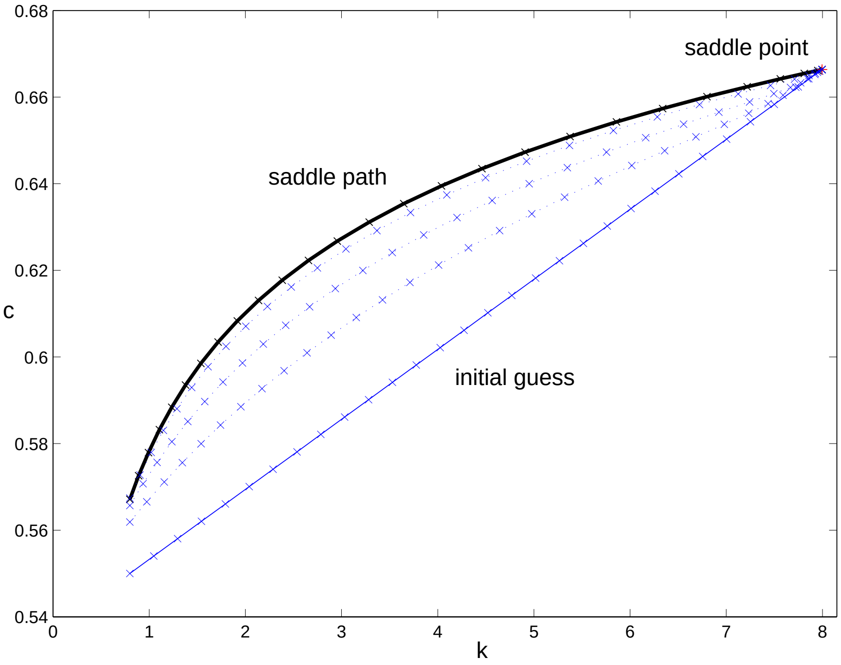

Figure 1: Relaxation in the Ramsey-Cass-Koopmans model

The outline of the algorithm proposed in this paper leans on Press, Flannery, Teukolsky and Vetterling (1989, pp. 645-672). We have implemented the algorithm in MatLab. The code is published for free download in the internet[4] and a print version is available on request.[5]

[4] http://www.rrz.uni-hamburg.de/IWK/trimborn/relaxate.htm

[5] In the appendix we give a detailed description of the algorithm.

We apply the method to the following kind of problem: Consider a system of $\tilde{N}$ ordinary differential equations together with $N - \tilde{N}$ (static) equations in N real variables. This system describes a vector field on an $\tilde{N}$-dimensional surface in $\mathbb{R}^N$. We impose a list of n_1 boundary conditions at the starting point and n_2 at the end point of a path sufficient to determine a particular trajectory. To meet all dimensional requirements n_1 and n_2 must add up to $\tilde{N}$.

For the finite representation of the problem we fix a time mesh of M points in time. In case of an infinite time horizon we choose a transformation to map the interval $[0, \infty]$ to $[0, 1]$. At each point in time an N-dimensional vector has to be determined. We approximate the differential equation by $M - 1$ systems of equations of dimension $\tilde{N}$ for the slope between neighboring mesh points. Together with $\tilde{N}$ boundary conditions we have an $M \times \tilde{N}$ dimensional system of equations. After adding the $N - \tilde{N}$ static equations which have to hold at each M mesh point we have incorporated all restrictions available. The final system of nonlinear equations is of dimension $M \times N$ and involves the same number of unknowns.

We apply a Gauß-Newton procedure to compute a root of this system. Step by step we adjust the trial solution until the error is sufficiently small. This involves the solution of a linear equation with the Jacobian matrix of the system of nonlinear equations. At first glance there seems little chance to achieve good solutions because the complexity of the problem is proportional to the size of the Jacobian matrix which is quadratic in M. However, the Jacobian is not an arbitrary matrix of dimension $M \times N$.

The Jacobian matrix inherits a specific structure from the approximation of the differential equation. The boundary conditions and the static equations each depend only on one respective vector, and the interior slope conditions only on neighboring vectors. Hence the Jacobian matrix shows nonzero entries only close to the diagonal. This can be used to solve the linear system by a special version of a Gauß algorithm carried out recursively on N-dimensional blocks along the diagonal. This recursive procedure allows to increase the number M of mesh points without increasing the dimension of the blocks. Only the number of blocks increases in proportion to M. The complexity of the problem is only linear in the number of mesh points and not quadratic. Hence, a fairly good approximation of the continuous path is possible without using too much computer time.

2.2 Implementation of the algorithm

In this section, we describe the steps which must be taken when implementing the relaxation algorithm using the Ramsey-Cass-Koopmans model (Ramsey, 1928; Cass, 1965; Koopmans, 1965) as an example. It should be noticed that this description serves as an illustration only. The researcher who intends to simulate a specific model using the program which is provided as a supplement to this paper need not follow these steps.

It is well known that this simple growth model exhibits saddle-point stability and hence the determination of the solution is all but trivial.[6] The model gives rise to a system of two differential equations for consumption and capital per effective labor (Barro and Sala-i-Martin, 2004, Chapter 2):

$$\dot{c} = \frac{c}{\theta}\left(\alpha k^{\alpha-1} - (\delta + \rho + x\theta)\right) \tag{1}$$

$$\dot{k} = k^{\alpha} - c - (n + x + \delta)k, \tag{2}$$

where α denotes the elasticity of capital in production, n the population growth rate, δ the depreciation rate, x the exogenous growth rate of technology, ρ the parameter for time preference and θ the inverse of the intertemporal elasticity of substitution, respectively. The steady state is at $k^* = \left(\frac{\alpha}{\delta+\rho}\right)^{\frac{1}{1-\alpha}}$ and $c^* = (k^*)^{\alpha} - (n + x + \delta)k^*$ and saddle point stable.

As a first step, one must choose a time mesh, i.e. a set of points in time at which the solution should be calculated. We select the time mesh to be uniform in the transformed time scale (as explained in section 2.1).

[6] Nonetheless, the model is comparably simple in that the stable manifold is one dimensional. We will turn to a model with a multi-dimensional stable manifold below.

Second, the two differential equations have to be transformed into two non-linear equations which describe the slope between two neighboring mesh points. These equations have to be satisfied between every two mesh points. For M mesh points this leads to $2 \cdot (M-1)$ nonlinear equations.

Third, two boundary condition have to be chosen to complete the set of equations to $2 \cdot M$. In this example the relaxation algorithm needs one initial boundary condition and one terminal boundary condition. We set the initial value of the state variable (capital) equal to 10% of its steady state value. For the terminal boundary condition there are several possibilities to formulate an equation. It would be possible to choose each of the two equations (1) or (2) and set the RHS equal to zero. However, here the steady state values for consumption and capital can be computed analytically and, therefore, we can set consumption equal to its steady state value as the terminal boundary condition. It should be noted that only one boundary condition is needed. Thus the algorithm does not make use of the knowledge of the steady state value of capital. It is reached automatically.

At last an initial guess for the solution has to be made. Here we choose c and k to be constant at their steady state values $(c_t, k_t) \equiv (c^*, k^*)$.[7] The Newton procedure always converged quickly, indicating a high degree of robustness with respect to the initial guess.

2.3 Evaluation of the procedure

For the special parametrization $\theta = \frac{\delta+\rho}{\alpha(\delta+n+x)-x}$ the solution can be expressed analytically; this is due to the fact that the representative consumer chooses a constant saving rate $s = \frac{1}{\theta}$ (Barro and Sala-i-Martin, 2004, pp. 106-110).[8] This allows us to compare the computed results with the analytical solution, which has a precision close to the machine epsilon. The relative er-

[7] This is in contrast to Figure 1 where the initial guess is an upward sloping line.

[8] The analytical solution is $k(t) = \left[\frac{1}{(\delta+n+x)\theta} + \left(k_0^{1-\alpha} - \frac{1}{(\delta+n+x)\theta}\right) e^{-(1-\alpha)(\delta+n+x)t}\right]^{\frac{1}{1-\alpha}}$ and $c(t) = (1-\frac{1}{\theta})k(t)^{\alpha}$.

ror is computed for every mesh point. Table 1 shows the maximum relative error of consumption and capital per effective labor for different numbers of mesh points. In addition, the quadratic mean error of combined c and k provides information about the distribution of the error.[9] Table 1 reveals that multiplying the number of mesh points by x reduces the maximum error of each solution vector by the factor $\frac{1}{x^2}$, which indicates the order 2 of the difference procedure. Even with a moderate number of mesh points and therefore a short computation time, a sufficiently high degree of accuracy can be achieved. Moreover, the accuracy can be improved to a very high degree by increasing the number of mesh points.[10] The treatment of higher dimensional systems with multi-dimensional stable manifolds is largely analogous to the example described above. This is the reason why the the algorithm performs similarly well for more complicated models.

Table 1: Accuracy of the relaxation algorithm for the Ramsey-Cass-Koopmans model

number of mesh points	max error c	max error k	mean error
10	$< 1.3 \cdot 10^{-2}$	$< 3.4 \cdot 10^{-2}$	$< 3.0 \cdot 10^{-3}$
100	$< 1.1 \cdot 10^{-4}$	$< 8.6 \cdot 10^{-5}$	$< 2.7 \cdot 10^{-6}$
1,000	$< 1.1 \cdot 10^{-6}$	$< 8.5 \cdot 10^{-7}$	$< 8.2 \cdot 10^{-9}$
10,000	$< 1.1 \cdot 10^{-8}$	$< 8.5 \cdot 10^{-9}$	$< 2.6 \cdot 10^{-11}$
100,000	$< 1.1 \cdot 10^{-10}$	$< 8.5 \cdot 10^{-11}$	$< 8.2 \cdot 10^{-14}$

3 Comparison to other procedures

In this section, we compare very briefly the relaxation procedure to common alternative solution methods. Other solution techniques for this type of problem are backward integration (Brunner and Strulik, 2002), multiple

[9] It is defined as $\varepsilon = \frac{1}{NM}\sqrt{\sum_{i=1}^{N} \varepsilon_{c_i}^2 + \sum_{i=1}^{N} \varepsilon_{k_i}^2}$ with ε_{c_i} and ε_{k_i} denoting the relative error of k and c at mesh point i, respectively.

[10] It should be mentioned that the allocation of the mesh was chosen exogenously. The accuracy of the algorithm could be improved with a self allocating time mesh.

shooting (e.g. Judd, 1998, Chapter 10), time elimination (Mulligan and Sala-i-Martin, 1991), projection methods (e.g. Judd, 1992; Judd, 1998, Chapter 11) and the method of Mercenier and Michel (1994). We keep this section brief and restrict our comparison to related methods since most of the procedures and their relative advantages are described in Judd (1998) and Brunner and Strulik (2002).

The method of backward integration as described in Brunner and Strulik (2002) exploits the numerical stability of the backward looking system by inverting time. By starting near the steady state of the transformed system, the resulting initial value problem is stable and the solution converges towards the stable manifold of the forward looking system quickly. Therefore, the method can solve systems with one-dimensional stable manifolds very conveniently. For multi-dimensional manifolds Brunner and Strulik (2002) suggest to generate starting values on an orbit around the steady state. To pass through a pre-specified point, which is determined by a specific shock under study, it is necessary to iterate until the trajectory hits this point. However, if the real parts of the eigenvalues associated with the stable manifold of the forward looking system differ substantially, the problem of stiff differential equations occurs. It is well-known that these problems are very hard to handle numerically. If the difference of the stable eigenvalues is sufficiently high, it is impossible to meet the pre-specified point, because the backward shooting trajectories will be attracted by the submanifold, which is determined by the eigenvalue with the smallest real part. Therefore, the resulting trajectories cannot represent a specified shock and potentially have no economic meaning.

Mercenier and Michel (1994) propose to transform the continuous time, infinite horizon problem into a finite horizon maximization problem in discrete time with the same steady state. The transformed problem can be solved with a static optimization procedure. This leads to a system of nonlinear equations, which can be solved with a Newton algorithm similar as in the relaxation algorithm. Our approach is to solve the system of differential equations. Here the discretisation is done at a later stage. To apply the relaxation algorithm the researcher simply has to paste the system of differential equation into the algorithm instead of converting the whole maximization problem. Apart from simplicity, the relaxation algorithm has some further advantages.

First, the relaxation procedure is more general, since the system of differential equations can be attained in different ways, not only by a single maximization problem. In particular, the approach of Mercenier and Michel for steady state invariance requires the discount factor to be constant. However, if the firm also faces an intertemporal optimization problem the discount factor is related to the real interest rate which might not be constant during the transition. Therefore, steady state invariance cannot be guaranteed and the performance of the method deteriorates. Second, the relaxation algorithm can deal with a compactification of the time interval. Therefore, it is not necessary to choose an adequate terminal time where the optimization is truncated. Also, the treatment of a post terminal stationary phase does not apply. Third, in the approach of Mercenier and Michel the discretization procedure is fixed. The relaxation algorithm leaves room for selecting different discretisation rules, also of higher order. This leads to a higher level of accuracy with the same number of mesh points. The discretisation rule of the method of Mercenier and Michel is a first order rule, whereas the relaxation procedure uses a second order rule.[11]

Projection methods as they are introduced in Judd (1992) and Judd (1998, Chapter 11) cover a very wide range of algorithms. They are considered to be fast and accurate, but also they require a high programming effort. Moreover, they are usually applied to solve for the policy function. This does not work for non-monotonic adjustments along multi-dimensional

[11] When multiplying the number of mesh points with x a first order rule leads to a reduction of the global error by $\frac{1}{x}$ whereas a second order rule reduces the error by $\frac{1}{x^2}$.

(stable) manifolds since the policy function cannot be computed at the turning points. In addition, the polynomial bases and therefore the computation costs grow exponentially when the dimension of the problem increases. To avoid this "curse of dimensionality", a special complete polynomial basis is chosen. But then also the dimension of the basis grows polynomially compared to the relaxation algorithm, where the cost of computation grows only quadratically.

For the time elimination method part of the same critique holds. With multi-dimensional stable manifolds the policy functions cannot be computed at the turning points and therefore this procedure cannot be applied.

4 An illustrative application

The relaxation procedure is employed to investigate the transition process of the well-known R&D-based semi-endogenous growth model of Jones (1995). This model is chosen since it implies a two-dimensional stable manifold with the potential of non-monotonic adjustments. Moreover, for standard calibrations the two stable eigenvalues differ drastically (by about a factor of ten) and, hence, usual procedures are inappropriate to solve the underlying dynamic system.

4.1 The Jones model

As in Jones (1995), the focus here is on the market solution. The final-output technology is given by $Y = \alpha_F(\phi L)^{\sigma_L} \int_0^A x(i)^{1-\sigma_L} di$, where Y denotes final output, ϕ the share of labor allocated to final-output production, $x(i)$ the amount of differentiated capital goods of type i, A the number of differentiated capital goods, α_F a constant overall productivity parameter and σ_L the elasticity of labor in final-output production. Noting the general symmetry among $x(i)$ and using the definition of aggregate capital $K := Ax$, the final-output technology can be written as $Y = \alpha_F(A\phi L)^{\sigma_L} K^{1-\sigma_L}$. The R&D technology is $\dot{A} = J = \alpha_J A^{\eta_A}[(1-\phi)L]^{\eta_L}$ with $\eta_L := \eta_L^p + \eta_L^e$,

$\eta_L^p = 1$,$-1 < \eta_L^e < 0$, where $\dot{A} := dA/dt$, α_J denotes a constant overall productivity parameter, η_A the elasticity of technology in R&D and η_L the elasticity of labor in R&D.

To simulate the transition process, one needs the complete dynamic system governing the evolution of the economy under study. Moreover, we conduct an adjustment of scale such that the long-run levels of all endogenous variables are constant. This dynamic system can be summarized as follows:[12]

$$\dot{k} = y - c - \delta k - \beta_K nk \tag{3}$$

$$\dot{a} = j - \beta_A nk \tag{4}$$

$$\dot{c} = \frac{c}{\gamma}[r - \delta - \rho - (1-\gamma)n] - \beta_K nc \tag{5}$$

$$\dot{v}_a = v_a[r - (\beta_K - \beta_A)n] - \pi \tag{6}$$

$$\frac{\sigma_L y}{\phi} = v_a \frac{\eta_L^p j}{1-\phi} \tag{7}$$

with $y = \alpha_F (a\phi)^{\sigma_L} k^{1-\sigma_L}$, $j = \alpha_J a^{\eta_A}(1-\phi)^{\eta_L}$, $r = \frac{(1-\sigma_L)^2 y}{k}$, $\pi = \frac{\sigma_L(1-\sigma_L)y}{a}$, $\beta_K = \frac{1-\eta_A+\eta_L}{1-\eta_A}$, $\beta_A = \frac{\eta_L}{1-\eta_A}$. The scale-adjusted variables are defined by $y := Y/L^{\beta_K}$, $k := K/L^{\beta_K}$, $c := C/L^{\beta_K}$, $a := A/L^{\beta_A}$, $j := J/L^{\beta_A}$ and $v_a := v/L^{\beta_K - \beta_A}$. The (unique) stationary solution of this dynamic system corresponds to the (unique) balanced growth path of the economy expressed in original variables.

Equations (3) and (4) are the equations of motion of (scale-adjusted) capital and technology, (5) is the Keynes-Ramsey rule of optimal consumption c, (6) shows capital market equilibrium with v_a denoting the (scale-adjusted) price of blueprints and (7) determines the privately efficient allocation of labor across final-output production and R&D.

[12] For a detailed derivation of the dynamic system for the general R&D-based non-scale growth model see Steger (2005).

4.2 Investigation of the transition process

The objective is to solve the four-dimensional system of differential equations (3) - (6), taking into account the static equation (7), which must hold at all points in time. The steady state is a saddle point with a two-dimensional stable manifold. Since the steady state can be determined numerically only, the algorithm computes the steady state of the system first by applying a Newton algorithm. The choice of $k(0) = k_0$ and $a(0) = a_0$ as initial boundary conditions is obvious since k and a are the state variables. Again, there is some freedom when it comes to the determination of boundary conditions. We have set the RHS of equations (5) and (6) equal to zero. Moreover, we choose once more, as an initial guess, all variables to be constant at their steady state values. This always leads to quick convergence indicating that the procedure is relatively robust with respect to the initial guess.

The transition process considered below results from a combination of two simultaneous shocks. Specifically, it is assumed that the overall productivity parameter in the production function for final output α_F increases from 1.0 to 1.3, while the overall productivity parameter in the production function for new ideas α_J decreases from 1.0 to 0.9. This shock was chosen to demonstrate that the adjustment can be non-monotonic (as can be recognized by inspecting Fig. 2 (vi), for instance) and therefore the policy functions cannot be computed at certain points with conventional methods.[13] Figure 2 gives a summary of the adjustment process. The plots (i) to (iii) show the time path of the jump variables c, ϕ, v_a, plots (iv) and (v) display the time path of the state variables k and a, while plot (vi) contains the projection of the adjustment trajectory into the (k,a)-plane.

Several aspects are worth being noticed: (1) The transition process shows a pronounced non-monotinicity for c and k. This overshooting pattern in scale-adjusted variables implies that the instantaneous growth rate of the

[13] The set of parameters used for simulation is: $\sigma_L = 0.6$, $\sigma_K = 0.4$, $\delta = 0.05$, $n = 0.015$, $\eta_A = 0.6$, $\eta_L = 0.5$, $\eta_L^p = 0.6$, $\rho = 0.04$ and $\gamma = 1$.

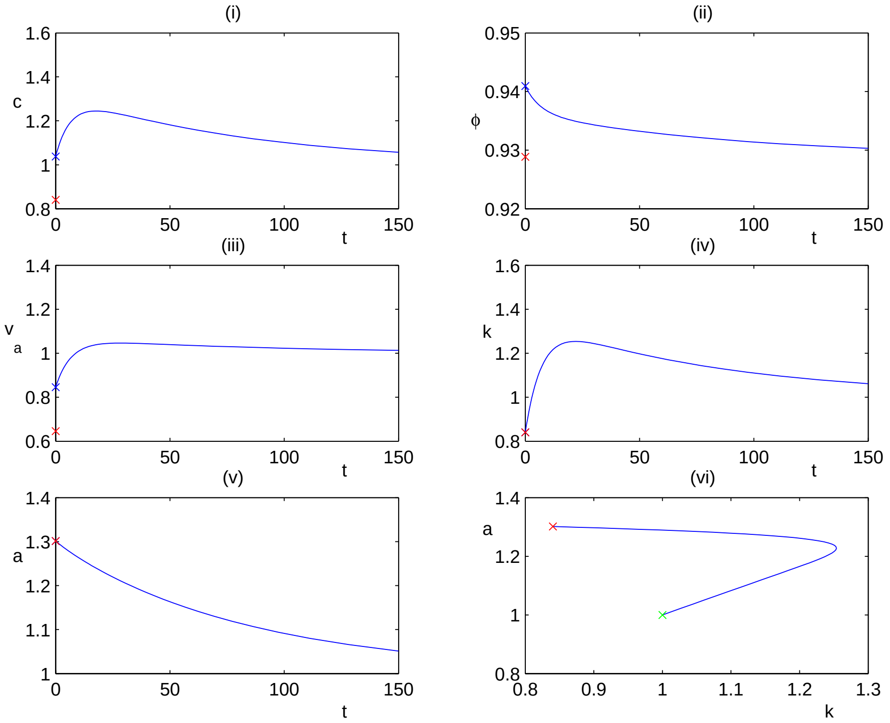

Figure 2: Summary of the transition of the Jones (1995) model

respective original variable is initially above the long-run growth rate, then undershoots and finally converges to the long-run value. (2) The (average) speed of convergence appears to be fairly low with half-lifes of more than 50 years. This observation underlines the importance of the analysis of transitional dynamics. (3) The intersectoral allocation variable ϕ first jumps up [indicated by the crosses in plot (ii)] and then converges to the initial long-run value.

5 Summary and conclusion

In this paper, we propose the relaxation procedure as a powerful and efficient procedure to investigate the transition process of dynamic macroeconomic models. At a very general level, this procedure has two main advantages: First, it is simpler than most other procedures. Second, and more importantly, the relaxation procedure can easily deal with complex dynamic systems for which standard procedures are generally inadequate. More specifically, the procedure can readily handle dynamic systems which are characterized by multi-dimensional stable manifolds (with the potential of non-monotonic adjustment patterns) and strongly differing (stable) eigenvalues. It is important to notice that such dynamic systems are not at all special cases but arise quite frequently from a large number of widely employed growth models.

As an example, the relaxation procedure has been used to investigate the transition process of the well-known Jones (1995), which represents one of the basic workhorses in modern growth theory. Usual procedures turn out to be inadequate for the analysis of this model. This is probably the reason for the fact that there are only few studies which take the adjustment process of this or related models into account.[14]

6 Appendix

In this section we go through some details of the algorithm. Consider a system of $\tilde{N}$ differential equations on an open set in $\mathbb{R}^N$, with $\tilde{N} \leq N$. Let $\tilde{x}$ be the vector of those components of the full vector $x \epsilon \mathbb{R}^N$ affected by f.

$$\frac{d\tilde{x}}{dt} = f(t, x) \ , \quad f : \mathbb{R}_+ \times \mathbb{R}^N \rightarrow \mathbb{R}^{\tilde{N}}$$

[14] To the best of our knowledge, Papageorgiou and Pérez-Sebastián (2003) is the only study which, using the projection method of Judd (1992), simulates the adjustment process of an (extended) non-scale R&D-based growth model.

If $\tilde{N}$ is strictly smaller than N the differential equations are to be supplemented by $N - \tilde{N}$ equations x has to satisfy at any time.

$$0 = g(t, x) \ , \quad g : \mathbb{R}_+ \times \mathbb{R}^N \to \mathbb{R}^{N-\tilde{N}}$$

Boundary conditions are supposed to be given in form of n_1 initial conditions and n_2 final conditions. For the solution to be well determined we need $n_1 + n_2$ to equal $\tilde{N}$. Finally, it is convenient to denote the codimension $N - \tilde{N}$ of the manifold given by $g(t, x) = 0$ by n_3. Summing up we have

n_1	initial conditions	
n_2	final conditions	with $n_1 + n_2 + n_3 = \tilde{N} + n_3 = N$
n_3	running equations	

For convenience, we rescale the time range $\mathbb{R}_+$ by introducing a new time parameter τ running from 0 to 1

$$\tau = \nu t / (1 + \nu t)$$

In terms of τ we get an equivalent differential-algebraic system

$$\begin{aligned} \frac{d\tilde{x}}{d\tau} &= \xi(\tau, x) = f(\tfrac{\tau}{\nu(1-\tau)}, x) \,/\, \nu(1-\tau)^2 \\ 0 &= \phi(\tau, x) = g(\tfrac{\tau}{\nu(1-\tau)}, x) \end{aligned} \tag{8}$$

Define a mesh of M points in (transformed) time τ by $T = \{\tau_1, \ldots, \tau_M\}$. Along the mesh, the dependent variable x falls into a list of vectors. To avoid confusion we denote it by $y = \{y_1, \ldots, y_M\}$ where y_k is the value of x at τ_k. We use the midpoint of each interval (τ_k, τ_{k+1}) for the discretization of the differential equation

$$\tilde{y}_{k+1} - \tilde{y}_k = (\tau_{k+1} - \tau_k)\, \xi(\bar{\tau}_k, \bar{y}_k) \quad \text{for } k = 1, \ldots, M-1 \tag{9}$$

where $\bar{\tau}_k = (\tau_k + \tau_{k+1})/2$ and $\bar{y}_k = (y_k + y_{k+1})/2$. An element of this sequence of difference equations yields an $\tilde{N}$-dimensional error function $H : ([0, \ldots, 1] \times \mathbb{R}^N)^2 \to \mathbb{R}^{\tilde{N}}$

$$H(\tau_k, y_k, \tau_{k+1}, y_{k+1}) = \tilde{y}_{k+1} - \tilde{y}_k - (\tau_{k+1} - \tau_k)\xi(\bar{\tau}_k, \bar{y}_k)$$

Note that the matrix of partial derivatives of H with respect to y_k and y_{k+1} differ only in their derivatives of $\tilde{y}_{k+1}$ and $\tilde{y}_k$, respectively, and this is plus or minus the identity matrix of dimension $\tilde{N}$.

Let B denote the initial conditions

$$B : \mathbb{R}^N \rightarrow \mathbb{R}^{n_1} \, ,$$

F denote the final conditions

$$F : \mathbb{R}^N \rightarrow \mathbb{R}^{n_2}$$

and let C denote the running conditions

$$C : [0, \ldots, 1] \times \mathbb{R}^N \rightarrow \mathbb{R}^{n_3}$$

All together this defines a system of equations in $y = (y_1, \ldots, y_M) \epsilon \mathbb{R}^{N \cdot M}$ given a mesh $\tau = (\tau_1, \ldots, \tau_M) \epsilon \mathbb{R}^M$, and we are looking for a root of this system.

For the description of the algorithm it is convenient to list the equations according to the unknown vectors y_k involved. We start with the initial conditions which only involve y_1 and end with the equations which only involve y_M. Ordered this way the system can be seen as a system of $M+1$ vector equations $E_0(y), \ldots, E_M(y)$. The first subsystem $E_0(y)$ depends only on y_1 and consists of n_1 initial conditions. The intermediate subsystems $E_k(y)$ for $k = 1, \ldots, M-1$ depend on y_k and y_{k+1} and are of dimension N. Each of these subsystems begins with n_3 running conditions and is completed by $n_1 + n_2$ difference equations. The last subsystem $E_M(y)$ depends on y_M and consists of n_3 interior conditions together with n_2 final conditions. It has dimension $n_2 + n_3$.

$$E(y) \equiv \begin{pmatrix} E_0(y) \\ \vdots \\ E_k(y) \\ \vdots \\ E_M(y) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} B(y_1) \end{pmatrix} \\ \vdots \\ \begin{pmatrix} C(y_k) \\ H(y_k, y_{k+1}) \end{pmatrix} \\ \vdots \\ \begin{pmatrix} C(y_M) \\ F(y_M) \end{pmatrix} \end{pmatrix} \tag{10}$$

Each step of the Newton algorithm applied to $E(y) = 0$ computes a change Δy by solving the linear equation

$$D_y E(y) \cdot \Delta y = -E(y)$$

Due to the ordering of subsystems E this equation is of following form:

$$\begin{pmatrix} S^{0,R} & & & & & \\ S^{1,L} & S^{1,R} & & & 0 & \\ & S^{2,L} & S^{2,R} & & & \\ & & & \ddots & & \\ & 0 & & & S^{M-1,L} & S^{M-1,R} \\ & & & & & S^{M,L} \end{pmatrix} \begin{pmatrix} \Delta y_1 \\ \vdots \\ \Delta y_M \end{pmatrix} = \begin{pmatrix} -E_0(y) \\ \vdots \\ -E_M(y) \end{pmatrix} \tag{11}$$

All $S^{k,L}$ and $S^{k,R}$ are Jacobian matrices defined by

$$S^{k,L} = \frac{\partial E_k(y)}{\partial y_k}\,, \quad \text{and} \quad S^{k,R} = \frac{\partial E_k(y)}{\partial y_{k+1}}$$

The upper left matrix $S^{0,R}$ has n_1 rows and the lower right matrix $S^{M,L}$ only $n_3 + n_2$, whereas all other matrices $S^{k,L}$ and $S^{k,R}$, resp, are $N \times N$. Hence, the system is not overdetermined. The solution Δy can be computed by a specialized Gaussian algorithm. This algorithm starts in the upper left corner of the matrix and works downward block by block to the lower right corner. The result is a system in upper triangular form with a sequence of $N \times (n_2 + n_3)$ non-zero blocks above the diagonal. Finally the vector Δy can computed from bottom to top. To be more precise:

step 0: Diagonalize the first n_1 columns of $S^{0,R}$.

step k, $k=1,\ldots,M-1$*:* Eliminate the first $N-n_1$ columns of $S^{k,L}$;
diagonalize the remainder of $S^{k,L}$ together
with the first $N-n_1$ columns of $S^{k,R}$.

step M*:* Eliminate the first $N-n_1$ columns of $S^{M,L}$;
Diagonalize the remainder of $S^{M,L}$

step $M+k$, $k=1,\ldots,M$*:* Solve for Δy_{M+1-k}.

The Newton algorithm refines the current guess of y by adding Δy or a fraction of this vector to y. The algorithm stops if the error E is sufficiently small according to an appropriate norm.

References

BARRO, R. J., AND X. SALA-I-MARTIN (2004): *Economic Growth.* MIT-Press, Cambridge, Massachusetts.

BENHABIB, J., AND R. PERLI (1994): "Uniqueness and Indeterminacy: On the Dynnamics of Endogenous Growth," *Journal of Economic Theory*, 63, 113–142.

BRUNNER, M., AND H. STRULIK (2002): "Solution of Perfect Foresight Saddlepoint Problems: A Simple Method and Applications," *Journal of Economic Dynamics and Control*, 26, 737–753.

CASS, D. (1965): "Optimum Growth in an Aggregate Model of Capital Accumulation," *Review of Economic Studies*, 32, 233–240.

EICHER, T. S., AND S. J. TURNOVSKY (1999): "Convergence in a Two-Sector Nonscale Growth Model," *Journal of Economic Growth*, 4, 413–428.

JONES, C. I. (1995): "R&D-Based Models of Economic Growth," *Journal of Political Economy*, 103(3), 759–784.

JUDD, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58, 410–452.

——— (1998): *Numerical Methods in Economics.* MIT Press.

KOOPMANS, T. C. (1965): "On the Concept of Optimal Economic Growth," In: *The Econometric Approach to Development Planning,* Amsterdam: North Holland.

LUCAS, R. E. JR.. (1988): "On the Mechanics of Economic Development," *Journal of Monetary Economics*, 22, 3–42.

Mercenier, J., and P. Michel (1994): "Discrete-Time Finite Horizon Approximation of Infinite Horizon Optimization Problems with Steady-State Invariance," *Econometrica*, 62(3), 635–656.

Mulligan, C., and X. Sala-i-Martin (1991): "A Note on the Time Elimination Method for Solving Recursive Growth Models," *Technical Working Paper No. 116*.

——— (1993): "Transitional Dynamics in Two-Sector Models of Endogenous Growth," *Quarterly Journal of Economics*, 108(3), 737–773.

Papageorgiou, C., and F. Pérez-Sebastián (2002): "Human Capital and Convergence in a Non-Scale R&D Growth Model," *Departmental Working Papers, Department of Economics, Louisiana State University*, 2002-10.

Press, W., B. Flannery, S. Teukolsky, and W. Vetterling (1989): *Numerical Recipes in Pascal*. Cambridge University Press.

Ramsey, F. (1928): "A Mathematical Theory of Saving," *Economic Journal*, 38, 543–559.

Romer, P. M. (1990): "Endogenous Technological Change," *Journal of Political Economy*, 98, S71–S102.

Steger, T. M. (2005): "Welfare Implications of Non-Scale R&D-based Growth Models," *Scandinavian Journal of Economics*, forthcoming.

Temple, J. (2003): "The Long-Run Implications of Growth Theories," *Journal of Economic Surveys*, 17(3), 497–510.