

# Recursive computation of heterogeneous agent models\*

Michael Reiter, Universitat Pompeu Fabra, Barcelona

This version, December 2002

## Abstract

The paper presents a method for the recursive computation of models with a continuum of heterogeneous agents. Following the literature, it is assumed that the cross-sectional distribution of wealth in the economy can be represented, to a sufficient degree of accuracy, by a finite number of summary statistics. The main innovation of the method is a very flexible representation of the cross-sectional distribution that can be adapted in the process of the solution. This helps in achieving high accuracy using only a small number of statistics.

The solution is computed by backward induction on a discretization of the state space. The paper discusses approximation techniques to make the algorithm stable and fast. Theoretical considerations and practical experience indicate that the method has very good convergence properties. It is illustrated by an application to a standard model of heterogeneous agents that face uninsurable income risk.

Address of the author:

Michael Reiter  
Universitat Pompeu Fabra  
Department of Economics and Business  
Ramon Trias Fargas, 25-27  
08005 Barcelona  
Spain  
e-mail: reiter@upf.es

---

\*Many thanks to Jim Costain, Wouter den Haan, two anonymous referees and the editor, John Rust, for very helpful comments on earlier drafts of this paper. Support from the Spanish Ministry of Education under the grant PB98-1066 is gratefully acknowledged.

# 1 Introduction

For many questions in macroeconomics, for example dynamic welfare analysis, it is essential to use stochastic heterogeneous agents models with incomplete markets. These models pose some very hard computational problems (for surveys see Rios-Rull, 1995; Rios-Rull, 1997; Judd, Kubler and Schmedders, 2000a). In models with a continuum of agents, a main problem is that the state vector includes the whole wealth distribution of the economy, which is an infinite-dimensional object. Recently, methods have been developed which represent the wealth distribution in the economy by a small number of moments, thereby making the dimension of the state space tractable. The best known examples of this approach are those of Krusell and Smith (1998) and den Haan (1997). Both papers parameterize the aggregate transition laws of the economy via flexible smooth functions, and find the solution by a kind of fixed-point iteration: the method alternates between solving for individual behavior (given aggregate law of motion), and solving for aggregate transition laws given individual behavior. The methods have been successfully applied to models that have indeed a very smooth solution.

This paper describes a new method, which differs from the cited ones in mainly two ways. The main innovation of the method is that it introduces a new and flexible way to handle the cross-sectional distribution of wealth, based on step function approximations of the density. The key aspect is that, for the given number of moments that we use to characterize the distribution, this class of distributions contains not just one, but many elements that have the same moments. In the process of solving the model, we learn something about the distributions that the model generates, and then we base our solution on those distributions. This allows us to achieve a good approximation of the cross-sectional distribution of the model, using only a very small number of statistics. In the present paper, the method is applied to an economy that is characterized by two one-dimensional distributions, but the method can be extended in a straightforward way to handle multivariate cross-sectional distributions.

Second, the method uses time-iteration, not fixed-point iteration, to find the solution of the model. The time-iteration algorithm is in the spirit of dynamic programming, which breaks down one high-dimensional optimization programming in many low-dimensional ones, by solving it recursively on a discrete grid of points in the state space. This is of course not a new idea; it has already been applied, for example, to a heterogeneous agent model with two types of agents in Judd, Kubler and Schmedders (2000b). In the present paper, I discuss some aspects of the implementation that are important to make the algorithm work well in the present context. In particular, I

discuss how to interpolate the value function so as to achieve a reasonable combination of stability, accuracy and speed. I see the main advantage of the time-iteration algorithm in its stability. Theoretical arguments suggests that this method has good convergence properties, and the practical experience so far with a number of models has fully confirmed this expectation.

The two mentioned aspects of the method are basically independent of each other. For example, one could combine the new flexible representation of the cross-sectional distribution with the algorithm of den Haan (1997). Equally, one could use den Haan's representation of the distribution with the time-iteration algorithm.

The plan of the paper is as follows. Section 2 presents a standard heterogeneous agent model that will serve as an example in the later discussions. Section 3 gives an outline of the solution algorithm, while Sections 4 and 5 fill in the details. Section 4 discusses how to solve recursively for an equilibrium and explains the interpolation of the value function. Section 5 discusses how to approximate the cross-sectional distribution. Section 6 compares the algorithm to existing approaches in the literature. In Section 7, the method is applied to the example model. Section 8 concludes.

## 2 A model of uninsurable income risk with a continuum of households

### 2.1 The model

The following model was chosen to illustrate the working of the solution method. It is similar to the models in den Haan (1997), Krusell and Smith (1997) and Krusell and Smith (1998).

There is a continuum of ex ante identical households, with constant mass normalized to 1. Ex post, households can differ w.r.t. to their current labor productivity, impatience, and accumulated wealth. Household  $i$  has a labor endowment of 1 each period, and chooses consumption  $c_{it}$  and labor supply  $l_{it}$  so as to maximize

$$\mathbb{E} \sum_{t=0}^{\infty} \left( \prod_{s=0}^{t-1} \beta_{is} \right) U(c_{it}, l_{it}) \quad (1a)$$

$$U(c, l) = \log(c) + \gamma \log(1 - l) \quad (1b)$$

subject to the budget constraint

$$k_{i,t+1} = (1 + r_t)k_{it} + w_t \xi_{it} l_{it} - c_{it} \quad (1c)$$

the borrowing constraint

$$k_{it} \geq 0 \tag{1d}$$

and the constraint on labor supply

$$0 \leq l_{it} \leq 1 \tag{1e}$$

Both individual productivity  $\xi_{it}$  and the current individual discount factor  $\beta_{it}$  are stochastic and independent of each other and across households. The discount factor can take on only two different values,  $\beta_1$  and  $\beta_2$ , following a time invariant Markov chain with a unique stochastic steady state. In the following, we will refer to the households with different discount factors as different “types”, but one should keep in mind that the types are transitory, and that a priori all households are identical. We assume that the economy starts with the steady state distribution of  $\beta$ , so that the fraction  $\pi_i$  of households that currently have discount factor  $\beta_i$  is constant over time. Individual productivity is i.i.d. over time, with continuous probability distribution  $F_\xi(\xi)$  over the support  $(\underline{\xi}, \bar{\xi})$ .

Asset and insurance markets are incomplete: the individual productivity and discount factor shocks can not be insured, and the household can invest only in one asset, namely  $k_t$ . The sum over all the individual wealth holdings is equal to the capital stock of the economy.

Aggregate gross production is

$$y_t = f(k_t, l_t, \theta_t) = \theta_t k_t^\alpha l_t^{1-\alpha} \tag{2}$$

where

$$k_t = \int k_{it} di \tag{3}$$

$$l_t = \int \xi_{it} l_{it} di \tag{4}$$

The aggregate productivity parameter  $\theta$  follows an exogenous two-state Markov chain. Since individual shocks are independent across agents, the aggregate productivity shock is the only source of aggregate uncertainty. The wage per efficiency unit of labor and the interest rate are set by competitive firms as

$$r_t = f_k(k_t, l_t, \theta_t) \tag{5a}$$

$$w_t = f_l(k_t, l_t, \theta_t) \tag{5b}$$

## 2.2 Equilibrium

Since we define and compute the equilibrium in a recursive fashion, we have to be clear about what are the state variables of the economy. The “natural” state variables, the ones that physically describe the economy, are the wealth holdings of all households, and the state of aggregate productivity. The wealth distribution can be described by a density function for each type of household (low and high patience).

An equilibrium in any set of state variables  $S$  (more precisely, a time-homogenous Markov equilibrium) is given by

1. a law of motion for the aggregate state variables

$$S_{t+1} = H(S_t, \theta_t, \theta_{t+1}) \tag{6}$$

2. Pricing functions  $r_t = r(S_t)$  and  $w_t = w(S_t)$  that are compatible with (5)
3. Individual decision functions  $c(k_{it}, \beta_{it}, \xi_{it}; S_t)$  and  $l(k_{it}, \beta_{it}, \xi_{it}; S_t)$ 
  - which solve the household problem (1) given (6) and (5).
  - which generate the aggregate law of motion in the sense that (6) results from summing over individual savings and labor supply decisions.

Implicit in this definition is that the vector  $S$  is a sufficient statistic to describe the future evolution of the dynamic system, including the probability distribution over all future prices, which is what is relevant for the solution of the individual decision problems.

To my knowledge, it has not been proven for this type of economy with a continuum of agents that such an equilibrium exists when  $S$  is taken to be the above mentioned set of natural state variables (cf. Duffie, Geanakoplos, Mas-Colell and McLennan, 1994, den Haan, 1997, p.359 and Judd et al., 2000b, Section 3, for discussions of this problem; Magill and Quinzii, 1994 prove existence in a similar model with a finite number of agents). Neither do I know a proof that it is impossible to find an equilibrium in a state vector of finite dimension (which means, finding a finite-dimensional aggregator of the model). However, both claims seem very likely, and I follow the literature (Krusell/Smith, den Haan and others) in assuming that the state space of the exact theoretical model consists of the cross-sectional distribution of wealth and aggregate productivity. The exact state space is therefore of infinite dimension, and the problem cannot be handled without a substantial simplification. To make the problem tractable, I introduce the usual approximation (Rios-Rull 1997) that household behavior is a function of only a finite number of statistics of the distribution. The most frequently used

statistics are the moments of the distribution. Judd et al., 2000a, p.18 provide a justification for this practice in terms of symmetric polynomial approximations. We will use a somewhat more general set of statistics, but will often refer to them as moments, for brevity.

### 2.3 Solution of the household problem

In the finite-state approximation, household behavior is based on an approximate aggregate law of motion where next period's moments  $m'$  are a function of today's aggregate state variables  $(m, \theta)$ . Given this, we can turn to solving the household optimization problem. Truncating the space of possible values of  $k$  to a compact set (which is necessary for the numerical solution), the utility function is effectively bounded, which implies (Stokey and Lucas 1989, Section 9.2) that the solution of the household problem satisfies the Bellman equation

$$V(k, \beta, m, \theta) = \int \max_{c(\xi), l(\xi)} \{U(c(\xi), l(\xi)) + \beta E V(k'(\xi), \beta', m'(m, \theta), \theta')\} dF_{\xi}(\xi) \quad (7a)$$

subject to

$$k'(\xi) = (1 + r(m, \theta))k + w(m, \theta)\xi l(\xi) - c(\xi) \quad (7b)$$

where the expectation is over the conditional distribution of next period's  $\beta'$  and  $\theta'$ . The wage and interest rate are functions of the aggregate state variables  $(m, \theta)$  which have to be found in general equilibrium. We have defined  $V$  as the value function of the household before the individual productivity shock is realized. Notice that  $c$  and  $l$  in (7a) are functions of  $\xi$ , since consumption and labor supply are chosen after the income shock has been observed.

**Fact 1.** The value function is strictly increasing and concave in  $k$ .

*Proof.* Analogous to Theorems 9.7 and 9.8 of Stokey and Lucas (1989). The boundedness of returns follows from the boundedness of the capital stock.  $\square$

**Fact 2.** The value function is differentiable in  $k$  for all  $k \geq 0$ .

The proof is given in Section A. As usual, the envelope theorem tells us that

**Fact 3.**

$$\frac{\partial V(k, \beta, m, \theta)}{\partial k} = (1 + r(m, \theta)) \int u'(c(k, \beta, \xi, m, \theta)) dF_{\xi}(\xi) \quad (8)$$

These results on the household value function are independent of the aggregate law of motion.

### 3 Overview of the solution algorithm

#### 3.1 Characterizing the distribution by statistics

The description of our economy contains a pair of distribution functions over wealth,  $F_1(\cdot)$  and  $F_2(\cdot)$ , one for each type of agents. We characterize the joint distribution by a set of statistics  $m$ . We will consider statistics that can be written in the form

$$m_j = \sum_{i=1}^2 \int M_{i,j}(k) dF_i(k) \quad (9)$$

with known functions  $M_{i,j}$ . Formulation (9) includes, for example,

- non-central moments
- fraction of the population in certain intervals
- point masses of known location, if we allow generalized functions  $M_{i,j}(\cdot)$ .

In each category, we can choose the relevant statistic of the joint or the marginal distributions, or any linear combinations of those (for example, the difference in mean between high and low patience types). Note that central moments cannot be directly represented in the form (9); this is no problem, since we can first translate the central into the equivalent non-central moments.

A main element of the algorithm is a mapping that assigns to any vector of statistics  $m$  a pair of distribution functions over  $k$ ,  $\hat{F}_1(\cdot, m)$  and  $\hat{F}_2(\cdot, m)$ . We will call it the distribution selection function. Of course, it has to satisfy the consistency condition

$$\mathcal{M}(\hat{F}_1(\cdot, m), \hat{F}_2(\cdot, m)) = m \quad (10)$$

where  $\mathcal{M}(F_1(\cdot), F_2(\cdot))$  is the function that assigns to the distributions  $F_1(\cdot), F_2(\cdot)$  their moments as in (9). The selection function is “good” if

$$\hat{F}_1(\cdot, \mathcal{M}(F_1(\cdot), F_2(\cdot))) \approx F_1(\cdot) \quad (11a)$$

$$\hat{F}_2(\cdot, \mathcal{M}(F_1(\cdot), F_2(\cdot))) \approx F_2(\cdot) \quad (11b)$$

holds in some sense for those distribution functions  $(F_1(\cdot), F_2(\cdot))$  that are generated in the dynamic solution of the model. Section 5 will explain how to obtain a good distribution selection function.

## 3.2 Schematic outline of the algorithm

We can now outline the basic idea of the solution algorithm. The next two sections will fill in the details.

1. Choose a discrete grid of aggregate points.

Since the individual problem will be solved using value function interpolation (cf. Section 4.2), it is convenient to use Cartesian grids in the aggregate state variables (or in some transformation of them).<sup>1</sup>

2. For each point in the aggregate grid, choose a discrete grid of individual wealth levels.

To facilitate interpolation, the upper and lower bound of the individual grid should be the same for all aggregate grid points. Otherwise, the grid can be very flexible and one can put more points in regions with higher curvature of the value function.

3. Choose an initial distribution selection function  $\hat{F}_i(\cdot, m)$  (cf. Section 5.2).
4. Initialize the value function of the individual household by  $V_0(k, \beta, m, \theta) = 0$ .
5. Separately for each point  $(m, \theta)$  of the grid of aggregate state variables, find the aggregate behavior that is consistent with individual behavior. In the above example, it means to find a vector of next period's moments  $m'$  and a value of aggregate labor supply  $L$  such that the labor market clears, and the evolution of the wealth distribution that follows from individual optimization leads to  $m'$ . For details, see Section 4.1.
6. Having found  $m'$  and  $L$ , update the value function by the Bellman equation (7a).
7. Iterate steps 5 and 6 until convergence of the relevant functions is achieved.
8. Simulate the economy with the obtained solution (cf. Section 5.3). Update the choice of  $\hat{F}_i(\cdot, m)$  using information about the wealth distribution that results from the simulation (cf. Section 5.2). If  $\hat{F}_i(\cdot, m)$  does not change significantly, stop. Otherwise, go to step 5.

---

<sup>1</sup>Reiter (2001b, Section 5) describes a non-Cartesian grid that may help to obtain higher accuracy in models with more than 2 or 3 aggregate state variables.

## 4 Recursive computation

### 4.1 Solving the fixed point problem

Nonlinear root finding can easily fail, and in this sense, solving the fixed point problem (Step 5 in Section 3.2) is the most critical step in the implementation of the algorithm. In our example, it means concretely to find, at a given aggregate state  $(m, \theta)$ , a vector  $(m', L)$  that is consistent with optimizing behavior in the sense that

$$L = \hat{L} \equiv \sum_{i=1}^2 \pi_i \int_0^\infty \int_{\underline{\xi}}^{\bar{\xi}} l(k, \beta_i, \xi, m, \theta, m', L) dF_\xi(\xi) d\hat{F}_i(k, m) \quad (12a)$$

$$m'_j = \hat{m}'_j \equiv \sum_{i=1}^2 \pi_i \int_0^\infty \int_{\underline{\xi}}^{\bar{\xi}} M_{i,j}(k'(k, \beta_i, \xi, m, \theta, m', L)) dF_\xi(\xi) d\hat{F}_i(k, m), \quad (12b)$$

$$j = 1, \dots, n_m$$

where the index  $j$  in the second equation runs over the  $n_m$  moments that we use to characterize the distribution. The  $k'(k, \beta, \xi, m, \theta, m', L)$  and  $l(k, \beta, \xi, m, \theta, m', L)$  are the choices that solve the problem (7). (12) defines a  $n_m + 1$ -dimensional nonlinear root finding problem that I solve by Broyden's method (Press, Flannery, Teukolsky and Vetterling 1986, Section 9.7).

In the course of the solution, the fixed point problem (12) has to be solved many times. To make this work, two points are crucial:

- To guarantee that the problem has a solution, the mapping  $(m', L) \rightarrow (\hat{m}', \hat{L})$  defined in (12) has to be continuous. In exact theoretical models with a continuum of agents, this is usually satisfied even if individual decisions are discontinuous, if one assumes enough heterogeneity across agents. But even if this map is continuous in the exact theoretical model, continuity may be lost in the practical computation due to inappropriate approximations. To avoid this, it is important to preserve the concavity of the value function, as will be discussed in Section 4.2. Then the optimal decisions of the individual react continuously to parameters which are exogenous to the household (such as prices).
- In those cases where the nonlinear root solver fails on (12), one can resort to some simple form of homotopy methods (cf. Judd, 1998, Chapter 5; what Judd calls "simple continuation method" should be sufficient for models with unique solutions). Assume, for example, that we cannot find a fixed point at the aggregate grid point  $x^0$ , but have found one at  $x^1$ . Then solve a sequence of fixed point

problems for  $x^2 = \lambda x^0 + (1 - \lambda)x^1$ , varying  $\lambda$  slowly from 0 to 1. If the fixed point is continuous in  $x$ , we have a very good starting point for each fixed point problem, and convergence will not be a problem.

## 4.2 Interpolating the value function

Since we solve the model on a discrete grid of points, the household value function has to be approximated between grid points. A good interpolation method is crucial for the stability and accuracy of the method. I first discuss how to interpolate in the endogenous state variable  $k$ , then I turn to the exogenous variables.

### Shape-preserving interpolation in $k$

Two properties of the value function are very helpful for the interpolation in  $k$ : First, the value function is differentiable w.r.t.  $k$  (cf. Fact 2) and the envelope condition (8) provides the partial derivative  $V_k$  at the node points. Second, the value function is concave in  $k$  (cf. Fact 1).

It is essential for the functioning of the algorithm to use an interpolation method that guarantees that the interpolant is monotonic and concave (this is called “shape-preserving” interpolation). A concave value function implies that the optimal policy of the household changes continuously w.r.t. parameters that are exogenous to the household (Stokey and Lucas 1989, Section 3.3), such as prices. Using an interpolation scheme that does not preserve concavity, we would induce artificial discontinuities of the household policy functions w.r.t. to these parameters, and substantially complicate the task of solving the fixed point problem described in Section 4.1. The standard shape-preserving interpolation schemes also exploit the derivative information, and this allows to obtain high accuracy in  $k$ -direction with relatively few grid points, an idea that was first applied in Judd and Solnick (1994). Many economists believe that value function iteration is less efficient than Euler equation approaches, but this is not true if one uses this kind of interpolation, because the derivative effectively contains the information that is used in the Euler equation approach (cf. the comparisons in Reiter, 2002b).

In our model, the wealth level  $k$  is the only endogenous state variable for each agent, which is typical for existing stochastic models with a continuum of heterogeneous agents. In one dimension, there are various shape-preserving interpolation schemes available. Schumaker’s quadratic spline (cf. Judd, 1998, p.231) is the best known nontrivial example.<sup>2</sup> In the computations of Section 7, I use a different method that I

---

<sup>2</sup>Two- and three-dimensional shape-preserving interpolation (Costantini and Fontanella, 1990; Renka,

have developed to exploit the fact that we deal with an isoelastic utility function. This is explained in detail in Reiter (2001b, Section 4).

### Interpolation in the exogenous state variables

The next step is to combine the interpolation in  $k$  with some interpolation in the aggregate and exogenous variables  $(\beta, \theta, m)$ . To do this in an efficient way, note first that we do not evaluate the value function randomly at different points  $(k, \beta, \theta, m)$ . We rather fix the aggregate variables  $m$  at some value  $m_0$  (in the process of fixed point finding, we guess next period's moments) and then evaluate the value function many times for different values of  $k$ . We therefore proceed as follows. For all realizations of the discrete variables  $\beta$  and  $\theta$ , and for every point in the  $k_i$ -grid, we compute interpolated values  $\hat{V}$  and  $\hat{V}_k$  of the value function and its partial derivate w.r.t.  $k$  by

$$\hat{V}(k_i, \beta, \theta, m_0) = \sum_j \Psi_j(m_0) V(k_i, \beta, \theta, M_j) \quad (13a)$$

$$\hat{V}_k(k_i, \beta, \theta, m_0) = \sum_j \Psi_j(m_0) V_k(k_i, \beta, \theta, M_j) \quad (13b)$$

where the  $\Psi_j(m)$  are known functions of  $m$ , and  $j$  runs over all the grid points of the aggregate moments. For given  $(\beta, \theta, m_0)$  this gives a grid in  $k$  that we can handle by the one-dimensional interpolation technique described above. As long as we deal with the same  $m_0$ , evaluations are done through interpolation on this one-dimensional grid, which is computationally efficient.

The functions  $\Psi_j(m)$  in (13) fall into two groups:

1. Piecewise linear (simplicial) and multilinear interpolation (Judd 1998, Section 6.12). In this case we have  $\sum_j \Psi_j(z) = 1$  and  $\Psi_j(z) \geq 0$ . The interpolation can be linear both in original state variables or in nonlinear transformations of the state variables.
2. Piecewise quadratic or higher order approximations, including many forms of splines. We still have  $\sum_j \Psi_j(z) = 1$ , but some of the  $\Psi_j(z)$  are negative.

Group 1. has two advantages. First, it guarantees that the interpolated function is concave in  $k$  for any given value of  $z$ . This is an immediate consequence of the fact that a linear combination of concave functions with nonnegative coefficients is concave (Rockafellar 1970, p.33). Second, these interpolation schemes are the ones that make 1996; Kaklis and Karavelas, 1997 is already rather complicated, and higher dimensional convex interpolation schemes are not available at the moment.

it most likely that the algorithm is stable. It is well known (Judd and Solnick 1994, Theorem 2) that dynamic programming with piecewise linear interpolation of the value function is globally convergent, since the contraction property of the Bellman iteration is preserved. In contrast, more general forms of interpolation such as cubic splines can destroy the stability of the algorithm by producing internodal oscillations (see the discussion in Judd, 1998, p.438).

The drawbacks of the linear interpolation schemes is that they often give low accuracy unless we use a dense grid.<sup>3</sup> Since the solution of models with a continuity of agents tends to be very smooth in *aggregate* variables, one can usually achieve higher accuracy by using piecewise quadratic or higher-order approximations. Since these methods lead to some negative coefficients  $\Psi_j(m)$ , the resulting interpolation is not necessarily concave  $k$ . One therefore has to check whether the one-dimensional grid constructed in (13) is compatible with a concave function, which is easy to do. In the application below, this proved to be always the case. If it turned out that those grids occasionally violate concavity, it would be easy to design some combination of piecewise linear and higher-order interpolation, close enough to a linear interpolation to be concave. I skip details since this will not be used in the following application.

Appendix B describes a simple piecewise cubic approximation that makes most of the  $\Psi_j(m)$  equal to zero, so that it can be computed sufficiently fast in the following applications, which use up to 4 moments.

### 4.3 Acceleration steps

In dynamic programming, acceleration steps are those recursive steps where the value function is updated as in (7a), but without updating the optimal controls. One can usually speed up the algorithm substantially by alternating one optimization step with many acceleration steps. This idea cannot be directly applied in our case, since (7a) without optimization would give us new levels of the value function, but no information on the *derivative* of the value function (neither it is guaranteed that the value function is concave after an acceleration step). However, we can apply a somewhat different type of acceleration steps where we do not solve the fixed point problem (Step 6 in the solution algorithm), but only solve the optimization problem of the individuals (Step 5). All

---

<sup>3</sup>We might be able to mitigate the first problem by using linear interpolation in suitable chosen *nonlinear transformations* of the state variables. This does not affect the stability properties of the algorithm. Notice the fact that what is essential for the contraction property is not that the interpolation is linear in the state variables, but that it is linear in the function values at the nodes, for any given  $x$ . However,

values that are exogenous to the individual are taken unchanged from the last fixed-point step. For the model of Section 2.1, this means to use the last obtained values of  $(m', L)$ , for each grid point.

Notice that after using acceleration techniques, step  $T$  of the recursion does not give the solution of the economy with finite horizon  $T$  any longer. This may deteriorate the stability of the algorithm. In the application below, I usually worked with 12 acceleration steps per optimization step, which reduces computation time by a factor of about 4. Using more acceleration steps, it sometimes turned out that the algorithm did not converge.

Obviously, one can try many different ways of updating the individual value function and/or the equilibrium values. Acceleration steps then effectively blur the distinction between “time iteration” and “fixed point iteration” methods.

## 5 Modeling the cross-sectional distribution

A point in the aggregate state space is characterized, among others, by a set of statistics  $m$  of the cross-sectional distribution of wealth. The equilibrium at this point (computed in Step 5 of the solution algorithm) does not just depend on  $m$ , but on the whole cross-sectional distribution of wealth, since the model cannot be exactly aggregated. We therefore have to take a stand on what the distribution is at this point. This is done by the distribution selection function  $\hat{F}_i(\cdot, m)$ , mentioned in Section 3.1. den Haan (1997) uses a parameterized family of distributions that assigns to each vector  $m$  exactly one distribution. The innovation of my method is to employ a very general class of distributions, which contains many different distributions for any given vector  $m$ . For a given  $m$ , we then have the chance to select the distribution that is most appropriate, in the sense that it is close to the cross-sectional distributions that are typically generated by the dynamics of the model.

As a sufficiently general class of distributions, I take densities that are step functions, with possibly many steps. Other choices are possible, for example linear combinations of smooth densities. I have chosen step functions since they are easy to handle, and can be made close enough to any given distribution if we allow for a big number of steps.

To choose the best one among the many step functions with moments  $m$ , we first have to solve the model based on some initial guess of the distribution selection function, then simulate the model, and look at the type of wealth distributions that it generates. Then we solve the model again. This process is iterated, each time updating our distribution selection function. I am now going to describe the details of this procedure.

## 5.1 Approximating distributions by step function densities

As explained above, we approximate each one-dimensional cross-sectional distribution by a density function that is a step function. To be more precise, we divide the state space (of individual wealth) into intervals by a choice of fixed node points  $x_1, \dots, x_{n+1}$  (in the calculations below I have chosen the same nodes for the distribution of both types). Then we denote by  $f_i^j$ ,  $j = 1, 2$ ,  $i = 1, \dots, n$ , the constant density of type  $j$  between node points  $x_i$  and  $x_{i+1}$ .

All the statistics of the form (9) can then be written as linear functions in the step densities  $f_i^j$ . For example, the constraint that the (non-central)  $j$ -th moment of the wealth distribution is equal to  $m_j$  can be expressed as

$$\sum_{i=1}^n (\pi f_i^1 + (1 - \pi) f_i^2) \frac{X_{i+1}^{j+1} - X_i^{j+1}}{j + 1} = m_j, \quad (14)$$

where  $\pi$  is the fraction of type-1 individuals. Point masses of known location can be handled because we can make the difference between  $X_i$  and  $X_{i+1}$  arbitrarily small.

If the state vector contains  $n_m$  distributional characteristics, we have a set of  $n_m + 2$  linear equality restrictions on the densities (including the restrictions that each density has to add to unity). They can be written in the compact form

$$Hf = m \quad (15a)$$

where  $m$  is a known  $n_m + 2$  vector,  $f$  is the  $(2n)$ -vector into which all the  $f_i^j$  are stacked, and  $H$  is a known  $(n_m + 2)$ -by- $2n$  matrix. Of course, the densities also have to satisfy

$$f \geq 0 \quad (15b)$$

Note that the system (15a) and (15b) does not always have a solution. If, for some vector of statistics  $m$ , we cannot find a corresponding step distribution, we have to enlarge the range of wealth ( $X_1, X_{n+1}$ ) and/or refining the nodes (increasing  $n$ ).

## 5.2 Choice of a distribution selection function

If the system (15a) and (15b) has a solution at all, it will usually have many, which gives us the opportunity to choose an appropriate one, as discussed above. We do this by choosing the step distribution that is as close as possible to a given “reference distribution”  $\mathcal{G}(m)$ , itself a function of  $m$ , which we have obtained from simulations of earlier trial solutions of the model. In the following, I first describe what we mean by closeness, and how to compute the closest distribution. Then I describe one way to choose a reference distribution for given  $m$ .

## Computing the distribution closest to a reference distribution

Assume we are given a “reference distribution”  $g = \mathcal{G}(m)$  for a given  $m$ , characterized by the step function densities  $g_i^1$  and  $g_i^2$  which are stacked in the vector  $g$ . As will become clear in the next subsection,  $\mathcal{G}(m)$  itself will in general not have the characteristics  $m$ . We look for an  $f$  that satisfies the restriction (15) and is as close as possible to  $g$ . We will do this in two steps:

1. Obtain  $\tilde{g}$  from  $g$  by a linear shift and stretch operation, which does the following:

- Assume  $m$  contains information about the first two moments. Denote by  $\mu_i(m)$  and  $\sigma_i(m)$  the mean and standard deviation of distribution  $i$  implied by the vector of statistics  $m$ . Denote by  $\mu_i^g$  and  $\sigma_i^g$  the mean and standard deviation of the reference distribution  $\mathcal{G}(m)$ .

For each type  $i = 1, 2$ , compute a step distribution that conforms approximately to the linear transformation of variables  $\tilde{k} = \mu_i(m) + \frac{\sigma_i(m)}{\sigma_i^g}(k - \mu_i^g)$ . Stack the densities into  $\tilde{g}$ . The word “approximately” appears because this linear transformation of variables will generally not give a step distribution on the prespecified nodes  $X$ .

(Details on how this was implemented in the algorithm used below are available from the author on request.)

- if  $m$  contains information only about the mean of the distributions, we compute the  $\tilde{g}$  that conforms to the shift  $\tilde{k} = k + \mu_i(m) - \mu_i^g$ .

2. Now pick the distribution  $f$  that is closest to  $\tilde{g}$  in the weighted mean square sense, i.e., solve

$$\min_{f_i^j} \sum_{i=1}^n \sum_{j=1,2} \omega_i^j (f_i^j - \tilde{g}_i^j)^2 \quad (16)$$

subject to (15a) and (15b), where the  $\omega_i^j > 0$  are given weights. The computations below only use uniform weights  $\omega_i^j = 1$ .

If  $m$  contains no more information than the first two moments, the second step of the operation only serves to make  $f$  fulfill the moment restrictions exactly rather than approximately. If  $m$  contains additional information, the second step may substantially change the distribution  $f$ .

The reason for using the least squares metric in (16) is tractability (cf. discussion below). I am not aware of any other criterion that would allow to pick an  $f$  quickly enough for high values of  $n$ . I decided to put the scale operation 1. in front, because I

feel that the least squares operation makes little sense if  $f$  and  $g$  are of very different scale. I do not claim that this is optimal in any sense, and better procedures may be found in the future.

The problem (16) s.t. (15) is a positive definite quadratic program. There exist algorithms that solve this kind of problem efficiently for moderately large values of  $n$ . However, we may want to use a big  $n$  to obtain good accuracy. For example, in the applications below I use  $n = 1000$ . The QP algorithms that solve the problem exactly are then still too slow, due to the inequality constraints (15b). In practice, it appears good enough to solve the problem *approximately*. Appendix C presents a simple algorithm that does this and is sufficiently fast even for very high values of  $n$  (up to 100000, say). If we are able to handle a large number of  $f_i$ , the method can be extended to handle two or even three-dimensional cross-sectional distributions (the case of models with two or three continuous individual state variables), using densities that are continuous on rectangles etc.

### Choosing a reference distribution

We complete the description of how to approximate the cross-sectional distribution by providing a way to choose a reference distribution  $\mathcal{G}(m)$  for given  $m$ .

1. Initialize  $\mathcal{G}(m)$  by the uniform distribution over the range of  $X$  (conforms to step 3 of Section 3.2).
2. Solve the model recursively (steps 4–7 of Section 3.2). Then simulate the model for  $(T_0 + T_1)$  periods. Discard the results for the first  $T_0$  periods. Save the distributions of the remaining  $T_1$  periods. Denote the stacked densities of these distributions by  $f_t^{sim}$ , and their moments by  $m_t^{sim}$ ,  $t = 1, \dots, T_1$ .
3. Update  $\mathcal{G}(m)$  by

$$\mathcal{G}(m) \equiv \sum_{t=1}^{T_1} \gamma_t f_t^{sim} \quad (17a)$$

$$\gamma_t \equiv \frac{\text{dist}(m, m_t^{sim})^\psi}{\sum_{j=1}^{T_1} \text{dist}(m, m_j^{sim})^\psi} \quad (17b)$$

where  $\text{dist}(x, y)$  is a distance function for vectors  $x$  and  $y$ , for example the Euclidian distance. In the applications below,  $\psi$  was chosen as  $-4$ . This describes step 8 of Section 3.2.

The initialization in 1. is rather arbitrary, and was made mainly in order to show that even with a bad initial guess we can quickly obtain better information on the distribution which allows to improve the solution.

Notice that the distributions used in (17) are close to the stochastic steady state. If we want to include more information on distributions further away from the steady state, we can use simulations that start from a specific point, or result from improbable series of shocks, etc.

### 5.3 Simulating the model

We start the simulation with a given wealth distribution, represented as a step function density. We find the equilibrium vector  $(m', L)$  by solving the fixed point problem described in Section 4.1. Note that this can be done for any value of the aggregate state, not just for points on the aggregate grid. We can therefore simulate the model without explicitly approximating or parameterizing the aggregate transition laws. Having found  $(m', L)$ , we compute next period's wealth distribution, again approximated by a step function density. Note that we do this not by simulating a large number of agents, but by integrating over the current cross-sectional distribution, using the optimal household decisions on  $k'$ . Details can be found in Appendix D.

## 6 Comparison to other solution methods

The basic structure of the algorithm of Krusell and Smith (1998) is the following. The cross-sectional distribution is approximated by a finite number of statistics, usually moments. The state space then consists of these moments, aggregate technology and the individual state variables. The aggregate transition law is approximated by a flexible functional form, which contains free parameters that have to be found in the process.

1. Start the algorithm by a guess on the parameters of the aggregate transition law.
2. Given the aggregate transition law, solve the household optimization problem (for example by value function iteration with interpolation of the value function between grid points).
3. Given individual behavior, simulate the economy for a large number of individuals. From the results of the simulation, update the guess of the parameters of the aggregate transition law.
4. Iterate 2. and 3. until convergence.

A remarkable feature of this algorithm is that one does not have to make any assumptions on the shape of the cross-sectional distribution, because the distribution evolves endogenously in the simulation step 3. The use of simulations, on the other hand, makes it necessary to parameterize the aggregate transition law.

The method of den Haan (1997) deviates from that in many details<sup>4</sup>, but the main innovation is that it parameterizes the cross-sectional density by a flexible functional form. The functional form assigns to any vector of moments (or other statistics) a specific cross-sectional distribution, analogous to our distribution selection function. This allows to solve the model without doing any simulations. den Haan (1997, Section 3.5) argues that this is an important advantage because simulation techniques are affected by sampling errors that disappear very slowly.<sup>5</sup> He maintains the feature of alternating between the solution of the individual problem and updating of the aggregate transition law. While den Haan uses a parameterization of the aggregate transition law, this could be easily avoided in his algorithm.

The most significant innovation of my method is the flexible representation of the cross-sectional distribution that was explained in Section 5. Like den Haan (1997), I can avoid simulation techniques, but with the additional advantage that I can base the solutions on cross-sectional distributions that are close to the ones generated by the model. This helps to achieve high accuracy with a small number of moments, as will be illustrated in the numerical examples below. The algorithm as presented in Section 3.2 does not use any parameterization of the aggregate transition law, but if it turns out that the aggregate dynamics can be well approximated by simple functional forms, there are several ways to exploit this in order to make the algorithm even more efficient. For example, one can solve for an equilibrium at a few points, obtain from that the parameters of the transition law, and use this either for acceleration steps or for having good starting values to compute the equilibrium at the other grid points.

The second significant difference is that I use “time-iteration” rather than “fixed-point iteration”. I think that this gives us the best chance of obtaining a stable algorithm. While the stability of this algorithm cannot be proven, there are good arguments why it should converge safely. After the  $T$ -th iteration of the algorithm, we have computed the equilibrium of the economy with a horizon of  $T$  periods. For all those models

---

<sup>4</sup>Den Haan (1996) presents 2 different methods, one based on simulation, the other one not, which both differ from the methods discussed in the text in a number of aspects.

<sup>5</sup>In standard homogenous agent models, there is evidence that simulation-based methods are clearly less efficient than alternatives such as weighted residual models, cf. Christiano and Fisher (2000). Similar considerations should apply to heterogenous agent models.

where the solution of the finite horizon economies converge to the infinite horizon solution, we then achieve convergence. This is true only if we use approximation techniques that do not induce numerical instabilities. I have chosen value function iteration with the interpolation techniques discussed in Section 4, because they are known to have very good stability properties in dynamic programming problems, and the same can be expected here. These expectations have been confirmed not only by the computation of the above example model, but in a number of ongoing projects with substantially more complicated models (Costain and Reiter, 2001; Määttänen, 2002; Reiter, 2002a). In all cases so far, the algorithm displays the linear convergence behavior with factor  $\beta$  that we know from dynamic programming.

To my knowledge, no similar arguments can be made for the iterative partial updating procedure of Krusell and Smith (1998) and den Haan (1997). However, their procedure is simpler to program, and their algorithms have been shown to converge in a number of interesting applications, at least if one has a reasonable starting value of the iterations and if one uses small enough updating steps. A reasonable procedure may then be to try such a fixed point iteration, for example the one described in den Haan (1997), and combine it with the modeling of the distribution described in Section 5. In case it does not converge, one can resort to the time-iteration method described in Section 4.

## 7 Solving the example model

### 7.1 Calibration

The model is calibrated for annual data. The weight of leisure in the utility function is  $\gamma = 2$ , which results in a rather elastic supply of labor, and a fraction of a bit more than one fourth of total time devoted to work. The share  $\alpha$  of capital in gross output is  $1/3$ , and the annual depreciation rate is  $0.06$ , which are standard values.

Regarding idiosyncratic labor productivity, it is convenient to assume a continuous distribution. Otherwise, the finite point mass in the cross-sectional distribution at  $k = 0$  would generate more and more small, but discrete point masses, which cannot be handled with high precision. I choose, rather arbitrarily, a uniform distribution between  $0.5$  and  $1.5$ . The continuous distribution of  $\xi$  was approximated by a nine-point distribution in the Bellman equation (7a), and by a 100-point distribution in the simulation, cf. Appendix D. For aggregate productivity, since I abstract from growth, I assume values of  $0.98$  and  $1.02$ , and a symmetric probability of switching between states of  $1/3$ , so

that on average we stay in a state for 3 periods (years). This is intended to be a rough representation of the US business cycle.

The individual discount factor can take on the values  $\beta_1 = 0.93$  and  $\beta_2 = 0.96$ , both with unconditional probability of 0.5. It is governed by a very persistent symmetric Markov process. Following Krusell and Smith (1997), the probability of changing within a year from one state to the other is only 2 percent, to represent the idea that the level of patience varies at the frequency of generational changes, or even less. Note that all stochastic processes (aggregate productivity, individual productivity, individual discount factor) are independent, unlike in Krusell and Smith (1997).

## 7.2 Statistics of the cross-sectional distribution

We will solve the same model using a variable number  $n_m$  of statistics describing the cross-sectional distribution, with  $n_m$  ranging between 1 to 4. The following table explains the used statistics. “Type 1” and “Type 2” refer to households with low and high current discount factor, respectively.

$m_1$  mean wealth, total economy

$m_2$  (mean wealth Type 2 - mean wealth Type 1) /  $m_1$

$m_3$  variance wealth -  $(0.5 \cdot m_2 \cdot m_1)^2$

$m_4$  fraction of households with wealth < 5% average annual wage

If we solve the model with  $n_m$  moments, it means we use the statistics  $m_1$  to  $m_{n_m}$ .

When working with Cartesian grids of statistics, one has to be careful to choose the statistics in such a way that all the combinations on the grid are feasible. For example, if the mean wealth in the economy is  $m_1$ , the difference between the mean wealth of the two types cannot be bigger than  $2m_1$ , because wealth is nonnegative. Expressing the difference in mean wealth between the types as a fraction of  $m_1$ , all combinations will be feasible if we restrict  $m_2$  to  $(-2, 2)$ . For the same reason, the third statistic is expressed as the total variance in excess of the variance that is already explained by the difference in average wealth between the two types. Furthermore, it is better to define  $m_1$  and  $m_2$  as above, rather than defining  $m_1$  as the mean of type 1 and  $m_2$  as the mean of type 2. It turns out that the most important statistic is average wealth. The above formulation then allows us to allocate more grid points in the direction  $m_1$ , and fewer grid points in the direction of the less important statistic  $m_2$ .

For  $m_1$ , the computations cover the interval  $(0.5\bar{K}^*, 1.5\bar{K}^*)$  where  $\bar{K}^*$  is the deterministic steady state level of capital. For the other state variables, I first observed the typical values that arise in the simulations of the solution, and then chose a range that is about  $\pm 50\%$  of that value.

### 7.3 Results

Table 1 presents the main results. They all refer to the same calibration (cf. Section 7.1), but use different grid sizes, interpolation techniques etc. The focus is on the analysis of the level of accuracy that has been achieved in the computation. The rhs of the table provides several measures of the accuracy of the individual solution, given the aggregate transition law. The first two columns at the rhs of the table give the maximum absolute normalized Euler residual. The normalization is as in Judd (1992, Section 5.2), so that it measures the proportional error in current consumption, conditional on past and future consumption. For example, a value of 1.81e-03 indicates that in the worst case, the relative error in consumption is about 0.2 percent. The maximum is over several starting points of the aggregate grid, the individual wealth levels, the two levels of  $\beta$  and of  $\theta$ . Individual wealth levels are represented roughly according to their importance in the steady state distribution. The third and fourth column at the rhs of the table present average rather than maximal Euler residuals. In each case, the column “StSt” refers to an aggregate level of capital near the center of the stochastic steady state distribution, while “Trans.” refers to a level 10-30% below or above the steady state value.

The fifth column reports an estimate of the value loss for the individual household that results from using the approximate numerical solution rather than the exact solution. In Reiter (2001a) it was shown that, based on simulation techniques, it is possible to compute a tight upper bound on this value loss without knowing the exact solution.<sup>6</sup> The value loss is reported as an average over households, expressed in permanent relative changes in consumption. For example, a value of 1e-05 means that the value loss is equivalent to a permanent proportional decrease in consumption of 0.001 percent. In interpreting differences in the value loss, one should take into account that it is an upper bound that is reported, and that its computation still contains a sampling error. Based on the experiments reported in Reiter (2001a), I would consider differences in reported value loss of more than about 50 percent as significant.

The results in Rows 1)–11) all use only one moment ( $m_1$  in Section 7.2) to characterize the cross-sectional distribution. Rows 1)–5) report results for different choices of the distribution selection function  $\hat{F}_i(\cdot, m)$ . An entry 0 in the column titled “F” means that we have used the initial choice (cf. Section 5.2), while a number  $n$  means that we have applied  $n$  updating steps of  $\hat{F}_i(\cdot, m)$  (step 8 of Section 3.2). All solutions in the table with the same entry in “F” then use the same selection function. The table reports

<sup>6</sup>Here it was computed from 10 simulations of 225 periods each, starting at a point close to the steady state, following 100 different paths of idiosyncratic shocks in each simulation.

	$n_m$	#P Aggr	W	F	Int.	$ER^{max}$		$ER^{ave}$		V-loss
						StSt	Trans.	StSt	Trans.	
1)	1	20	40	0		1.25e-03	1.74e-03	6.43e-04	7.56e-04	9.53e-03
2)	1	20	40	1		2.26e-03	7.18e-03	3.01e-04	6.25e-04	5.30e-04
3)	1	20	40	2		1.04e-03	3.85e-03	2.06e-04	4.20e-04	9.26e-05
4)	1	20	40	3		8.08e-04	4.86e-03	1.49e-04	4.46e-04	1.02e-05
5)	1	20	40	4		8.20e-04	4.54e-03	1.50e-04	4.36e-04	1.04e-05
6)	1	40	40	4		7.81e-04	3.75e-03	8.14e-05	3.27e-04	2.51e-06
7)	1	10	40	4		1.22e-03	5.56e-03	4.61e-04	1.13e-03	1.28e-04
8)	1	10	40	4	C	1.54e-03	3.45e-03	1.26e-04	3.33e-04	9.84e-06
9)	1	20	40	4	C	6.35e-04	3.71e-03	3.47e-05	3.24e-04	4.46e-07
10)	1	40	40	4	C	7.51e-04	3.71e-03	4.65e-05	3.25e-04	1.30e-06
11)	1	20	100	4	C	3.19e-04	3.68e-03	2.29e-05	3.24e-04	4.26e-07
12)	2	10,6	40	0	C	1.26e-03	1.48e-03	1.53e-04	2.12e-04	2.75e-04
13)	2	10,6	40	4	C	1.36e-03	3.90e-03	1.08e-04	3.54e-04	1.44e-05
14)	2	20,6	40	0	C	1.24e-03	1.27e-03	1.49e-04	1.53e-04	2.69e-04
15)	2	20,6	40	4	C	5.95e-04	4.16e-03	3.72e-05	3.65e-04	1.51e-06
16)	2	20,6	40	5	C	5.49e-04	4.23e-03	3.79e-05	3.71e-04	6.39e-07
17)	2	20,10	40	5	C	6.13e-04	4.23e-03	3.54e-05	3.79e-04	2.83e-06
18)	3	10,6,4	40	0	C	1.72e-03	9.45e-03	2.43e-04	5.23e-04	7.00e-05
19)	3	10,6,4	40	4	C	9.85e-04	1.28e-03	3.05e-04	3.25e-04	5.02e-05
20)	4	8, 4, 4, 4	40	0	C	9.87e-04	1.78e-03	7.10e-05	2.97e-04	1.16e-04
21)	4	8, 4, 4, 4	40	4	C	9.05e-04	1.70e-03	6.78e-05	3.07e-04	3.22e-07

*Notes:*

$n_m$ : number of statistics used for distribution

#P Aggr: number of grid points for each moment of distrib.

#P W: number of grid points for individual wealth holdings

F: number of iterations in update of distribution selection function

Int.: interpolation in aggregate variables; C: piecewise cubic, otherwise piecewise linear

$ER^{max}$ : maximum absolute Euler residual

StSt: close to stochastic steady state; Trans: 10-30% away from steady state

$ER^{ave}$ : average absolute Euler residual

V-loss: value loss of numerical solution, cf. text

Table 1: Accuracy of different solutions

4 updating steps of the distribution, more iterations bring only very small changes. We can observe the enormous increase in accuracy that comes from an improvement in the distribution selection function, reflected both in the Euler residuals and the value loss. The improvement is more clearly reflected in average than in maximum absolute Euler residuals; average residuals also seem to be more closely related to the value loss.

Rows 6)–11) illustrate the effect of different grid sizes and of the use of piecewise linear vs. cubic interpolation (entry “C” in column “Int.”). With 10 points and cubic interpolation we obtain higher accuracy than with 20 points and linear interpolation, but not quite as good as 40 points and linear interpolation. With 20 points and cubic interpolation we do clearly better. Further increases in aggregate or individual grid points do not bring measurable improvements.

Rows 12)–17) contain results based on 2 moments of the distribution, now taking into account the difference in average wealth between low and high discount factor agents (regression analysis of simulations suggests that this is the second most important statistic after mean wealth). Comparing 12) and 14) to 1), we see that the additional moment brings a clear reduction in the value loss, if we use the initial choice of the distribution selection function. However, comparing 15) to 9), there is no visible improvement, which shows that the right choice of  $\hat{F}_i(., m)$  allows us to represent the distribution precisely enough that the inclusion of more statistics brings no measurable gain. Rows 18) and 19) show that a further update of  $\hat{F}_i(., m)$  and an increase in the grid points of  $m_2$  do not change the Euler residuals significantly. The value loss never falls below the value obtained in 9). The results for 3 and 4 moments in rows 18)–21) confirm this. We see that more moments bring only a small improvement in accuracy, even with the initial choice of  $\hat{F}_i(., m)$ . All in all, these results are in line with the findings of Krusell and Smith (1997), (1998), which say that it is the first moment of wealth that matters in this type of model (den Haan, 1997 shows that more moments do matter if we make further changes to the model).

Figure 1 helps to understand the above results. The solid line in both panels shows a “typical” distribution of wealth of Type 1 (low discount factor) agents, close to the stochastic steady state, resulting from a simulation of the model. The other lines show the distributions  $\hat{F}_1(., m)$  where  $m$  are the moments of the distribution depicted by the solid line. We see in the top panel that the initial choice of  $\hat{F}_1(., m)$  is wildly wrong. After the first iteration (conforming to an entry “1” in column “F” of the table) we have greatly improved, after 4 iterations,  $\hat{F}_1(., m)$  is visually almost indistinguishable from the distribution it should approximate. The bottom panel shows the outcome for the initial

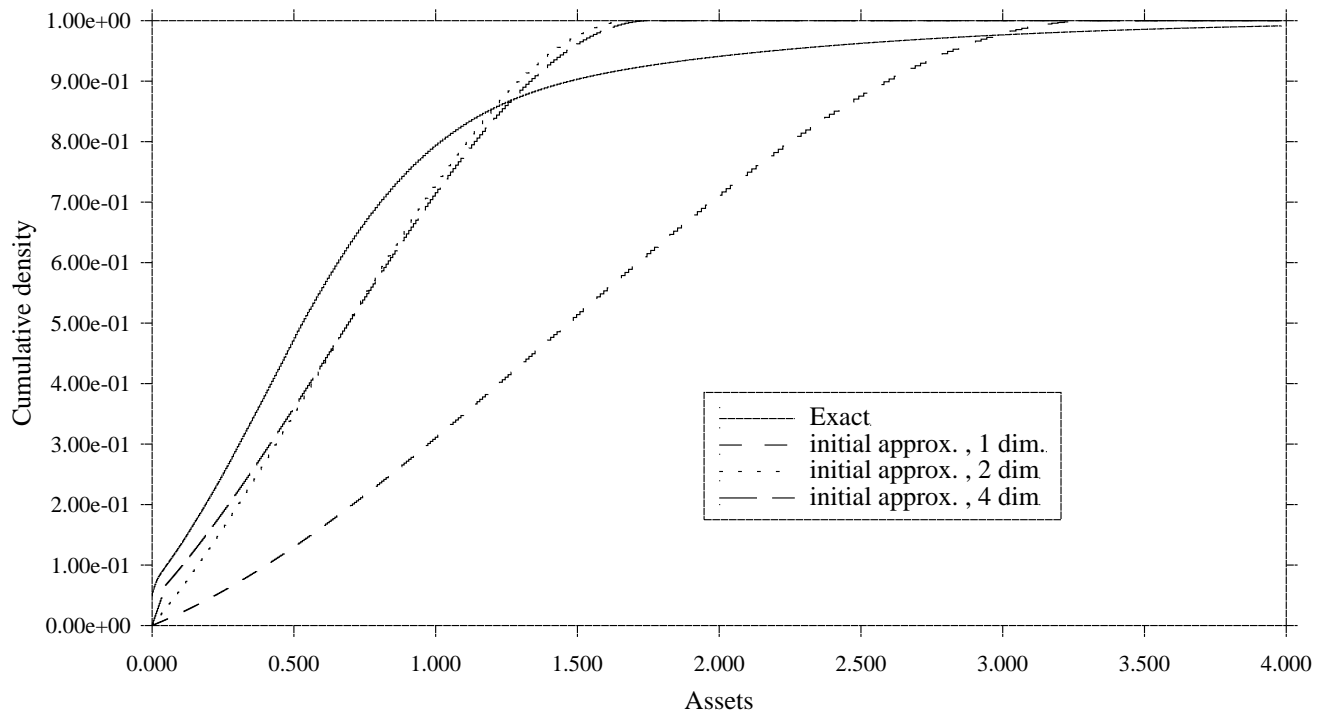
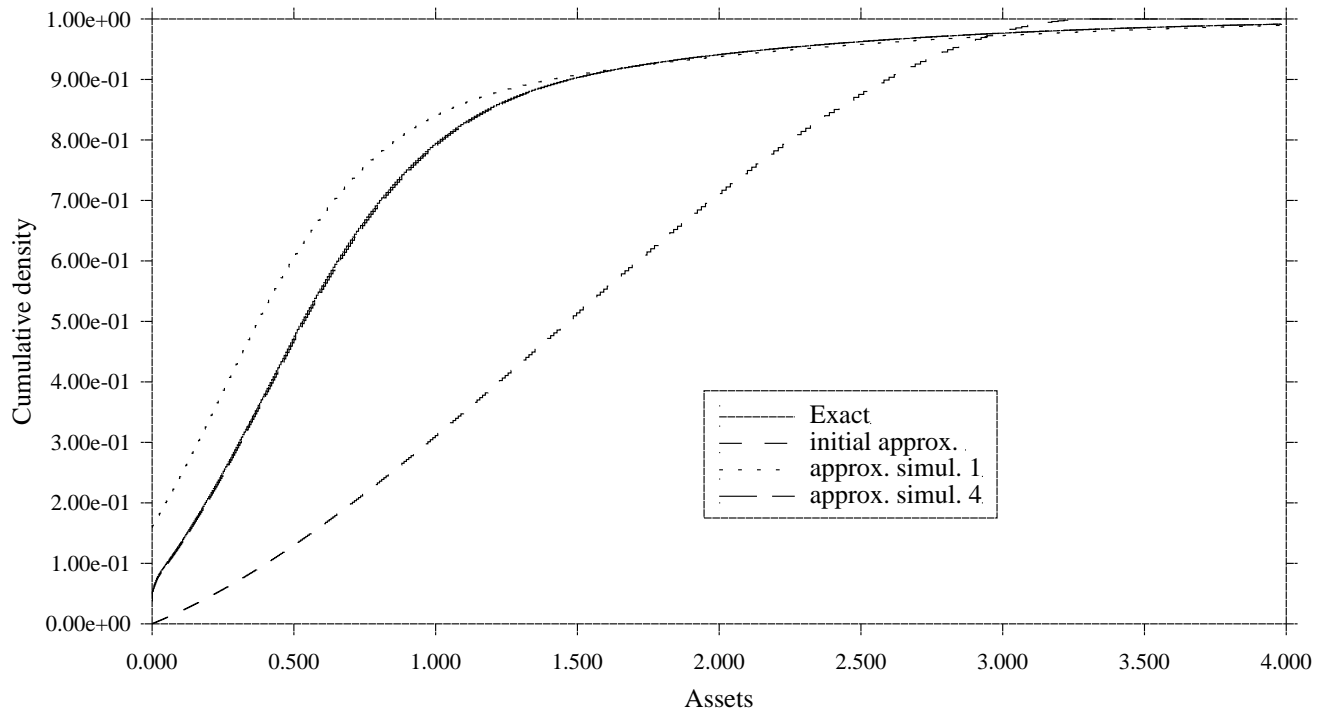


Figure 1: Approximation of cross-sectional wealth distribution, low discount factor households

choice of  $\hat{F}_1(., m)$  using a different number of moments. We see that with 2 moments, the initial  $\hat{F}_1(., m)$  is much closer to the true distribution than with 1 moment, but by far not as close as the updated choice of  $\hat{F}_1(., m)$ . Going to 3 or 4 moments does not add much. The pictures illustrate that the right choice of a distribution selection function allows us to economize on state variables. I expect this to be true quite generally, even if the exact number of statistics needed to come to an accurate solution varies from model to model.

ID	Mean cap.	Variance cap.	Skewness cap.	Wage
1)	3.0e-04 (1.2e-05)	7.4e-02 (1.1e-03)	5.5e-01 (3.2e-02)	9.4e-05 (1.6e-05)
2)	3.6e-04 (1.7e-05)	2.0e-02 (7.8e-04)	1.1e-01 (2.2e-03)	4.5e-05 (2.1e-06)
3)	7.7e-05 (7.2e-06)	3.1e-03 (1.5e-04)	3.2e-02 (8.3e-04)	9.0e-06 (1.7e-06)
4)	3.1e-05 (5.2e-06)	9.3e-04 (3.6e-05)	8.5e-03 (1.6e-04)	4.0e-06 (1.3e-06)
5)	3.0e-05 (5.4e-06)	4.2e-04 (3.1e-05)	7.3e-04 (4.4e-05)	3.9e-06 (1.3e-06)
6)	2.3e-05 (1.2e-06)	6.4e-04 (5.4e-05)	2.8e-04 (7.4e-05)	2.0e-06 (2.4e-07)
7)	1.2e-03 (1.1e-04)	4.4e-03 (4.7e-04)	1.7e-03 (9.9e-04)	1.7e-04 (1.5e-05)
8)	1.4e-04 (3.8e-06)	3.5e-03 (3.0e-04)	1.3e-03 (3.1e-04)	1.1e-05 (2.1e-07)
10)	8.6e-06 (2.8e-07)	2.3e-04 (2.4e-05)	8.3e-05 (6.3e-05)	7.0e-07 (1.3e-08)
11)	1.9e-08 (4.0e-09)	4.3e-06 (5.5e-08)	1.0e-04 (2.1e-06)	6.4e-09 (4.0e-09)

Table 2: “1-correlation coefficient” of aggregate variables (of first differences in parenthesis) between simulations of solution ID and solution 9)

The question remains to what extent all these differences in accuracy matter for aggregate dynamics. Table 2 provides information on the correlation of aggregate variables across simulations of the model obtained from different solutions. All the simulation use the same series of aggregate shocks. The statistic reported is (1-correlation coefficient), and the comparison is between the solution indicated in the table (referring to the numbering in Table 1), and solution 9), which is considered a benchmark for this purpose. The correlation is based on a simulation of the model of 300 periods, of which the first 50 periods were discarded, so that we do not capture any trends that may result from the convergence to a steady state. The numbers in parenthesis refer to the correlation of first differences. For mean capital and wages, all the correlations are very close to unity, even for the least accurate solutions, which indicates that the inaccuracies in the solution of the household problem or the modeling of the distribution matter little if we are only interested in aggregate dynamics. Only variance and skewness show some

nontrivial differences between solutions, but even here the differences are small.

Having confirmed that we can obtain a rather accurate solution using only one moment, one can ask whether this is because the cross-sectional distribution has no impact on the dynamics, or because it varies very little over time. After 50 or 100 periods, when the stochastic steady state is approximately reached, the distribution varies indeed very little (not shown here). To explore the effect of the distribution,

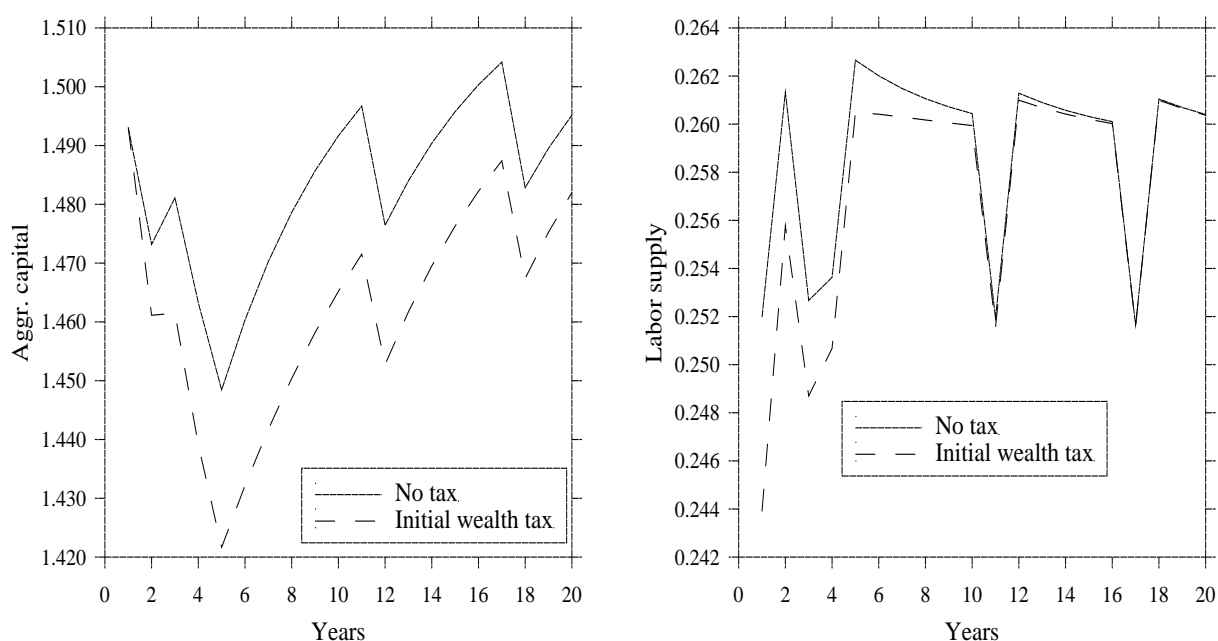


Figure 2: Effect of initial distribution

Figure 2 compares the first periods of a simulation which starts from two different wealth distributions. The first starting point is close to the stochastic steady state, the second one is obtained from the first one by a 20 percent capital levy at the beginning of the first period, which is redistributed as a uniform lump sum subsidy. We see that the decreased inequality reduces labor supply by more than 3 percent, which in turn reduces capital formation, with an accumulated effect that reaches about 3 percent after 10 periods. These differences are not huge, but significant; they reflect the fact that labor supply is very elastic in this model. This suggests that in models with more short-run distributional variation, the distribution does matter. One should note, however, that the redistribution that is engineered by the capital levy is very big compared to short-run

changes that are observed in reality.

Finally, I want to point out that the method is reasonably fast. Arriving at the solution 9) takes about half an hour on a Pentium with 833MHz, including the repeated updating of the distribution selection function. Including more grid points and higher dimensions of the state vector, the computational burden increases substantially. The solutions that use four moments then need about 8 hours. For these calculations, I have not yet exploited any efficiency gains from the fact that the aggregate dynamics is well approximated by a linear function.

## 8 Conclusions

The paper has given a detailed description of a method to solve heterogeneous agents incomplete market models. It uses a new and very flexible way to represent the cross-sectional distribution of wealth (or other state variables). The application to a standard example has shown that this improves the accuracy of the solution, using only a very small number of statistics to characterize the distribution. The method is competitive in terms of speed, but its main attraction is stability. Its good convergence properties have now been confirmed in several applications.

While the proposed method is conceptually straightforward, the practical application requires a lot of programming work. Fortunately, many of the computational tasks are handled by general purpose routines. For the example solved below, all the tasks related to the construction of grids, interpolation, and approximation of distributions, are coded in a C++-library that has been used in a number of projects. Having these libraries available, the programming effort should not be higher than the one required by alternative solution methods.

## A Proof of Fact 2

The proof is a slight modification of Stokey and Lucas (1989, Theorem 9.10). For given  $\beta$ ,  $m$  and  $\theta$  fix a value of  $k_0$ , and define

$$\hat{c}(k, \xi) \equiv c(k_0, \beta, \xi, m, \theta) + (1 + r(m, \theta))(k - k_0) \quad (18)$$

as the level of consumption that gives, with initial capital  $k$  and labor choice  $l(k_0, \beta, \xi, m, \theta)$ , a value of next period's capital of  $k'(k_0, \beta, \xi, m, \theta)$ . Since  $\xi > 0$  and utility is logarithmic, consumption is strictly positive on the compact interval  $(\underline{\xi}, \bar{\xi})$ . Furthermore, since the value function is concave, it is clear that  $c(k_0, \beta, \xi, m, \theta)$  is continuous in  $\xi$ . This implies

that there is some constant  $\underline{c}$  s.t.  $c(k_0, \beta, \xi, m, \theta) \geq \underline{c}$  for all  $\xi$ . Then there is some neighborhood of  $k_0$  s.t.  $\hat{c}(\xi, k) > 0$  for all  $\xi$ . In this neighborhood, define

$$W(k, \beta, m, \theta) = \int \{U(\hat{c}(k, \xi), l(k_0, \beta, \xi, m, \theta)) + \beta E V(k'(k_0, \beta, \xi, m, \theta), \beta', m'(m, \theta), \theta')\} dF_\xi(\xi) \quad (19)$$

Obviously,  $W(k, \beta, m, \theta)$  is concave in  $k$ ,  $W(k, \beta, m, \theta) \leq V(k, \beta, m, \theta)$  and  $W(k_0, \beta, m, \theta) = V(k_0, \beta, m, \theta)$ . The function  $W$  then fulfills all the requirements of Stokey and Lucas (1989, Theorem 4.10), and the claim follows.

## B Piecewise polynomial approximation in exogenous variables

For a higher-order approximation in exogenous variables, I do not use splines, because I have to handle a vector of moments of dimension up to 4. Splines are rather complicated for more than 2 dimensions, and software packages are not available to my knowledge. I rather use the tensor product of a 4-point formula. While the computational effort grows exponentially with the dimension of the state space, the method is still cheap for the 4-dimensional case. This is because we do not interpolate at individual points, but first construct a one-dimensional grid for given moments, as explained in Section 4.2.

We first illustrate the formula for the one-dimensional case. Assume we want to interpolate the function  $f(m)$ , where  $m$  is a scalar, between equidistant node points  $M_j$ ,  $j = 1, \dots, g$ . Between nodes  $M_l$  and  $M_{l+1}$  we use the four-point formula  $f(m) \approx \sum_{j=-1}^2 c_j f(M_{l+j})$ , where the coefficients  $c_j$  are given by

$$c_{-1} = 0.5(-x^3 + 2x^2 - x) \quad (20a)$$

$$c_0 = 0.5(3x^3 - 5x^2) + 1 \quad (20b)$$

$$c_1 = 0.5(-3x^3 + 4x^2 + x) \quad (20c)$$

$$c_2 = 0.5(x^3 - x^2) \quad (20d)$$

and  $x \equiv (m - M_l)/(M_{l+1} - M_l)$ . To interpolate between  $(M_1, M_2)$  and  $M_{g-1}, M_g$  we introduce the auxiliary nodes  $M_0$  and  $M_{g+1}$  and define the function values there as the one obtained by a quadratic interpolation based on the first three and the last three node points, respectively. The interpolation is piecewise cubic, and it is easy to check that the first derivative is continuous at node points. The interpolation is not a cubic spline, since it is not twice-differentiable at node points. Splines are more complicated

because the interpolation at a given point depends on the function value on all the nodes, while the above interpolation only depends on the 4 neighboring function values. The coefficients (20) derive from a linear combination of a quadratic interpolation with nodes  $(M_l, M_{l+1}, M_{l+2})$ , and one with  $(M_{l-1}, M_l, M_{l+1})$  with weights  $x$  and  $1-x$ , respectively. The generalization to non-equidistant nodes is straightforward.

We handle the multidimensional case by tensor products. Assume we use  $n_m$  moments, and denote by  $M_{i,l}$  the  $l$ -th node point in  $i$ -dimension. For given vector of moments  $m$  with components  $m_i$ , define  $l(i)$  such that  $M_{i,l(i)}(m) \leq m_i \leq M_{i,l(i)+1}$  for all  $i$ . For any node point  $M_l$  of the form

$$M_l = \begin{pmatrix} M_{1,l(1)+j_1} \\ M_{2,l(2)+j_2} \\ \dots \\ M_{n_m,l(n_m)+j_{n_m}} \end{pmatrix}, \quad j_i \in \{-1, 0, 1, 2\}, \quad i = 1, \dots, n_m \quad (21)$$

the corresponding weight in (13) is

$$\Psi_l(m) = \prod_{i=1}^{n_m} c_{j_i}(i) \quad (22)$$

where each  $c_{j_i}(i)$  is given by the corresponding formula in (20) with  $x \equiv (m_i - M_{i,l(i)}) / (M_{i,l(i)+1} - M_{i,l(i)})$ . For any  $M_l$  not of the form (21),  $\Psi_l(m) = 0$ .

## C Computing the distribution function for given moments

The task is to solve (16) s.t. (15). Assume both step functions have  $n$  steps, and write  $N = 2n$ . We can write the objective in matrix notation as

$$\min_f (f - g)' \Omega (f - g) \quad (23)$$

where  $\Omega$  is a given diagonal  $N$ -by- $N$  matrix. The matrix  $H$  in (15a) is  $(n_m + 2)$ -by- $N$  and known, where  $n_m$  is the number of statistics to describe the distribution.

For our purpose it is not essential to find the *exact* solution of (23) s.t. (15). For big  $N$ , it is better to use a fast method that provides an approximate solution. Note that the solution to the quadratic problem without the inequality constraints (15b) is given by

$$f = g + \Omega^{-1} H' \lambda \quad (24a)$$

where the vector  $\lambda$  solves

$$H \Omega^{-1} H' \lambda = m - Hg \quad (24b)$$

If  $n_m \ll N$ , the most expensive operation in (24) is the computation of  $HD^{-1}H'$ , which requires  $N(n_m + 2)(n_m + 5)/2$  multiplications. That means, for given  $n_m$ , the effort grows only linearly in  $N$ . Based on this, the following algorithm implements a primitive “active-set” technique to find an approximate solution:

1. Define  $N^{act} \equiv N$ .
2. Solve the system ignoring the non-negativity constraints (15b), i.e., compute (24). If the solution  $f$  satisfies (15b), stop.
3. Denote by  $\mathcal{I}$  the set of  $i$  such that  $f_i < 0$  in the above solution. If the number of negative elements of  $f$  is bigger than some prespecified  $N^{elim}$ , include in  $\mathcal{I}$  only the  $N^{elim}$  smallest elements of  $f$ .  
Set  $f_i = 0$  for all  $i \in \mathcal{I}$ . Compute new matrices  $H$ ,  $\Omega$ ,  $m$  and  $g$  by eliminating the rows (for  $\Omega$ , both rows and columns) that correspond to  $\mathcal{I}$ . Interpret the new  $f$  as containing only the values that have not been set to 0.
4. Set  $N^{act} = N^{act} - \text{number of elements in } \mathcal{I}$ .  
If  $N^{act} < n_m + 2$ , stop. The algorithm has failed.
5. Repeat steps 2–4 until termination.

It is obvious that the algorithm terminates in a finite number of steps, either by finding an approximate solution  $f$  or by signaling failure. Note that it is possible that (23) s.t. (15) has a solution but the above algorithm fails, because at some point we may set an element of  $f$  equal to 0 that is positive in the exact solution. Reducing  $N^{elim}$  will often resolve the problem, but not always.

My practical experience has shown that with  $N^{elim} \approx 0.05N^{act}$  the algorithm performs well for  $n$  up to 100000.

## D Simulating the model

Take as given the current cross-sectional distribution  $F_i(k)$  and the equilibrium values  $(M', L)$ . Then element  $f_i^j$  of the new step function density is obtained as

$f_i^j = \text{Prob}\{k^j \in (X_i, X_{i+1})\} / (X_{i+1} - X_i)$  where

$$\text{Prob}\{k^j \in (X_i, X_{i+1})\} = \sum_{l=1,2} \pi_l T(l, j) \int_0^\infty \int_{\underline{\xi}}^{\bar{\xi}} \mathcal{I}\{k'(k, \beta_l, \xi, m, \theta, m', L) \in (X_i, X_{i+1})\} dF_\xi(\xi) dF_i(k) \quad (25)$$

Here  $\pi_l$  is the fraction of type- $l$  individuals,  $T(l, j)$  is the transition probability between types and  $\mathcal{I}\{\cdot\}$  is the indicator function. The continuous distribution  $dF_\xi(\xi)$  was approximated by a 100-point distribution. For the household decision function  $k'(k, \beta_l, \xi, m, \theta, m', L)$  a piecewise linear approximation in  $k$  was used. The integration in  $dF_i(k)$  can then be done exactly, since  $F_i(k)$  has a step function density.

Due to approximation errors, there will be a slight discrepancy between next period's moments  $m'$  as obtained from integration in (12b), and the moments of the calculated new distribution. The relative size of this difference was typically of the order  $10^{-5}$ . I therefore made a small adjustment of the new distribution to eliminate this discrepancy.

## References

- Christiano, L. J. and Fisher, J. D.: 2000, Algorithms for solving dynamic models with occasionally binding constraints, *Journal of Economic Dynamics and Control* **24**(8), 1179–1232.
- Costain, J. and Reiter, M.: 2001, Business cycles, labor market policy, and precautionary saving, Universitat Pompeu Fabra.
- Costantini, P. and Fontanella, F.: 1990, Shape-preserving bivariate interpolation, *SIAM Journal of Numerical Analysis*.
- Den Haan, W. J.: 1996, Heterogeneity, aggregate uncertainty, and the short-term interest rate, *Journal of Business and Economic Statistics* **14**(4), 399–411.
- den Haan, W. J.: 1997, Solving dynamic models with aggregate shocks and heterogeneous agents, *Macroeconomic Dynamics* **1**, 355–86.
- Duffie, D., Geanakoplos, J., Mas-Colell, A. and McLennan, A.: 1994, Stationary Markov equilibria, *Econometrica* **62**(4), 745–81.
- Judd, K. L.: 1992, Projection methods for solving aggregate growth models, *Journal of Economic Theory* **58**(2), 410–52.
- Judd, K. L.: 1998, *Numerical Methods in Economics.*, MIT Press, Cambridge and London.
- Judd, K. L. and Solnick, A.: 1994, Numerical dynamic programming with shape-preserving splines, mimeo.

- Judd, K. L., Kubler, F. and Schmedders, K.: 2000a, Computational methods for dynamic equilibria with heterogeneous agents, mimeo.
- Judd, K. L., Kubler, F. and Schmedders, K.: 2000b, Computing equilibria in infinite-horizon finance economies: The case of one asset, *Journal of Economic Dynamics and Control* **24**(5-7), 1047–78.
- Kaklis, P. and Karavelas, M.: 1997, Shape-preserving interpolation in  $\mathbb{R}^3$ , *IMA Journal of Numerical Analysis* **17**, 1–46.
- Krusell, P. and Smith, Anthony A, J.: 1997, Income and wealth heterogeneity, portfolio choice, and equilibrium asset returns, *Macroeconomic Dynamics* **1**, 387–422.
- Krusell, P. and Smith, Anthony A, J.: 1998, Income and wealth heterogeneity in the macroeconomy, *Journal of Political Economy* **106**(5), 867–96.
- Määttänen, N.: 2002, Housing, incomplete markets, and the welfare cost of business cycles, Universitat Pompeu Fabra.
- Magill, M. and Quinzii, M.: 1994, Infinite horizon incomplete markets, *Econometrica* **62**, 853–880.
- Press, W., Flannery, B., Teukolsky, S. and Vetterling, W.: 1986, *Numerical Recipes*, Cambridge University Press.
- Reiter, M.: 2001a, Estimating the accuracy of numerical solutions to dynamic optimization problems, Universitat Pompeu Fabra.
- Reiter, M.: 2001b, Recursive computation of heterogeneous agent models, Working paper version, Universitat Pompeu Fabra.
- Reiter, M.: 2002a, The business cycle dynamics of retail inventories, Universitat Pompeu Fabra.
- Reiter, M.: 2002b, Dynamic programming and shape-preserving interpolation, Universitat Pompeu Fabra.
- Renka, R. J.: 1996, SRFPACK: Software for scattered data fitting with a constrained surface under tension, *ACM Transactions on Mathematical Software* **22**(1), 9–17.
- Rios-Rull, J.-V.: 1995, Models with heterogeneous agents, in T. F. Cooley (ed.), *Frontiers of Business Cycle Research*, Princeton University Press, Princeton.

Rios-Rull, J.-V.: 1997, Computation of equilibria in heterogeneous agent models, Federal Reserve Bank of Minneapolis Staff Report: 231.

Rockafellar, R. T.: 1970, *Convex Analysis*, Princeton University Press.

Stokey, N. L. and Lucas, R. E.: 1989, *Recursive Methods in Economic Dynamics*, Harvard University Press, Cambridge, MA.