Improving tatonnement methods for solving heterogeneous agent models*

Alexander Ludwig†
MEA, University of Mannheim

First Version: January 10, 2003
This Version: February 27, 2004

Abstract

This paper develops a globally convergent algorithm which modifies standard block Gauss-Seidel iterations used by tatonnement methods for solving large scale deterministic heterogeneous agent models. It is shown that the restrictions on the structure of the Jacobi matrix implicit in any such first-order iterative method can easily be relaxed for these models. Instead of relying on *ad hoc* and fixed dampening factors, standard Quasi-Newton methods can be used to determine the exact Jacobi matrix for steady state calculations and to update its elements by Broyden's method as the iteration proceeds. By transforming variables such that they are constant in the steady states, very few elements of the Jacobian have to be determined. For transition calculations the resulting steady state Jacobi matrix can be used as an approximation of the true transition Jacobi matrix. This extension of standard Gauss-Seidel iterations is shown to considerably improve convergence both in terms of speed as well as robustness relative to an *ad hoc* choice of fixed dampening factors. In addition, the relative advantage of the modified algorithm increases in the number of state variables of the model. The algorithm is particularly attractive since it is easy to implement - it only augments conventional and intuitive tatonnement iterations by standard numerical methods.

JEC classification: C63, C68, E13
Keywords: OLG models; Gauss-Seidel iterations

*I thank Axel Börsch-Supan, Wouter Denhaan, Ken Judd, Michel Juillard, Daniel Schunk, Gabriele Steidl and Joachim Winter as well as several seminar participants at the University of Mannheim for helpful comments.

†MEA, University of Mannheim; L13,17; 68131 Mannheim; Germany. Phone: +49-621-181-1866, Fax:+49-621-181-1863, email: alexander.ludwig@mea.uni-mannheim.de

1 Introduction

This paper modifies Gauss-Seidel iterations used to solve large-scale deterministic heterogeneous agent models. Such models are increasingly used for analysis of economic questions. Standard procedures use domain truncation methods and resort to general methods for solving large systems of (nonlinear) equations. Three types of such conventional solution methods can be distinguished: (i) Newton based methods such as the L-B-J method[1], (ii) the Fair-Taylor (extended path) method[2] and (iii) tatonnement methods[3], see (Judd, Kubler, and Schmedders 2000). Let $Q = \{q_i\}_{i=1}^m$ where $q_i = \{q_{i,t}\}_{t=0^T} \forall i$ be sequences of m endogenous economic variables such as factor supplies starting from intital conditions up to some fixed T. Conventional methods have in common that they solve the model for each $q_{i,t}$. More recently, Judd (2002) has proposed an alternative route. Rather than explicitely solving for each $q_{i,t}$, Judd suggests to use prior information about the time path of q_i and to approximate it by a functional form $q_{i,t} = \Phi_i(t, a)$ with a low-dimensional parameter vector a. Judd's method can be regarded as a more modern approach.

This paper is concerned with traditional methods and suggests a simple procedure for improvement of tatonnement methods. While L-B-J and Fair-Taylor methods regard any perfect foresight general equilibrium model simply as a system of (non-linear) equations including aggregate and disaggregate variables and iterate over this entire system, tatonnement methods break variables into *aggregate* and *disaggregate* variables. Outer loops then proceed via block Gauss-Seidel algorithms using aggregate variables only, while inner loops are used to solve for disaggregate variables in a disaggregate model. Outer loops work as follows: Let $P = S^{-1}(Q)$ denote a sequence of factor prices corresponding to the sequence of factor supplies Q. Equilibrium of tatonnement methods is defined sequentially as a fixed point, $Q = D(S^{-1}(Q))$, where D denotes the demand function. $S^{-1}(Q)$ and $D(P)$ are solved by inner loops of the disaggregate model and by aggregating individual decisions. The fixed point problem suggests to execute the iteration $Q^{k+1} = D(P^{k+1}) = D(S^{-1}(Q^k))$, which is the familiar *hog-cycle* process, where k is the iteration number. Since P^{k+1} and not P^k is used to form an update of Q^{k+1} the iterations performed are nonlinear Block-Gauss-Seidel iterations. Depending on the relative shape of S and D such iterations may however not converge. These convergence problems force researchers to rely on *ad hoc* dampening factors such that the iteration rewrites as $Q^{k+1} = Q^k - w(Q^k - D(S^{-1}(Q^k))) = Q^k - w(Q^k - \tilde{Q}^k)$, where w is the dampening factor.[4] Dampening factors play a similar role as adaptive expectations in the cobweb model. Such modifications of standard Gauss-Seidel iterations have been

[1] See Laffargue (1990), Boucekkine (1995), Juillard (1996) and Juillard et al. (1998).

[2] See Fair and Taylor (1983).

[3] See Auerbach and Kotlikoff (1987).

[4] The relative weight w attached to Q^k and $\tilde{Q}^k$ respectively

referred to as *fast* Gauss-Seidel (FGS) iterations (Hughes Hallet 1984). Since these methods only use values of $D(S^{-1}(Q^k))$ to solve the fixed point problem, they belong to the class of first-order iterative methods. While intuitive, convergence of these methods is slow (linear at best) and they may not converge at all even after various dampening factors have been tried out.

As an alternative to using *ad hoc* dampening factors, *optimal* dampening factors can be determined. However, they are difficult to determine even for linear models, see, e.g., Hagemann and Young (1981) and Judd (1999). Therefore, various adaptive techniques to update dampening factors as the iteration proceeds have been suggested in the literature (Hagemann and Young 1981; Hughes Hallet 1982). The approach developed in this paper instead departs from the well established fact that first-order iterations can be regarded as approximate Newton methods. Hence equations like $Q^{k+1} = Q^k - w(Q^k - D(S^-1(Q^k)))$ can be written as $Q^{k+1} = Q^k - wG(Q^k)$ where $G(Q^k)$ is a set of simultaneous non-linear equations and Q^k is the root of these equations. The approximation of the true Jacobi matrix, J, is $\hat{J} = w^{-1}I$, where I is an identity matrix. One obvious way for improvement of standard Gauss-Seidel iterations would be to perform Gauss-Seidel-Newton iterations and to determine the elements of J in each iteration step, see Ortega and Rheinboldt (2000, chapter 7.4). An alternative is to use Gauss-Newton methods which approximate the objective by a quadratic form of $G(Q^k)$, see Ortega and Rheinboldt (2000, chapter 8.5). However, if T becomes large such approaches become too costly.[5]

The approach followed in this paper is much simpler and hence easier to implement. It exploits that first, the number of state variables denoted by m will in general be relatively small. For a typical one sector closed economy general equilibrium growth model with endogenous capital formation and endogenous labor supply $m = 2$. Second, the elements of the true Jacobi matrix will be constant in the steady state if the variables in Q are defined such that they are constant in the steady state. As a consequence, the Jacobi matrix can be written as a Kronecker product of the inverse of a low-dimensional matrix, $W_{(m\times m)}$, and an identity matrix, $J = W^{-1} \otimes I$. The - generally few - m^2 elements of matrix W can easily be determined by Newton's method in (fast) steady state calculations. In addition, they can be used as good starting values for an approximate Jacobi matrix in transition calculations. It is suggested to update W by Broyden's method during both steady state and transition iterations. Asymptotically, the rate of convergence therefore increases from (at best) linear to super-linear. Robustness of the iterations can further be achieved by using standard line search methods like the familiar backtracking algorithm. Since an approximate Jacobian is used for almost all iteration steps, the algorithm will be referred to as Gauss-Seidel-Quasi-Newton (GSQN) method. The attractiveness of GSQN stems from its simplicity: the intuitive appeal and relatively low computational demands of

[5]Even though J is a sparse matrix, see below.

tatonnement iterations is combined with standard Newton based methods that are implementable at little extra costs.

As an illustration of the procedure, a large-scale multi-country overlapping generations (OLG) model with endogenous labor supply is used such that the number of states, the dimension m, can be increased from $m = 1$ (closed economy model with exogenous labor supply) to $m = 4$ (three country model with endogenous labor supply). In order to compare the relative performance of both algorithms (FGS and GSQN), the model is simulated under various combinations of parameters. Previewing results, the simple modifications suggested in this paper quite considerably improve convergence compared to FGS. For the latter, only relatively low values for the dampening factor such as $w = 0.1$ lead to almost sure convergence. In fact, even for this choice of the dampening factor, standard Gauss-Seidel iterations are shown not converge for about five percent of cases. For higher values of w, robustness of FGS decreases sharply: for $w = 0.3$, FGS does not converge for about 50 to 60 percent of cases. In contrast, GSQN always converges. For transition calculations, average convergence speeds of GSQN are about double than those of FGS with $w = 0.1$ when $m = 1$ and about seven times higher when $m = 4$. Hence, GSQN considerably improves convergence both in terms of speed as well in terms of robustness relative to standard FGS.

The paper proceeds as follows: Section 2 develops the suggested modification of the conventional Gauss-Seidel algorithm, GSQN. Section 3 briefly describes the OLG model to be used for illustration and section 4 compares the relative performance of FGS and GSQN. Section 5 concludes and proposes some directions for further research.

2 The Gauss-Seidel-Quasi-Newton algorithm

2.1 General considerations

Let $Y = \{y_i\}_{i=1}^{n}$ where $y_i = \{y_{i,t}\}_{t=0}^{T} \forall i$ be a list of endogenous variables. y_i includes wage rates and interest rates as aggregate variable (a_i) as well as disaggregate variables (d_i) such as consumption and assets of individual households, etc.. Further, let $Z = \{z_i\}_{i=1}^{l}$ where $z_i = \{z_{i,t}\}_{t=0}^{T} \forall i$ be a list of exogenous variables such as population data of cohorts living at time t. Deterministic perfect foresight heterogeneous agent models can be written in a general form as

$$\begin{aligned} F(Y, Z) &= 0 \\ y_{i,0} = \bar{y}_{i,0},\ i = 0, 1, ..., n_i,\ n_i < n & \\ y_{i,t} \text{ bounded for all } i & \end{aligned} \tag{1}$$

where $F(Y, Z)$ are nT non-linear functions that represent equilibrium. Since Z are exogenous they are dropped from here on. The equations in (1) include Euler equations, asset accumulation equations, market clearing conditions as well as

any other equations that define equilibrium. Domain trunction has been applied in equation (1) since the time horizon starts in period $t = 0$ departing from some initial conditions and is restricted to T. Traditional solution methods, such as Fair-Taylor and L-B-J, try to directly solve systems of equations such as (1) for each element in $y_{i,t}$ by Gauss-Seidel iterations or Newton based methods respectively.

In contrast, conventional tatonnement methods break the system of equations in (1) into a factor *supply* and a factor *demand* model. Both require inner loops to solve and to aggregate individual decision problems. Let $A = (a_1, a_2, ...)$ be aggregate variables and $B = (b_1, b_2, ...)$ be dis-aggregate variables. Further split A as $A = (Q, P, R)$ where Q are aggregate factor supply variables such as the aggregate capital stock and aggregate labor supply of an economy and P are the associated factor price variables such as aggregate interest and wage rates and let $Q = (q_1, ..., q_m) \in \mathbb{R}^{mT}$ as well as $P = (p_1, ..., p_m) \in \mathbb{R}^{mT}$. R are all other aggregate variables such as aggregate consumption and savings which are not necessary to define equilibrium since they are functions of Q, P and B and will therefore not be considered from here on. A perfect foresight OLG model of the form given in equation (1) can be re-written as

$$\begin{aligned} \textit{Supply model: } P &= S^{-1}(Q) \\ \textit{Demand model: } Q &= D(P) \\ \textit{Aggregators: } S^{-1}(Q) &= \Psi^S(s^{-1}(Q, B)) \\ \text{and } D(P) &= \Psi^D(d(P, B)), \end{aligned} \tag{2}$$

where S^{-1} is the inverse aggregate supply function and D is the aggregate demand function. The aggregators are only used to indicate that aggregate demand and supply functions are derived from individual decisions of heterogeneous agents and will be ignored from here on. Since $P = S^{-1}(Q)$, there are m endogenous variables that are sufficient to define equilibrium. Combining the first two lines of equation (2) leads to the definition of equilibrium of a heterogeneous agent model as a fixed point by

$$Q = D(S^{-1}(Q)) \tag{3}$$

This suggests to use standard (block) Gauss-Seidel iterations to solve for Q and hence to iterate over the system

$$\begin{aligned} P^{k+1} &= S^{-1}(Q^k) \\ Q^{k+1} &= D(P^{k+1}) \end{aligned} \tag{4}$$

which can be more concisely written as

$$Q^{k+1} = D(S^{-1}(Q^k))$$

It is well-known that such methods may not converge. An alternative approach is to first transform equation (3) into a root-finding problem as

$$G(Q) = Q - H(Q) = Q - D(S^{-1}(Q)) = 0, \tag{6}$$

where $H(\cdot)$ is introduced as a short-cut notation for $D(S^{-1})(\cdot)$.

Applying a first-order Taylor series approximation to equation (6) leads to the familiar Newton updating formula of Q given by

$$Q^{k+1} = Q^k - J^{-1}[Q^k]G(Q^k), \tag{7}$$

where $J[Q^k]$ is the Jacobi matrix of the system of equations in (6) evaluated at Q^k.

Recall that $Q = \{q_i\}_{i=1}^m$, where $q_i = \{q_{i,t}\}_{t=1}^T$. Due to the specific form of the functions $G = \{g_i(Q)\}_{i=1}^m$ where $g_i(Q) = \{g_{i,t}(Q)\}_{t=0}^T$ in equation (6), the elements of the Jacobi matrix given by

$$J[Q^k] = \begin{bmatrix} \frac{\partial g_{1,0}(Q^k)}{\partial q_{1,0}^k} & \frac{\partial g_{1,0}(Q^k)}{\partial q_{1,1}^k} & \dots & \frac{\partial g_{1,0}(Q^k)}{\partial q_{1,T}^k} & \frac{\partial g_{1,0}(Q^k)}{\partial q_{2,0}^k} & \frac{\partial g_{1,0}(Q^k)}{\partial q_{2,1}^k} & \dots & \frac{\partial g_{1,0}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \frac{\partial g_{1,1}(Q^k)}{\partial q_{1,0}^k} & \frac{\partial g_{1,1}(Q^k)}{\partial q_{1,1}^k} & \dots & \frac{\partial g_{1,1}(Q^k)}{\partial q_{1,T}^k} & \frac{\partial g_{1,1}(Q^k)}{\partial q_{2,0}^k} & \frac{\partial g_{1,1}(Q^k)}{\partial q_{2,1}^k} & \dots & \frac{\partial g_{1,1}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial g_{2,0}(Q^k)}{\partial q_{1,0}^k} & \frac{\partial g_{2,0}(Q^k)}{\partial q_{1,1}^k} & \dots & \frac{\partial g_{2,0}(Q^k)}{\partial q_{1,T}^k} & \frac{\partial g_{2,0}(Q^k)}{\partial q_{2,0}^k} & \frac{\partial g_{1,0}(Q^k)}{\partial q_{2,1}^k} & \dots & \frac{\partial g_{2,0}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \frac{\partial g_{2,1}(Q^k)}{\partial q_{1,0}^k} & \frac{\partial g_{2,1}(Q^k)}{\partial q_{1,1}^k} & \dots & \frac{\partial g_{2,1}(Q^k)}{\partial q_{1,T}^k} & \frac{\partial g_{2,1}(Q^k)}{\partial q_{2,0}^k} & \frac{\partial g_{1,1}(Q^k)}{\partial q_{2,1}^k} & \dots & \frac{\partial g_{2,1}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{8}$$

can be re-written as

$$\begin{bmatrix} 1 - \frac{\partial h_{1,0}(Q^k)}{\partial q_{1,0}^k} & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{1,1}^k} & \dots & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{1,T}^k} & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{2,0}^k} & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{2,1}^k} & \dots & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{2,T}^k} & \dots \\ -\frac{\partial h_{1,1}(Q^k)}{\partial q_{1,0}^k} & 1 - \frac{\partial h_{1,1}(Q^k)}{\partial q_{1,1}^k} & \dots & -\frac{\partial h_{1,1}(Q^k)}{\partial q_{1,T}^k} & -\frac{\partial h_{1,1}(Q^k)}{\partial q_{2,0}^k} & -\frac{\partial h_{1,1}(Q^k)}{\partial q_{2,1}^k} & \dots & -\frac{\partial h_{1,1}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -\frac{\partial h_{2,0}(Q^k)}{\partial q_{1,0}^k} & -\frac{\partial h_{2,0}(Q^k)}{\partial q_{1,1}^k} & \dots & -\frac{\partial h_{2,0}(Q^k)}{\partial q_{1,T}^k} & 1 - \frac{\partial h_{2,0}(Q^k)}{\partial q_{2,0}^k} & -\frac{\partial h_{1,0}(Q^k)}{\partial q_{2,1}^k} & \dots & -\frac{\partial h_{2,0}(Q^k)}{\partial q_{2,T}^k} & \dots \\ -\frac{\partial h_{2,1}(Q^k)}{\partial q_{1,0}^k} & -\frac{\partial h_{2,1}(Q^k)}{\partial q_{1,1}^k} & \dots & -\frac{\partial h_{2,1}(Q^k)}{\partial q_{1,T}^k} & -\frac{\partial h_{2,1}(Q^k)}{\partial q_{2,0}^k} & 1 - \frac{\partial h_{1,1}(Q^k)}{\partial q_{2,1}^k} & \dots & -\frac{\partial h_{2,1}(Q^k)}{\partial q_{2,T}^k} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{9}$$

which can be partitioned as

$$\begin{bmatrix} J_{1,1}^k & J_{1,2}^k & \dots & J_{1,m}^k \\ J_{2,1}^k & J_{2,2}^k & \dots & J_{2,m}^k \\ \dots & \dots & \dots & \dots \\ J_{m,1}^k & J_{m,2}^k & \dots & J_{m,m}^k \end{bmatrix} \tag{10}$$

according to the endogenous variables in q_i. Hence, each sub-matrix $J_{i,j}$, for $i, j = 1, ..., m$ is of dimension $T \times T$ with each element given by

$$\begin{aligned} J_{i,j,t,\Delta_t} \quad &= \quad \begin{cases} 1 - \frac{\partial h_{i,t}(Q^k)}{\partial q_{j,t+\Delta_t}^k} & \text{for } \Delta_t = 0 \text{ and } i = j \\ -\frac{\partial h_{i,t}(Q^k)}{\partial q_{j,t+\Delta_t}^k} & \text{else} \end{cases} \\ \text{for} \quad & t = 0, ..., T, \\ \text{where} \quad & -t \leq \Delta_t \leq T - t \end{aligned} \tag{11}$$

Due to the heterogeneous agent structure of the model, $-\frac{\partial h_{i,t}(Q^k)}{\partial q^k_{j,t+\Delta_t}} = 0$ for Δ_t sufficiently large. Hence $J[G(Q^k)]$ is sparse. Despite this it is generally quite costly to determine all elements of the Jacobi matrix $J[G(Q^k)]$ as T becomes large.

Given these computational costs an obvious solution is to approximate the Jacobi matrix. Conventional Gauss-Seidel iterations with one-parameter dampening by factor w iterate on

$$Q^{k+1} = Q^k - wG(Q^k) = Q^k - wI_{(mT\times mT)}G(Q^k) \tag{12}$$

and hence restrict the elements of the true Jacobi matrix $J[Q^k]$ to $w^{-1}I_{(mT\times mT)}$. The restrictions of the iteration matrix wI on the Jacobian can be summarized as follows: first, the iteration matrix is constant across all iteration steps k, second, the difference of the sub-matrices along the diagonal of the partitioned matrix in equation (10), $J_{i,j}$, $i = 1, ...m$, $i = j$, is ignored and third, the off-diagonal elements of the partitioned matrix in equation (10), $J_{i,j}$, $i = 1, ...m$, $i \neq j$, are ignored.

This paper suggests to restrict the Jacobi matrix to be

$$\hat{J}^k = [W^{-1}]^k_{m\times m} \otimes I_{(T\times T)} = \begin{bmatrix} \omega^k_{1,1}I_{(T\times T)} & \omega^k_{1,2}I_{(T\times T)} & ... & \omega^k_{1,m}I_{(T\times T)} \\ \omega^k_{2,1}I_{(T\times T)} & \omega^k_{2,2}I_{(T\times T)} & ... & \omega^k_{2,m}I_{(T\times T)} \\ & ... & & \\ \omega^k_{m,1}I_{(T\times T)} & \omega^k_{m,2}I_{(T\times T)} & ... & \omega^k_{m,m}I_{(T\times T)} \end{bmatrix} \tag{13}$$

where W can be interpreted as an $m \times m$ matrix of multiple dampening factors that vary with the iteration number k. This structure of the Jacobi matrix is similar to the partitioned matrix given in equation (10). It therefore relaxes (some of) the restrictions imposed by the Gauss-Seidel algorithm. The rationale for this form of the Jacobi matrix is derived from a steady state situation of a heterogeneous agent model where the iteration matrix in (13) is the exact Jacobi matrix if the time series in Q, $q_1, ..., q_m$, are transformed such that they are constant in the steady state.

2.2 The steady state

Suppose the variables in Q are defined such that they are constant in the steady state. E.g., q_1 could be a time series of the capital to output ratio and q_2 of the labor supply ratio in a closed economy growth model with endogenous labor supply. Further, note that domain truncation imposes a restriction on the equation system which is mirrored by a Jacobi matrix of finite dimension and hence by the restriction on Δ_t in equation (11) requiring that $-t \leq \Delta_t \leq T - t$. This restriction is invalid if the economy was in steady state forever. For such a model the restriction on Δ_t is $-T_0 - t \leq \Delta_t \leq T_0 - t$, since, as noted above,

$-\frac{\partial h_{i,t}(Q^k)}{\partial q^k_{j,t+\Delta_t}} = 0$ for Δ_t sufficiently large. Further, since the elements of each $\{q_i\}_{i=1}^m$ are constant in the steady state, the starting values in iteration $k = 0$, $\{q_i^0\}_{i=1}^m$, will be constant as well and so will be the partial derivatives in equation (11). Hence, an "infinite dimension" representation of the Jacobi matrix in equation (11) would be

$$J^{T_0}_{i,j,\Delta_t} = \begin{cases} 1 - \frac{\partial h_i(Q^k)}{\partial q^k_{j,\Delta_t}} & \text{for } \Delta t = 0 \text{ and } i = j \\ -\frac{\partial h_i(Q^k)}{\partial q^k_{j,\Delta_t}} & \text{else} \end{cases} \quad \text{where } -T_0 - t \leq \Delta_t \leq T_0 - t \tag{14}$$

and therefore the sub-matrices of the iteration matrix can be written as

$$J^{T_0}_{i,j} = 1 - \sum_{\Delta_t = -T_0 - t}^{T_0 - t} \frac{\partial h_i(Q^k)}{\partial q^k_{j,\Delta_t}} \tag{15}$$

and hence the inverse of matrix W in equation (13) is

$$W^{-1} = \{J^{T_0}_{i,j}\}_{i,j=1}^m = \left\{ 1 - \sum_{\Delta_t = -T_0 - t}^{T_0 - t} \frac{\partial h_i(Q^k)}{\partial q^k_{j,\Delta_t}} \right\}_{i,j=1}^m \tag{16}$$

which gives the exact Jacobi matrix of the Newton iteration in equation (7).

In general, it would be possible to start the iteration with any initial estimate of the Jacobi matrix, $\hat{J}^0$ such as $\hat{J}^0 = wI_{mT \times mT}$ which is just the Gauss-Seidel weight and then to update it via Broyden's multi-variate secant method as[6]

$$\begin{aligned} \hat{J}^{k+1} &= \hat{J}^k + \frac{(\Delta G(Q^k) - J^k \Delta Q^k))(Q^k)'}{(Q^k)' Q^{k-1}}, \text{ where} \\ \Delta G(Q^k) &= G(Q^k) - G(Q^{k-1}) \text{ and} \\ \Delta Q^k &= Q^k - Q^{k-1} \end{aligned} \tag{17}$$

But, since m, the number of endogenous variables, will in general be small and since steady state solutions are fast to compute, it is more reasonable to spend the first m^2 iterations to evaluate all elements of $\hat{J}^0$ by finite difference methods. Since the model is in steady state, these evaluations must only be carried out for each element $\hat{J}_i(Q)$ for $t = t_0$ by finite difference methods as

$$\hat{J}^0_{i,j} = \frac{(g_{i,t_0}(Q + hI_{T \times T})) - g_{i,t}(Q)}{h} \tag{18}$$

where h is some small number. t_0 is the steady state period, hence $t_0 = 1$ for an (artificial) initial steady state or $t_0 = T$ for a final steady state. In contrast, for iterations $k > 0$, it is more reasonable to save these additional m^2 iterations and to use Broyden's method for an update of $\hat{J}^k$ instead. In case the system is divergent, the Jacobian will be recomputed, see below.

[6] Of course, the *Sherman-Morrison* formula for a direct update of $\left[J^k\right]^{-1}$ could also be used but inversion is cheap since m is small.

2.3 The transition

Recall that in general m is small but T is quite large. Therefore, while steady state solutions are fast to compute, transition calculations may take considerable time. Against this background, the general idea of the implementation of GSQN for transition calculations is to use the Jacobi matrix derived during (fast) steady state calculations as an initial approximate Jacobi matrix for transition calculations and to update it by Broyden's method as the iteration proceeds. The exact implementation of the algorithm during transition calculations however depends on the restrictions on the structure of the equation system imposed by (initial and final) steady states or (and) arbitrary initial conditions. Four different models can be distinguished.

- **Model 1** (*initial value problem*): The economy starts from an initial steady state and converges to a final steady state. The final steady state has been calculated.
- **Model 2** (*final value problem*): The economy starts from an initial steady state and converges to a final steady state. The initial steady state has been calculated.
- **Model 3** (*final value problem*): The economy starts from arbitrary initial conditions and converges to a final steady state. The initial conditions are known.
- **Model 4**: The economy starts from an initial steady state and converges to a final steady state. Both steady states have been calculated.

Permanent structural changes are implicit in the definition of all models. However, for temporary changes, the economy starts from the same steady state as it converges to and hence such a specification is nested in model 4 (or model 1).

In terms of equations the four different models can be written as follows. For ease of presentation it is assumed that $m = 1$. Recall that the variables in Q are transformed such that they are constant in the steady state.

- **Model 1**:

$$\begin{aligned} q_1 &= q_2 \\ q_2 &= h_2(Q) \\ q_3 &= h_3(Q) \\ &\dots \\ q_T &= q^{fss}, \end{aligned}$$

  where fss stands for final steady state.

- **Model 2**:

$$\begin{aligned} q_1 &= q^{iss} \\ q_2 &= h_2(Q) \\ q_3 &= h_3(Q) \\ &... \\ q_T &= q_{T-1} \end{aligned}$$

  where iss stands for initial steady state.

- **Model 3**:

$$\begin{aligned} q_1 &= \bar{q}_1 \\ q_2 &= h_2(Q) \\ q_3 &= h_3(Q) \\ &... \\ q_T &= q_{T-1} \end{aligned}$$

- **Model 4**:

$$\begin{aligned} q_1 &= q^{iss} \\ q_2 &= h_2(Q) \\ q_3 &= h_3(Q) \\ &... \\ q_T &= q^{fss} \end{aligned}$$

For models 1 to 3 it is hence assumed that the final (or initial) steady state is calculated during the transition solution while the initial (or final) steady state (or the initial condition) is already known from steady state calculations. For model 4 it is assumed that both steady states were calculated during steady state calculations. For the latter, GSQN is implemented by using the iteration matrix derived during steady state calculations (either initial or final), J^{ss}, throughout all transition iterations. For model 1 (2) the iteration matrix for final (initial) steady state calculations, J^{fss} (J^{iss}), will be used as initial dampening factor matrix and will be updated by Broyden's method using the information contained in $Q^k_{i,t^{ss}} = \{q_{i,t^{ss}}\}_{i=1}^m$ and $G(Q^k_{i,t^{ss}}) = \{g(q_{i,t^{ss}})\}_{i=1}^m$ where $t^{ss} = 1$ ($t^{ss} = T$), i.e., the information contained in the initial (final) steady state period.[7] The updating procedure for model 3 is equivalent to the procedure for model 2 but it requires an

[7]Updating J^k by Broyden's method is not necessary, but using the additional information contained in each iteration step k is more robust than using constant dampening factors throughout.

initial iteration matrix. The latter could be chosen as a scaled identity matrix or could be determined by finite difference methods by evaluating J^0 for time period $t = T$. As before, determining the initial iteration matrix by finite difference methods is recommended. For all models, the Jacobian will be recomputed if the system is divergent, see below.

To summarize, the use of $\hat{J}^0 = \hat{J}^{ss}$ or a Jacobi matrix determined by finite different methods (model 3) as initial iteration matrix and to update it by Broyden's method results in good approximations of the elements of the true Jacobian for the steady state period(s) of transition calculations. It is also a good approximation of the true Jacobian for all other periods. Applying different dampening factors for different time periods is not reasonable since it would create artificial kinks in the time paths Q^{k+1}. Thus, while the Jacobi matrix determined by the suggested method is asymptotically optimal as Q^k approaches Q^{ss} for steady state calculations, it is a good approximation for transition calculations.

2.4 Implementation of Gauss-Seidel-Quasi-Newton iterations

It is well-known that if $G(Q)$ is continuously differentiable over a convex set D containing the equilibrium values Q^* with $G(Q^*) = 0$, then there exists an open set C about Q^* such that equation (7) converges at least linearly from any $Q^0 \in C$. If in addition the Lipschitz condition $\|Q^k - Q^*\| \leq \ d|Q^{k-1} - Q^*\|$ holds for $Q^0 \in C$ and some $d > 0$, the rate of convergence becomes quadratic. However, if the starting values Q^0 are not within C, then Newton iterations such as equation (7) may not be convergent. In order to obtain a globally convergent method, i.e. an iteration that converges from almost any starting value, it is therefore reasonable to augment the Newton iteration by a line search method to get

$$Q^{k+1} = Q^k - s^k \hat{J}^{-1}[Q^k]G(Q^k), \tag{19}$$

where s^k is a standard variable step-size parameter and $\hat{J}$ denotes the Broyden iteration matrix. A fast algorithm for line searches is by backtracking, see e.g., Press et al. (1992). It relies on a quadratic approximation of the (unknown) objective function given by $g(Q^k) = \frac{1}{2}G(Q^k)'G(Q^k)$ and determines a step that minimizes this quadratic approximation. If the resulting step is not acceptable, then the algorithm iterates over a cubic approximation of the objective function until an acceptable step is found.

However, since $\hat{J}^k$ is not the exact Jacobian, it is not guaranteed that the line search algorithm will give a descent step direction. Hence, the Jacobian will be re-initialized (by finite difference methods) in case that the line search algorithm does not return a suitable step (after a maximum of three line search iterations or when reaching a minimum value for s^k). For transition calculations, both line search algorithm and even more re-initializing the Jacobian can by costly in terms

of computational time and should therefore be avoided if possible. Therefore, it has proven useful to first switch to the initial Jacobian, $\hat{J}^0$, in case $g(Q^k)$ increases. If this is not a suitable step, then line search is called. If the line search algorithm does not return a suitable step, then the Jacobian is re-initialized by first-difference methods and the resulting matrix replaces $\hat{J}^0$.

Moreover, it will be useful to re-initialize the Jacobian if the updated Jacobian $\hat{J}^k$ or if ΔQ^k fail to satisfy a number of conditions: (i) if ΔQ^k is too small, (ii) if $\hat{J}^k$ is ill conditioned[8] and (iii) if some of the elements of $\hat{J}^k$ do not satisfy certain criteria reflecting prior knowledge regarding their value. E.g., for the applications considered in section 4, it is required that the diagonal elements of J^k are positive. Conditions (i) and (ii) are standard conditions and condition (iii) would automatically be fixed in the next iteration step by the methods just described (it would result in a divergent process and hence the Jacobian would be re-computed in the next iteration step). Making use of prior information is therefore not necessary and saves at most one iteration step.

While the application of Broyden's method is well-established it will be useful to more concisely summarize the GSQN algorithm as follows:

1. Chose some initial value Q^0 and a stopping criterion ϵ. For steady state calculations, Q^0 consist of time series of any - but reasonable - constant values and for transition calculations $Q^0 = Q^{*,s}$, i.e., the equilibrium values from steady state calculations (or other constant or non-constant values, e.g., obtained during previous transition calculations).

2. Initialize the Jacobian, $\hat{J}^0$. Use finite difference methods for steady state calculations and $\hat{J}^0 = \hat{J}^{*,s}$ for transitions calculations, i.e. the last approximate Jacobian matrix of steady state iterations (or any other initial matrix such as a scaled identity matrix).

3. For iteration k, determine Q^{k+1} by

$$Q^{k+1} = Q^k - s^k \hat{J}^{-1}[Q^k] G(Q^k), \text{ for } s^k = 1$$

   and evaluate $G(Q^k)$ as well as

$$g(Q^k) = \frac{1}{2} G(Q^k)' G(Q^k)$$

   - If $g(Q^k) < g(Q^{k-1})$ then continue with step 4, `ELSE` re-initialize the Jacobian by setting $\hat{J}^k = \hat{J}^0$ and re-evaluate $g(Q^k)$.

[8] For models where the Jacobian is ill-conditioned at at equilibrium, J^k would not be further updated in case Q^k approaches Q^*. In case iterations are divergent, J^k would only be scaled by line search methods.

- If $g(Q^k) < g(Q^{k-1})$ continue with step 4, `ELSE` start a line-search algorithm. Use a standard backtracking algorithm for line search that stops if $g(Q^k) < g(Q^{k-1})$, if $s^k = s^{\min}$ or a maximum number of line search iterations of three is reached. A good choice for s^{min} is 0.1, see Press et al. (1992) for details.
- If the line search algorithm is successful then continue with step 4, `ELSE` re-initialize $\hat{J}^k$ by finite difference methods, reset $\hat{J}^0 = \hat{J}^k$, re-evaluate $G(Q^k)$ as well as $g(Q^k)$ and continue with step 4.

4. If $\max(|G(Q^k)|) < \epsilon$ then `STOP` and report success, `ELSE` if $\Delta Q^k > \eta$, where η is some small number, determine $\hat{J}^{k+1}$ by Broyden's method as

$$\tilde{\hat{J}}^{k+1} = \hat{J}^k + \frac{(\Delta G(Q^k) - J^k(Q^k - Q^{k-1}))(Q^k)'}{(Q^k)'Q^{k-1}}$$

   If

   - $\Delta Q^k < \eta$ or
   - $\tilde{\hat{J}}^{k+1}$ is ill-conditioned or
   - $\tilde{\hat{J}}^{k+1}$ does not satisfy any prior information regarding its structure

   then re-initialize $\hat{J}^{k+1}$, otherwise proceed with $\hat{J}^{k+1} = \tilde{\hat{J}}^{k+1}$. Re-initialize $\hat{J}^{k+1}$ by first-differences in steady state iterations and by resetting $\hat{J}^0 = \hat{J}^k$. For transition iterations re-initialize by $\hat{J}^{k+1} = \hat{J}^0$ if the starting values are good (i.e. if a steady state was first calculated), otherwise re-initialize by first differences. Continue with step 3.

3 The simulation model

This section develops an OLG model which is used in section 4 for comparison of the relative performance of FGS and GSQN. The simulation model is a three-country version of a multi-country OLG model developed by Börsch-Supan, Ludwig, and Winter (2003) in the tradition of Auerbach and Kotlikoff (1987). It extends the former by allowing for endogenous labor supply and simplifies by using stylized demographic data consisting of less generations which results in a smaller model. Such a smaller model is only used for simplification since it speeds up computations per iteration but otherwise does not affect results regarding the relative performance of the two algorithms. The analysis in section 4 distinguishes between four alternative scenarios increasing m from one to four: (i) one-country closed economy model with exogenous labor supply ($m = 1$), (ii) one-country closed economy model with endogenous labor supply ($m = 2$), (iii) two-country open economy model with endogenous labor supply ($m = 3$) and

(iv) three-country open economy model with endogenous labor supply ($m = 4$). The macroeconomic simulation model is based on a stylized demographic model used to simulate transitions which is described next.

3.1 The demographic model

Demographic projections enter the simulation model via time-specific sizes of living cohorts in year t denoted by $N_{t,a,i}$ where a is age and i is the country index. Sex is irrelevant for the economic model and hence an index for sex is dropped for ease of presentation. Cohorts face mortality risk: $\varsigma_{t,a,i}$ denotes the age and time specific conditional survival probability and $\pi_{t,a,i}$ the unconditional survival probability. There is no migration. The size of a living cohorts is determined recursively by $N_{t+1,a+1,i} = N_{t,a,i}\varsigma_{t,a,i}$ for $a = 1, ..., 20$. Each year the number of newborns is determined by age and time specific fertility rates. Birth is given between the ages $3 - 9$ and fertility rates linearly increase from zero to a peak at the age of 6 and then linearly decrease to zero.

For all three countries, a demographic transition is assumed lasting for 30 years. The assumptions regarding the demographic transitions are arbitrary and are only set to simulate heterogeneous transitions across countries. Departing from an initially constant total population of size 100 in countries one and two and of 200 in country three, the transition starts in year 20 and is characterized by a steadily increasing life expectancy at birth from 15 to 16, 13 to 16 and 14 to 16 for countries one to three respectively.[9] In addition, a fertility transition is assumed for countries one and three. In country one, a baby boom is simulated for years 20 to 30 - an increase of the total fertility rate (TFR) from replacement level of about 2.1 to 2.5 - which is followed by a baby bust for years 30 to 40 characterized by a steady decrease of TFR to 1.5. After the bust, the TFR again increases to replacement level of about 2.1 by year 50. In country three, the same baby boom is simulated for years 20 to 30 but it is not followed by a baby bust. Instead, fertility steadily decreases back to replacement level until year 50.

3.2 The macroeconomic simulation model

General equilibrium of the overlapping generations model is constructed via the production sector where, given factor inputs (capital and labor), output and factor prices are determined. The production sector in each country consists of a representative firm that uses a CES production function which is identical across

[9] A different initial life-expectancy in each country is chosen such that variables across countries differ in the initial steady state. Due to the higher initial population size of country three, country three has about the same initial weight as the other two countries in the three-country open economy version of the model.

countries and given by

$$Y_{t,i} = F(\Omega_{t,i}, K_{t,i}, L_{t,i}) = \left(\alpha K_{t,i}^{-\theta} + (1-\alpha)\left(\Omega_{t,i} L_{t,i}\right)^{-\theta}\right)^{-\frac{1}{\theta}}, \tag{20}$$

where α is the factor share and $\beta = \frac{1}{1+\theta}$ is the elasticity of substitution between the two production factors. $K_{t,i}$ denotes the capital stock, $L_{t,i}$ the aggregate labor force and $\Omega_{t,i}$ is labor augmenting technological change (Harrod neutral) growing at a constant rate g.[10]

From static profit maximization and by the assumption of perfect capital markets, the (world) interest rate is given by

$$r_t = \alpha \frac{Y_{t,i}}{K_{t,i}} - \delta, \tag{21}$$

and the wage rate in each country is

$$w_{t,i} = (1-\alpha)\frac{Y_{t,i}}{L_{t,i}} \tag{22}$$

In order to determine aggregate consumption, optimal household behavior is derived from intertemporal utility maximization. By choosing an optimal consumption and labor supply path, each generation, economically active from period t on, maximizes the sum of remaining discounted life-time utility taking interest rates and wage rates as given from equations (21) and (22). The economic life of a cohort begins at the age of 4, for which $a = 1$ below. The maximum age people can reach is denoted by $Z = 16$. For the exogenous labor supply mode, it is assumed that all households supply one unit of labor for a period of 8 years, $a = 1, ..., 8$, and are retired thereafter. The endogenous labor supply mode does not restrict retirement age nor is a social security system or any other taxes modeled. Of course, leaving out these aspects does not affect the relative performance of the two algorithms.[11] Agents face the risk of prematurely dying with positive wealth. To rule out accidental bequests it is assumed that one-period ahead perfect annuity markets exist which perfectly insure agents against the event of early death. Again, this assumption is irrelevant for the results presented in section 4.[12]

[10] This specification of technological progress insures a steady state in the presence of growth, see, e.g., Barro and Sala-i-Martin (1995).

[11] If agents decide to supply zero units of labor, then shadow wages are calculated, see below. These shadow wages are updated in the next iterations step by relating them to the aggregate wage level and hence move in correspondence with the aggregate wage rate.

[12] As an alternative to assuming perfect annuity markets - which is equivalent to a specific distribution scheme for accidental bequests - accidental bequests could also be distributed by some other scheme, e.g., to all newborns or all children. Since the household model is solved by looping from the last to the first living household, such a redistribution scheme can easily be handled within the household model and otherwise does not affect computations.

In a given period a representative cohort of age a born in year t maximizes the sum of discounted life-time utility. In order to insure a stationary steady state, utility is assumed to be of the familiar Cobb-Douglas form and is given by

$$U(C_{t,a,i}, 1 - l_{t,a,i}) = \frac{1}{1-\sigma} \left(C_{t,a,i}^{\phi} (1 - l_{t,a,i})^{1-\phi} \right)^{1-\sigma}, \tag{23}$$

where $C_{t,a,i}$ is consumption, $l_{t,a,i}$ the labor supply ratio, σ is the coefficient of relative risk aversion and ϕ is a weight attached to leisure.[13]

A household born in time period t maximizes

$$\max_{\{C_{t+a,a,i}, l_{t+a,a,i}\}_{a=1}^{Z}} U = \sum_{a=1}^{Z} \left(\frac{1}{1+\rho} \right)^{a-1} \pi_{t+a,a,i} U(C_{t+a,a,i}, 1 - l_{t+a,a,i}), \tag{24}$$

subject to a dynamic budget constraint given by:

$$A_{t+a+1,a+1,i} = \frac{1}{\varsigma_{t+a,a,i}} \left(A_{t+a,a} (1 + r_{t+a}) + w_{t+a,i} l_{t+a,a,i} - C_{t+a,a,i} \right) \tag{25}$$

where the term $\frac{1}{\varsigma_{t+a,a,i}}$ reflects the assumption of perfect annuity markets.[14] A second constraint requires leisure to be less than one:

$$1 - l_{t,a,i} \leq 1 \iff l_{t,a,i} \geq 0 \tag{26}$$

Maximization yields the inter-temporal Euler equation of consumption as

$$\frac{C_{t+a+1,a+1,i}}{C_{t+a,a,i}} = (\beta(1 + r_{t+a+1}))^{\frac{1}{\sigma}} \left(\frac{v_{t+a+1,i}}{v_{t+a,i}} \right)^{\frac{1}{\sigma}}, \tag{27}$$

and the intra-temporal Euler equation between consumption and leisure as

$$1 - l_{t,a,i} = u_{t,i} C_{t,a,i}, \tag{28}$$

where

$$u_{t,i} = \frac{1-\phi}{\phi} \frac{1}{\tilde{w}_{t,i}}, \tag{29}$$

$$v_{t,i} = u_{t,i}^{(1-\sigma)(1-\phi)} \tag{30}$$

[13] As Auerbach and Kotlikoff (1987) point out a steady state does not exist under CES utility if wages are growing. To avoid such complications (Altig et al. 2001) assume, that growth does not affect productivity but the time endowment of households. Since it is irrelevant for the questions addressed in this paper whether utility is CES or CD, the simpler CD specification is chosen.

[14] By the assumption of perfect annuity markets, end of period assets of households prematurely dying with positive (or negative) wealth are equally shared by the surviving members of the same cohort.

and

$$\tilde{w}_t = w_t + \mu_t. \tag{31}$$

μ_t is the shadow value of wages which is above zero if the household chooses to retire in year $t+a$. In that case, an explicit solution of the household model does not exist and a shooting algorithm would be required to approximate the solution for each household. However, it is much more efficient in terms of computational costs to only calculate shadow wages that would correspond with the labor supply decision using equation (29) and to update these shadow wages during outer loop iterations, see Auerbach and Kotlikoff (1987) and footnote 11.

Equilibrium is constructed via aggregating all household's assets and labor supply decisions in any time period t. As described above, aggregate variables which are stationary will be used to solve the fixed point problem of equation (3). Due to the assumption of perfect world capital markets, the aggregate world capital to output ratio is given by

$$k_t^y = \frac{\sum_{i=1}^{R} A_{t,i}}{\sum_{i=1}^{R} Y_{t,i}}, \tag{32}$$

where R denotes the number of countries considered and $A_{t,i} = \sum_{a=1}^{16} A_{t,a,i}$. The aggregate labor force participation rate in any country i is given by

$$l_{t,i} = \frac{L_{t,i}}{N_{t,i}}, \tag{33}$$

where $L_{t,i} = \sum_{a=1}^{16} l_{t,a,i} N_{t,a,i}$. In terms of notation of section 2, the variables P and Q depend on the scenarios considered:

- Exogenous labor supply / closed economy:
  $P = \{r_t\}_{t=1}^{T}$ and $Q = \{k_t^y\}_{t=1}^{T}$
- Endogenous labor supply / closed economy:
  $P = (\{r_t\}_{t=1}^{T}, \{w_t\}_{t=1}^{T})$ and $Q = (\{k_t^y\}_{t=1}^{T}, \{l_t\}_{t=1}^{T})$
- Endogenous labor supply / two-country open economy:
  $P = (\{r_t\}_{t=1}^{T}, \{w_{t,1}\}_{t=1}^{T}, \{w_{t,2}\}_{t=1}^{T})$ and $Q = (\{k_t^y\}_{t=1}^{T}, \{l_{t,1}\}_{t=1}^{T}, \{l_{t,2}\}_{t=1}^{T})$
- Endogenous labor supply / three-country open economy:
  $P = (\{r_t\}_{t=1}^{T}, \{w_{t,1}\}_{t=1}^{T}, \{w_{t,2}\}_{t=1}^{T}, \{w_{t,3}\}_{t=1}^{T})$ and
  $Q = (\{k_t^y\}_{t=1}^{T}, \{l_{t,1}\}_{t=1}^{T}, \{l_{t,2}\}_{t=1}^{T}, \{l_{t,3}\}_{t=1}^{T})$

The model is calibrated with stylized demographic data as described in section 3.1. Calibration of structural parameters is described in the next section.

4 Results

The sensitivity analysis for comparison of the relative performance of FGS and GSQN presented in this section is grouped into two subsections. First, a steady state analysis is carried out to determine starting values of Q, $Q^{*,ss}$, to be used for the transition analysis. Second, the performance of the two algorithms is compared for the demographic transition scenarios of section 3.1. The transition analysis is carried out by using $Q^0 = Q^{*,ss}$ as starting values. In terms of notation of section 2.3, results reported below refer to model 2, i.e., I first solve for an initial steady state and then use the initial steady state values as initial conditions for the transition calculations.

The structural model parameters of the above OLG model are $\Psi = (\Omega_0, \alpha, \beta, \sigma, \rho, \delta, g)$. In order to compare the performance of the algorithms, three different values parameters of an arbitrary subset of these structural parameters, $\Psi_1 = (\alpha, \beta, \rho, \sigma)$, are combined with each other which results in 81 different parameterizations of the OLG model per model simulation, see table 1.[15] These parameterizations reflect standard parameterizations chosen for OLG models in the literature. For steady state simulations, the starting value of the capital to output ratio is constant at three for the closed economy scenario with exogenous labor supply ($m = 1$). For all other models ($m > 1$), the steady state capital to output ratio resulting from previous models with $m - 1$ endogenous variables is used. The same procedure is adopted for the choice of starting values regarding the labor supply ratio: it is assumed constant at 0.5 for the closed economy model with endogenous labor supply ($m = 2$) and equilibrium labor supply shares resulting from previous computations are used for all subsequent models with $m > 2$. In addition, two alternative dampening factors $w_1 = 0.1$ and $w_2 = 0.3$ will be compared for FGS. Note that more linear models obviously solve faster than more non-linear models.

The convergence criterion ϵ is set to $1e^{-4}$. This is an arbitrary choice. The relative advantage of GSQN increases the lower the convergence criterion since it asymptotically converges at a super-linear rate whereas FGS converges at a linear rate. No convergence may occur under two cases: first, when Q^k is divergent or exhibits cyclical behaviour and second, when $\max(\|G(Q^{k^{max}})\|) > \epsilon$ for some maximum number of iteration steps k^{max}. To best rule out the latter case, k^{max} is set to 100. The convergence properties of the two algorithms are evaluated along two dimensions, number of cases without convergence as well as running time (average and median) as the time it takes for convergent runs (in seconds). Since running time per iteration step differs between the two algorithms, results for the number of iterations it takes until convergence are only reported for sake of completeness.[16]

[15] Except for Ω_0 which is normalized in each iteration step by requiring the model to match arbitrary GDP levels of 100 for countries one and two and 200 for country three.

[16] Running time differs between FGS and GSQN since GSQN requires additional iterations

4.1 The steady state analysis

Convergence results for the steady state analysis of the model are reported in table 2. The table is organized in four sections in increasing order of m. The first two rows of each section show the results for FGS with $w = 0.1$ and $w = 0.3$ respectively. The third row shows the results for GSQN. The last two columns of each row show the relative cases without convergence respectively. GSQN always converges whereas FGS may not converge even if a low value for the dampening factor ($w = 0.1$) is chosen.[17] The fourth and fifth row of each section show the relation between running time (and number of iterations) between FGS ($w = 0.1$) and ($w = 0.3$) and GSQN for those convergent runs of FGS respectively. For example, column one shows the average running time it takes for convergent simulations of FGS divided by the average running time of those GSQN simulations for which FGS also converges.

FGS may not converge even for the low value of the dampening factor of $w = 0.1$. For model simulations considered here, FGS does not converge for about 6 percent of cases when $m > 1$. For values of m larger than 2 robustness of FGS slightly improves largely because starting values are improved. Average convergence speed for $w = 0.1$ is about four times lower than GSQN when $m = 1$ and about 7 times lower when $m = 3$. For $m = 4$ relative convergence speed of FGS again improves due to better starting values and since GSQN wastes relatively more time on Jacobi evaluations. But still, GSQN is about four times faster. For FGS with $w = 0.3$ the algorithm fails to converge in quite many cases (about 14 percent for $m = 1$ up to 50 to 60 percent when $m > 1$), but convergence speed (of the convergent runs) is higher than for FGS with $w = 0.1$. Hence, a higher value of the fixed dampening factor trades robustness for speed. The sections of the table also show median running times since some cases of difficulties in convergence may be driven by outliers, but results do not look much different according to this criterion.

4.2 The transition analysis

While these steady state results already show that FGS is clearly inferior, this may not seem very compelling since non-convergent cases of FGS with $w = 0.1$ are relatively rare and absolute overall speed is high since steady state solutions are fast to compute. But first, non-convergent cases require discretionary fixes by the user which may be time consuming and second, convergence speed slows down if a larger model is used for both transition as well as steady state iterations.[18]

if the Jacobia matrix is re-computed, compare section 2.4.

[17] As a matter of fact, the maximum number of iterations is rarely reached. Most cases of no convergence are due to divergent or cyclical behaviour of the iterations even for $w = 0.1$.

[18] This is the case when the number of generations per time period - here equal to 8 - is increased. Results are available upon request.

Hence computational speed may become relevant after all even for steady state simulations and it is especially important for transition simulations.

Results for transition calculations are shown in table 3 where steady state solutions for $Q^0 = Q^{*,ss}$ and $\hat{J}^0 = \hat{J}^{*,ss}$ are used as initial conditions. First, GSQN again always converges whereas the number of non-convergent cases of FGS for both dampening factors slightly increases to roughly 6 to 7 percent for $w = 0.1$ and 14 to 67 percent for $w = 0.3$. Second, compared to FGS with $w = 0.1$, GSQN is again 2 to 7 times faster than FGS and this speed advantage strictly increases in the number of variables m. Third, the user may be lucky when using FGS with $w = 0.3$ for $m = 1$ since the algorithm might converge even faster than GSQN if it converges at all. Fourth, according to the median running time criterion, the relative performance of FGS is slightly better.[19] However, GSQN is still 2 to about 5.5 times faster.

These results are striking and suggest to use GSQN with good starting values derived from steady state solutions of the simulation model or earlier transition iterations since GSQN is so much superior and since it is so easy to implement.

5 Conclusions

This paper suggests to use Gauss-Seidel-Quasi-Newton instead of conventional (FGS) fast Gauss-Seidel iterations for solving heterogeneous agent models. Standard Quasi-Newton based methods (Broyden's method) are used to determine elements of a Jacobi matrix for Gauss-Seidel iterations which considerably improves convergence both in terms of speed as well as robustness of the iterations. Sensitivity analysis shows, that GSQN increases convergence speed by a factor of two to seven relative to FGS. For transition simulations, this relative speed advantage strictly increases in the number of aggregate endogenous variables, m, required for tatonnement iterations. The particular attractiveness of the algorithm stems from the combination of low computational costs of conventional tatonnement methods with the speed of Newton based methods. The algorithm is globally convergent which is achieved by the additional use of standard line search methods.

Comparing the performance of GSQN to standard one-parameter fixed dampening Gauss-Seidel iterations may seem too restrictive. Intermediate cases - e.g., multi-variate fixed dampening or standard adaptive methods for one-parameter dampening - are of course possible. However, GSQN is likely to dominate any such other - more or less arbitrary - approach as well since it efficiently uses the information of each iteration step and since it is particularly easy to implement.

[19] Since outliers require costly line searches or Jacobi evaluations in case GSQN is used.

References

Altig, D., A. J. Auerbach, L. J. Kotlikoff, K. A. Smetters, and J. Walliser (2001). Simulating fundamental tax reform in the united states. *American Economic Review 91*, 574–595.

Auerbach, A. J. and L. J. Kotlikoff (1987). *Dynamic Fiscal Policy*. Cambridge, MA: Cambridge University Press.

Barro, R. J. and X. Sala-i-Martin (1995). *Economic Growth*. Cambridge, Massachusetts: The MIT Press.

Börsch-Supan, A., A. Ludwig, and J. Winter (2003). Aging, pension reform and capital flows: A multi-country simulation model. MEA Discussion Paper 28-03, University of Mannheim.

Boucekkine, R. (1995). An alternative methodology for solving nonlinear forward-looking models. *Journal of Economic Dynamics and Control 19 (4)*, 711–734.

Fair, R. C. and J. B. Taylor (1983). Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models. *Econometrica 51(4)*, 1169–1186.

Hagemann, L. and D. Young (1981). *Applied Iterative Methods*. Orlando: Academic Press.

Hughes Hallet, A. J. (1982). Alternative techniques for solving systems of nonlinear equations. *Journal of Computational and Applied Mathematics 8 (1)*, 35–48.

Hughes Hallet, A. J. (1984). Multiparameter extrapolation and deflation methods for solving equation systems. *International Journal of Mathematics and Mathematical Sciences 7 (4)*, 793–802.

Judd, K. L. (1999). *Numerical Methods in Economics*. Cambridge, Massachusetts: The MIT Press.

Judd, K. L. (2002). The parametric path method: An alternative to Fair-Taylor and L-B-J for solving perfect foresight models. *Journal of Economic Dynamics and Control 26*, 1557–1583.

Judd, K. L., F. Kubler, and K. Schmedders (2000). Computational methods for dynamic equilibria with heterogenous agents. mimeo, Stanford University.

Juillard, M. (1996). DYNARE: A program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm. CEPREMAP Working Paper, No. 9602, Paris, France.

Juillard, M., D. Laxton, P. McAdam, and H. Pioro (1998). An algorithm competition: First-order iterations versus newton-based techniques. *Journal of Economic Dynamics and Control 22*, 1291–1318.

Laffargue, J. (1990). Resolution d'un modele macroeconomique avec anticipations rationelles. *Annales d'Economic et Statistique 17*, 97–119.

Ortega, J. M. and W. C. Rheinboldt (2000). *Iterative Solution of Nonlinear Equations in Several Variables*. Philadelphia: SIAM.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C*. New York: Cambridge University Press.

Table 1: Calibration parameters, starting values and dampening factors

Parameter	**Value**		
capital share α	0.3	0.4	0.5
substitution elasticity β	0.8	1	1.2
coefficient of relative risk aversion σ	1	2	3
discount rate ρ	0.01	0.02	0.03
growth rate g		0.015	
depreciation rate δ		0.05	

Notes: Parameter values for the discount rate, ρ, the growth rate, g, and the depreciation rate, δ, refer to an annual model. Corresponding discount factors, growth factors and depreciation factors are calculated during solution of the quinquennial model used for illustration of the algorithm.

Table 2: Convergence of FGS and GSQN for the steady state

	Running time		Iteration number		No convergence
	Mean	Median	Mean	Median	%NC
	Closed economy, exogenous labor supply ($m = 1$)				
FGS($w = 0.1$)	2.48	2.55	27.04	27	0.00%
FGS($w = 0.3$)	1.15	0.78	12.86	9	**13.58%**
GSQN	0.61	0.59	6.28	6	0.00%
FGS($w = 0.1$)/GSQN	**4.04**	4.29	4.30	4.50	
FGS($w = 0.3$)/GSQN	1.82	1.30	2.00	1.50	
	Closed economy, endogenous labor supply ($m = 2$)				
FGS($w = 0.1$)	6.76	6.83	65.25	66.5	**6.17%**
FGS($w = 0.3$)	2.79	2.24	24.15	18	**59.26%**
GSQN	1.15	1.09	7.17	7	0.00%
FGS($w = 0.1$)/GSQN	**5.77**	6.24	8.92	9.50	
FGS($w = 0.3$)/GSQN	2.07	1.83	2.91	2.25	
	Two-country model, endogenous labor supply ($m = 3$)				
FGS($w = 0.1$)	12.37	11.92	60.86	61	**4.94%**
FGS($w = 0.3$	5.47	4.56	26.56	21	**51.85%**
GSQN	1.82	1.75	3.74	4	0.00%
FGS($w = 0.1$)/GSQN	**6.78**	6.81	16.33	15.25	
FGS($w = 0.3$)/GSQN	2.83	2.25	6.86	5.25	
	Three-country model, endogenous labor supply ($m = 4$)				
FGS($w = 0.1$)	15.77	14.97	55.59	55	**3.70%**
FGS($w = 0.3$)	6.89	5.77	24.10	19	**50.62%**
GSQN	3.85	3.70	3.06	3	0.00%
FGS($w = 0.1$)/GSQN	**4.09**	4.04	17.99	18.33	
FGS($w = 0.3$)/GSQN	1.66	1.36	7.14	6.33	

Notes: FGS: Conventional fast Gauss-Seidel algorithm with one-parameter dampening. GSQN: Gauss-Seidel-Quasi-Newton algorithm. This table shows steady state convergence results of FGS and GSQN for four different scenarios with 81 model simulations each. The last two rows of each section show the relative performance of FGS and GSQN for convergent runs of FGS only.

Table 3: Convergence of FGS and GSQN for the transition

	Running time		Iteration number		No convergence
	Mean	Median	Mean	Median	%NC
	Closed economy, exogenous labor supply ($m = 1$)				
FGS($w = 0.1$)	8.20	7.88	35.37	34	0.00%
FGS($w = 0.3$)	4.12	3.34	17.26	14	**13.58%**
GSQN	4.11	3.63	15.17	14	0.00%
FGS($w = 0.1$)/GSQN	**1.99**	2.17	2.33	2.43	
FGS($w = 0.3$)/GSQN	0.99	0.94	1.14	1.04	
	Closed economy, endogenous labor supply ($m = 2$)				
FGS($w = 0.1$)	20.75	12.64	44.87	43	**7.41%**
FGS($w = 0.3$)	13.58	12.50	19.78	14	**66.67%**
GSQN	5.88	4.22	13.37	13	0.00%
FGS($w = 0.1$)/GSQN	**3.52**	3.06	3.50	3.31	
FGS($w = 0.3$)/GSQN	1.54	1.71	1.83	1.27	
	Two-country model, endogenous labor supply ($m = 3$)				
FGS($w = 0.1$)	52.92	26.34	52.43	52	**6.17%**
FGS($w = 0.3$)	35.37	28.03	23.60	18	**62.96%**
GSQN	8.65	5.81	10.78	10	0.00%
FGS($w = 0.1$)/GSQN	**6.07**	4.57	5.06	5.20	
FGS($w = 0.3$)/GSQN	2.74	2.32	2.72	2.25	
	Three-country model, endogenous labor supply ($m = 4$)				
FGS($w = 0.1$)	92.47	41.47	57.08	56	**6.17%**
FGS($w = 0.3$)	56.72	56.48	27.73	19	**59.26%**
GSQN	13.22	7.78	10.19	9	0.00%
FGS($w = 0.1$)/GSQN	**6.97**	5.41	5.94	6.22	
FGS($w = 0.3$)/GSQN	3.14	3.09	3.49	2.38	

Notes: FGS: Conventional fast Gauss-Seidel algorithm with one-parameter dampening. GSQN: Gauss-Seidel-Quasi-Newton algorithm. This table shows transition convergence results of FGS and GSQN for four different scenarios with 81 model simulations each when results derived from steady state calculations are used as starting values. The last two rows of each section show the relative performance of FGS and GSQN for convergent runs of FGS only.