Computational patterns in exchange economy models

N. Botta, A. Mandel, C. Ionescu, C. Jaeger

Draft 26.03.2009

Abstract

We propose a functional framework for studying dynamical models of exchange economies and argue that – in comparison to standard descriptions – a functional framework has a number of advantages: 1) it allows to express precisely the relationships between the classical notion of general equilibrium and agent-based models of exchange economies; 2) it provides computational description of models of exchange economies which can be unambiguously implemented; 3) it supports the numerical investigation of models of exchange economies by providing a specific set of notions and computational primitives. These can be used to precisely formulate numerical conjectures and to unambiguously setup "crucial" numerical experiments.

We apply the framework to the investigation of a simple model of exchange economies in which multiple equilibrium prices coexist and prices evolve according to a simple trading scheme and to a generic genetic rule.

1 Introduction

1.1 Preliminaries

The idea of exchange economies has been at the core of economic modeling at least since Walras [13]. In such economies, a number of economic actors own certain quantities of goods. Goods have different types and can be freely exchanged between actors. Good prices emerge as ratios at which goods of a given type are exchanged for goods of another type.

The most influential mathematical formulation of this idea was probably the notion of general equilibrium proposed by Arrow and Debreu in [4]. We rehearse this notion in section 3.

While influential, the notion of general equilibrium as a mathematical model of exchange economies has been confronted with a number of criticisms.

One criticism has been that the notion of general equilibrium relies on modeling assumptions which are not supported by empirical evidence, see for instance Kahneman's criticism of the "individual rationality" assumption in [11].

Since the Sonnenschein-Mantel-Debreu "anything goes" result [19], the usefulness of the notion of general equilibrium for decision making and policy advise has been authoritatively questioned, see e.g., **TODO: add references**.

Also the intrinsic incapability of the model to describe dynamic processes such as business cycles has been criticized.

While some criticism might originate from a genuine confusion between a model and its application[1], it is certainly true that the relationships between the mathematical theory of general equilibrium and "real" economies are tenuous and that the application of such theory to, e.g., computational models for decision making and policy advise is not straightforward and is in many ways questionable, see [5].

In the last two decades, a number of alternative formulations of the idea of exchange economies have been proposed as *computer-based* models [2]. More recently, so-called *agent-based* models of exchange economies have been proposed, see [20]. While a precise, established notion of agent-based models is lacking, there seems to be some shared understanding that agent-based models are particular computer-based models. H. Gintis, for instance, maintains that agent-based models are computer simulations of certain games, see [8], page 2.

While computer-based models and, in particular, agent-based models of exchange economies have been applied to the simulation of "real" economies with some success, see [20], their usage in critical applications such as decision making and policy advise is still uncommon.

1.2 Computer-based models, exploratory programming

When considering computer-based models it is useful to distinguish between models which are developed to solve well defined problems and models which are obtained through *exploratory programming* [18].

Examples of the first class of computer-based models are often found in scientific computing end engineering. Here, the problems to be solved are often well understood and can be stated precisely. Computer-based models deliver (usually approximate) solutions to such problems. The accuracy of such solutions can be measured without resorting to empirical data. The availability of well defined problems allows model developers and implementers to design *crucial* experiments. These are experiments for which a negative outcome unambiguously indicate errors in the model or in its implementation.

As an example, let $r2 \in Real \times Real \times Real \rightarrow Complex$ be a program which implements a computer-based model for finding roots of quadratic equations of the form $ax^2+bx+c=0$. A suitable requirement for $r2$ could be

$$\forall a,b,c \in Real, a \neq 0: \quad y = r2\,(a,b,c) \;\Rightarrow\; |ay^2+by+c| <= \epsilon \tag{1}$$

where $\epsilon \in Real$ is a suitable residuum upper bound. Conversely, a tuple $a^*, b^*, c^* \in Real, a^* \neq 0$ for which $|a^*y^2+b^*y+c^*| > \epsilon$ where $y = r2\,(a^*,b^*,c^*)$ would unambiguously indicate that $r2$ does not fulfill the requirement (1). This leaves two possibilities open:

- $r2$ is a wrong implementation of a correct computer-based model for finding roots of quadratic equations of the form $ax^2+bx+c=0$.
- The computer-based model for finding roots of quadratic equations of the form $ax^2+bx+c=0$ is not correct.

Of course, we can rule out the second possibility by choosing a method for finding roots of quadratic equations which has been proved to be correct.

[1] we maintain that few mechanical engineers would criticize the notion of equilibrium of forces on the basis of its failure to describe the dynamics of rigid bodies. At the same time, few mechanical engineers would feel comfortable traveling in an airplane whose wing structure has been computed solely on the basis of static loads analysis.

In programming, requirements like (1) are often called *specifications*. In the example above (1) is a specification for computer-based model for finding roots of quadratic equations of the form $ax^2 + bx + c = 0$. The example makes clear the following remarks:

1. When considering computer-based models which are designed to solve well defined problems, it is useful to distinguish between three notions: the specification, the computer-based model and its implementation.
2. Specifications can be expressed in a clear mathematical notation. They are often written in terms of equations. Often such equations play a prominent role (in the problem solved by computer-based models which fulfill that specification and expressed by the specification itself) and are given special names like problem equations, governing equations, governing laws.
3. Though expressed in mathematical terms, specifications necessarily rely on notions and values which can depend on the computing architecture: in (1), for instance *Real* represents a (computing architecture dependent) set of floating point numbers. The upper bound ϵ is a value which also meaningfully depends on the sets *Real* and *Complex*.

   In many cases it is convenient to trade accuracy of specifications for intuitiveness and replace (1) with

   $$\forall a, b, c \in \mathbb{R}, a \neq 0 : \quad y = r2\,(a, b, c) \;\Rightarrow\; ay^2 + by + c = 0 \tag{2}$$

   This is the approach taken throughout this paper. Notice, however, that (2) is just a suggestive, convenient replacement for the original specification (1). Crucial experiments and proofs of incorrectness have to be based on this specification.
4. Computer-based models are often expressed in algorithmic form using pseudo programming languages [1]. These are more difficult to read and analyze than specifications but still more readable than implementations. Implementations are written in programming languages. *Imperative* programming languages, e.g. C, C++, FORTRAN, Java, trade understandability for efficiency: the lack of *referential transparency* makes reasoning on implementations difficult and error-prone. Functional programming languages, e.g. Haskell [6], close the gap between computer-based model descriptions and implementations and support model analysis, e.g. via equational reasoning. For most realistic applications, however, the run-time efficiency of functional languages is unacceptable.
5. Ideally, developers of computer-based models would like to derive, possibly automatically, computer-based models implementations from their specification. Alternatively, they would like, at least for critical applications such as air traffic control, financial markets, policy advise, to prove that actual implementations are correct i.e. they fulfill their specification. These goals are often too ambitious for complex computer-based models. A realistic alternative is to implement computer-based models by combining software components which are themselves correct or, at least, by combining software components for which crucial experiments in the sense made clear above can be set-up straightforwardly.

Computer-based models of exchange economies and, in particular, agent-based models are often developed through exploratory programming.

Exploratory programming, sometimes also called software prototyping, is often adopted in domains which are not well understood or when it is not clear which software components might be needed for (non-exploratory) implementations.

Implementations of computer-based models obtained through exploratory programming are not designed to solve well-understood problems and do not rely on precise specifications.

They are not primarily developed for application purposes, at least not for critical applications. Because of the lack of specifications, empirical data often play an important role in guiding exploratory programming.

Exploratory programming is, in the development cycle of computer-based models, an activity with a limited lifespan. It is adopted during the first development stages and its primary goal is that of gaining a better understanding of the problems involved and to derive precise specifications. These are, in turn, the pre-conditions for the next development cycle: re-factoring.

Re-factoring on the basis of well-understood problem and precise specifications is a precondition for using computer-based models in critical applications.

1.3 Motivation

In this paper we propose a functional framework for describing and specifying dynamical models of exchange economies.

Our work has been largely inspired by two papers of Herbert Gintis, [8] and [9]. The first paper is remarkable in at least two ways. First because the author provides, together with a mathematical description of an agent-based model of exchange economies a simple, understandable implementation. Second because the model implemented does not correspond, according to our understanding, to the mathematical description.

We turn back to model in some detail in section 5 but let us make very clear that we do not intend to raise any question about the correctness of the implementation made available in `http://people.umass.edu/gintis/` or of our understanding of it. As explained in the previous section this question is, in absence of precise specifications, fruitless.

Instead, our main goal is to address the question of how to describe and specify computer-based models of exchange economies in a language which is more accessible to non-programmers then program listings and yet less ambiguous than mathematical descriptions like the one presented in [8].

The functional framework proposed in this paper is a tentative answer to this question.

A second motivation for our work has been the *Lagom GeneriC* developed at the Potsdam Institute for Climate Impact Research in view of applications to the German economy, see [10] and [14].

Lagom GeneriC is an agent-based model inspired by the model presented in [9]. It has been developed through exploratory programming in the sense discussed in the previous section. The functional framework proposed in this paper has been conceived to assist the development and, in particular, a re-factoring of *Lagom GeneriC*. Our goal is to provide specifications for *Lagom GeneriC* components which can be studied, implemented and validated in isolation. An example of such a component is the trading scheme presented in section 4.5.

1.4 Outline of the paper

The paper is organized as follows: in section 3 we introduce the notion of a system of n_A agents and n_G goods. We formulate the classical notion of general equilibrium in a functional framework and introduce a particular class of time-dependent models of exchange economies. This is the class of models considered throughout this paper.

Section 4 is devoted to the modeling of trade processes. We introduce the notions of transition function, flux functions and trade schedule and show that models of stocks adjustment driven by trade processes can be written as folds of elementary bilateral trades on

random trade schedules. In turn, elementary trades are essentially defined in terms of offer and demand functions and trade schedules are defined by random events. We propose precise specifications for elementary trades, offer and demand functions and for random events and we show that, under reasonable demand functions and for fixed prices, one can define stocks iterations which converge towards equilibrium stocks.

In section 5 we investigate numerically the dynamics of a trade-based stocks adjustment process nested in a genetic algorithm for prices. The coupled system is of the kind described in [9]. In contrast to the results shown there, however, the system appears to allow a much richer dynamics: depending on the number of goods, the mutation factor of the genetic algorithm and on the initial data we observe convergence towards stochastically stable prices, limit cycles or price divergence. We formulate a numerical conjecture about the dynamic of prices and discuss the outcome of a number of crucial experiments designed to falsify this conjecture.

Preliminary conclusions and outlook are discussed in section 6.

2 Notation

Throughout this paper we mainly consider finite sets and finite functions. We represent whatever finite sets of m elements with the zero-based interval of natural numbers $\{0, \ldots, m-1\}$ and denote this interval by $[\,0, m\,)$.

For example, we describe a set of m points in $\mathbb{R}$ through a function $x \in [\,0, m\,) \to \mathbb{R}$ or, equivalently, $x \in \mathbb{R}^m$. While the first notation is more used in computing science, the second one is quite common in engineering and economics. Similarly, we use

$$y \in [\,0, m\,) \to ([\,0, n\,) \to \mathbb{R}) \text{ or } y \in (\mathbb{R}^n)^m$$

to denote a function of two variables with obvious extension to the case of three or more variables. Whenever there is no risk of misunderstandings, we write $x \in \mathbb{R}^{n \times m}$ and $x \in [\,0, m\,) \to [\,0, n\,) \to \mathbb{R}$ instead of the above expressions.

We denote function application by juxtaposition, following a common usage in category theory where "evaluation is a kind of composition" ([12]): $y\,i \in \mathbb{R}^n$ is the i-th coordinate of y and $y\,i\,j \in \mathbb{R}$ the j-th coordinate of $y\,i$.

3 Static models, equilibrium, dynamic models

We consider models of exchange economies which are based on the notion of a *system* of n_A economic agents and n_G goods. Depending on the kind of model considered, the system might be described by different sets of functions. We say that these functions describe the *state* of that system.

3.1 Static models, equilibrium

In static models, the *state* of each agent is solely described by a point in a n_G-dimensional space of goods. Thus, the state of all agents can be represented by a function $x \in \mathbb{R}^{n_G \times n_A}$. We say that $x\,i$ is the stock of the i-th agent. The total quantity of the j-th good present in the system is $\sum_{i=0}^{i<n_A} x\,i\,j$.

In static models of exchange economies, equilibrium is a relation between three functions. Given $x_0 \in \mathbb{R}^{n_G \times n_A}$ and $u \in [\,0, n_A\,) \to \mathbb{R}^{n_G} \to \mathbb{R}$, $x \in \mathbb{R}^{n_G \times n_A}$ is in *equilibrium* if there

exists $p \in \mathbb{R}^{n_G}$ such that

$$\forall\ j \in [\,0, n_G\,) \quad \sum_{i=0}^{i<n_A} x\,i\,j = \sum_{i=0}^{i<n_A} x_0\,i\,j \tag{3}$$

$$\forall\ i \in [\,0, n_A\,) \quad x\,i \in \underset{y\cdot p\ \le\ (x_0\,i)\cdot p}{\operatorname{argmax}} u\,i\,y \tag{4}$$

$$\forall\ j \in [\,0, n_G\,) \quad p\,j \geq 0 \tag{5}$$

In equation (4), $y \cdot p$ represents the scalar product between y and p

$$y \cdot p = \sum_{j=0}^{j<n_g} (y\,j)\ (p\,j) \tag{6}$$

and argmax is an agent-specific set. For the i-th agent, this set contains all stocks that maximize the utility $u\,i$ subject to the constraint $y \cdot p \leq (x_0\,i) \cdot p$.

We say that $p\,j$ is the *price* of the j-th good and $y \cdot p$ the *value* of the stock y according to p; x_0 are the *initial* stocks of the system; u is the *utility profile* of the agents and $u\,i$ the utility function of the i-th agent. The equilibrium stocks $x\,i$ are obtained by re-allocating the total quantities $\sum_{i=0}^{i<n_A} x\,i$ among the n_A agents, see equation (3). This is done in such a way as to maximize the utility of each agent under a *budget* constraint, see equation (4). The budget constraint requires that, for every single agent, the value of its equilibrium stock according to the prices p does not exceed the value of its initial stock.

The theory of general equilibrium uses fixed-point theorems to provide sufficient conditions for an equilibrium to exist, see [4]. While existence is granted under fairly general conditions, uniqueness requires very restricting conditions, see [15].

A prominent example of a utility function which supports multiple equilibria is one proposed by H. Scarf in [17]:

$$u\,i\,y = \min_{j\in[\,0,n_G\,)} (y\,j)\ /\ (w\,j) \tag{7}$$

where the vector of *utility weights* $w \in \mathbb{R}^{n_G}$ is collinear with the total quantities:

$$w = \lambda \left(\sum_{i=0}^{i<n_A} x_0\,i \right)\ \wedge\ \lambda \neq 0 \tag{8}$$

Under these conditions, any arbitrary strictly positive vector of prices $p \in \mathbb{R}^{n_G}$ defines the equilibrium state

$$x\,i = \frac{(x_0\,i)\cdot p}{w \cdot p} w\ , \tag{9}$$

see Appendix 1.

3.2 Dynamic models

In dynamic models of exchange economies the state of the system or part of it evolves in time. In this paper we only consider *time-discrete* dynamic models. For such models, the time-dependent components of the system's state can be represented by functions on natural numbers. For instance, the time-dependent stocks can be represented by

$$x \in \mathbb{N} \to \mathbb{R}^{n_G \times n_A}\ .$$

We say that $x\,t \in \mathbb{R}^{n_G \times n_A}$ are the stocks of the system at time t. The difference $x\,(t+1) - x\,t \in \mathbb{R}^{n_G \times n_A}$ is called a stock transition. We are interested in dynamic models of exchange economies in which:

- The state of the system is given at $t = 0$. For $t > 0$, the state of the system can be computed by iterating a state *transition function*.
- Transition functions take as arguments the present the system's state (stocks, prices and and possibly other state variables), a random variable but neither time nor the next state.
- Transition functions are computed by composing *elementary* transitions. Elementary transitions are transitions in low-dimensional subspaces of the state space: they represent idealized economic processes, e.g., goods *trading*, *production*, *consumption* involving a (small) number of agents independent of n_A.

Examples of state transitions which fulfill these rules are the "barter pairing process" described in [8] and bargaining games, see [16], [7]. In the barter pairing process, pairs of agents interact by exchanging 2 goods in a fixed system of "private" prices. The exchanges are computed by a simple mechanism on the basis of certain "demand" and "offer" functions.

Because the development of dynamic models of exchange economies is heavily based on heuristic, possibly application-specific rules, it is particularly important to anchor such models on some firm, common ground. This is provided by the notion of equilibrium. The idea is that dynamic models which are solely driven by goods exchange processes should converge, for constant prices and is a sense made precise and verifiable by a specification, towards equilibrium stocks.

In the next section we introduce such specification. We construct a particular class of transition functions that fulfill that specification. This is based on *folding* elementary bilateral trade transitions on certain random events. We investigate the asymptotic behavior of sequences of state transitions with numerical experiments and argue that our implementation fulfill the specification.

4 Stocks dynamics: trade-driven models

4.1 System

We consider dynamic models of exchange economies of the kind proposed in [8]. In such models, the state of each agent is described by a stock of positive goods and by three additional variables.

The first one is a vector of n_G strictly positive prices, one for each good. In contrast to the prices introduced in section 3 in defining the notion of equilibrium, prices here are agent-specific. They are sometimes called *private* prices.

The second variable is a good "tag": it associates to the agent a so-called "offer" good. The good tag defines a *partitioning* of the set of agents into n_G disjoint subsets of "offerers", see below. Each subset is called a *sector*.

The third variable is a utility profile of the kind introduced in section 3. We focus the attention on models in which (private) prices, offer goods and utility functions are given and control the time-discrete dynamics of stocks. This is driven by elementary trades between pairs of agents belonging to different sectors. To describe such dynamic we need a fourth variable. This is a sector-to-sector trades upper bound. For any pair $j,\ j' \in [\,0, n_G\,)$, it

defines the maximal number of j'-offerers any j-offerer can trade with. Thus, the state of the whole system can be represented by five functions

$$x \in \mathbb{N} \to \mathbb{R}^{n_G \times n_A} \ ,$$

$$p \in \mathbb{R}^{n_G \times n_A} \ ,$$

$$g \in [\,0, n_G\,)^{n_A} \ ,$$

$$u \in (\mathbb{R}^{n_G} \to \mathbb{R})^{n_A} \ ,$$

$$c \in \mathbb{N}^{n_G \times n_G} \ .$$

We say that $x\,t\,i$ is the stock of the i-th agent at time t; $p\,i$, $g\,i$ and $u\,i$ the prices, the offer good and the utility of the i-th agent; $c\,j\,j'$ is the maximal number of agents of the j'-th sector (of j'-offerers) any agent of the j-th sector (any j-offerer) can trade with.
The function g defines the partitioning of $[\,0, n_A\,)$ into n_G disjoint sectors:

$$[\,0, n_A\,) = \bigcup_{j=0}^{j<n_G} A_j \quad A_j = \{k \in [\,0, n_A\,) \mid (g\,k) = j\} \ . \tag{10}$$

If $i \in A_j$, we say that the i-th agent is an *offerer* of the j-th good or, equivalently, that the i-th agent belongs to the j-th sector.

4.2 Transition function, flux function

We are interested in dynamic models of exchange economies in which stocks evolve in time according to a governing equation of the form:

$$x\,(t+1) \ = \ x\,t \ + \ f\,c\,u\,g\,p\,(x\,t)\,\omega \tag{11}$$

As required in section 3, the transition function

$$y \ \to \ y \ + \ f\,c\,u\,g\,p\,y\,\omega$$

depends, in the governing equation (11), on the present stocks $x\,t$, on p, g, u, c, on a random event $\omega \in \Omega$ but neither on t nor on $x\,(t+1)$. We define Ω in section 4.6 below. We say that (11) is *explicit*. In contrast, governing equations in which the transition function also depends on the next state are called *implicit*.
Moreover, we say that (11) is an explicit *single-step* governing equation in contrast to explicit *multi-step* equations in which the transition function also depends on the values of x at previous times. The function

$$f\,c\,u\,g\,p \ \in \ \mathbb{R}^{n_G \times n_A} \to \Omega \to \mathbb{R}^{n_G \times n_A}$$

is called the *flux* function and $f\,c\,u\,g\,p\,y\,\omega$ is called the flux of y and ω. We say that y is a *stationary* stock of (11) if

$$f\,c\,u\,g\,p\,y\,\omega \ = \ 0 \tag{12}$$

$\forall \omega \in \Omega$.

4.3 Conservative, positivity preserving, stocks value preserving, non-decreasing fluxes

A flux function $f\,c\,u\,g\,p$ is called *conservative* and (11) is called a *conservation form* for x if

$$\forall y \in \mathbb{R}^{n_G \times n_A},\ \omega \in \Omega \quad \sum_{i=0}^{i<n_A} (f\,c\,u\,g\,p\,y\,\omega)\,i \;=\; 0\ .$$

A flux function $f\,c\,u\,g\,p$ is called *positivity preserving* if

$$\forall y \in \mathbb{R}^{n_G \times n_A},\ \omega \in \Omega \quad y \geq 0 \;\Rightarrow\; y \;+\; f\,c\,u\,g\,p\,y\,\omega \geq 0\ .$$

A flux function $f\,c\,u\,g$ is called *stocks value preserving* if

$$\forall p \in \mathbb{R}^{n_G \times n_A},\ y \in \mathbb{R}^{n_G \times n_A},\ \omega \in \Omega \quad y' = y + f\,c\,u\,g\,p\,y\,\omega \;\Rightarrow\; \forall k \in [\,0, n_A\,)\,(y'\,k)\cdot(p\,k) \;=\; (y\,k)\cdot(p\,k)\ .$$

A flux function $f\,c\,u\,g\,p$ is called *non-own-good non-decreasing* if

$$\forall y \in \mathbb{R}^{n_G \times n_A},\ \omega \in \Omega \quad y' = y + f\,c\,u\,g\,p\,y\,\omega \;\Rightarrow\; \forall i \in [\,0, n_A\,), j \in [\,0, n_G\,) \quad j \neq g\,i \;\Rightarrow\; y'ij \;\geq\; y'ij\ .$$

Own-good non-increasing flux functions are defined in a similar fashion. In section 4.7 we show that stocks value preserving and non-own-good non-decreasing flux functions play an important role in studying the converge of stocks of trade-driven models of exchange economies towards equilibrium stocks.

4.4 Elementary trades, demand, offer

As mentioned in section 3, we are interested in dynamic models of exchange economies of the form (11) in which the flux is computed by composing elementary trades.
We describe how elementary trades are composed in the next paragraph. Here we focus the attention on elementary trades. An elementary trade h is a function which models a trade between two agents belonging to two different sectors. It takes as arguments the system's utilities, good tags, prices, stocks and the indexes of the two traders. It computes new stocks:

$$h\,u\,g\,p \;\in\; \mathbb{R}^{n_G \times n_A} \to [\,0, n_A\,) \times [\,0, n_A\,) \to \mathbb{R}^{n_G \times n_A}\ .$$

Trades between pairs of agents belonging to different sectors can be modeled according to a number of different patterns. For instance, one can think of trades based on binding offers or on price negotiation processes. Here we propose a *minimal specification* for elementary trades. We say that any function that fulfills the specification is an elementary trade and we provide a few examples of such functions.
The most intuitive way to express specifications for h is perhaps to consider the flux

$$\delta y \;=\; h\,u\,g\,p\,y\,(i, i') \;-\; y\ .$$

We say that h is an elementary trade if $\forall i, i' \in [\,0, n_A\,)$ such that $g\,i \neq g\,i'$, the flux δy fulfills:

$$\forall k \in [\,0, n_A\,)\ \ k \neq i \ \wedge\ k \neq i' \ \Rightarrow\ \delta y\,k = 0\ , \tag{13}$$

$$\forall j \in [\,0, n_G\,)\ \ j \neq g\,i \ \wedge\ j \neq g\,i' \ \Rightarrow\ \delta y\,i\,j = \delta y\,i'\,j = 0\ , \tag{14}$$

$$\begin{aligned} \delta y\,i\,(g\,i) \;&=\; -\delta y\,i'\,(g\,i) \;\leq\; 0 \\ \delta y\,i\,(g\,i') \;&=\; -\delta y\,i'\,(g\,i') \;\geq\; 0\ , \end{aligned} \tag{15}$$

$$\begin{aligned}(-\delta y\, i\,(g\, i))\,(p\, i\,(g\, i)) &\;\leq\; (\delta y\, i\,(g\, i'))\,(p\, i\,(g\, i')) \\ (-\delta y\, i'\,(g\, i'))\,(p\, i'\,(g\, i')) &\;\leq\; (\delta y\, i'\,(g\, i))\,(p\, i'\,(g\, i))\ ,\end{aligned} \tag{16}$$

$$\begin{aligned}-\,\delta y\, i\,(g\, i) &\;\leq\; y\, i\,(g\, i) \\ -\,\delta y\, i'\,(g\, i') &\;\leq\; y\, i'\,(g\, i')\ .\end{aligned} \tag{17}$$

Condition (13) requires an elementary trade between the i-th and the i'-th agents to affect at most the i-th and the i'-th stocks. Thus, elementary trades model direct, non-mediated interactions between agents. No third-part is affected by such trades.
Condition (14) and (15) require elementary trades between two agents i and i' to be exchanges of their offer goods: the i-th "gives" to the i'-th agent $\delta y\, i\,(g\, i)$ units of its offer good $g\, i$ in exchange for $\delta y\, i\,(g\, i')$ units of the offer good $g\, i'$ of the i'-th agent. Consequently, $\delta y\, i\,(g\, i)$ is required to be negative while $\delta y\, i\,(g\, i')$ must be positive. Thus, the flux of elementary trades is positivity preserving, non-own-good non-decreasing and own-good non-increasing. Because of (15), the flux of elementary trades is also conservative.
Condition (16) requires elementary trades to be win-win exchanges: for both agents the value of the amount of good received has to be at least equal to the value of the good given away. The last condition prevents agents from giving away more offer good than they actually own. It ensures that the flux of elementary trades is positivity preserving.

$$h\, u\, g\, p\, y\,(i, i') \;=\; h\, u\, g\, p\, y\,(i', i)\ .$$

In the following, we give three examples of elementary trades that comply with the specification (13)-(17).

Trivial trade. The first example is that of a trivial trade $h0$. For any index pair i, i', $h0$ simply leaves the stocks unchanged:

$$h0\, u\, g\, p\, y\,(i', i) \;=\; y\ .$$

It is easy to see that the flux associated to $h0$ is zero and fulfills (13)-(17) for any positive y. Trivially, a trivial trade is symmetric.

Demand-based cooperative elementary trade. The second example of an elementary trade is based on the notion of *demand*. A demand function defines, for a given agent and for a given good, a "target" amount of that good.
The demand function takes as arguments a set of utility functions, the good tags, the prices, the stocks, an agent index and a good index. It returns a real number:

$$d\, u\, g\, p\, y \;\in\; [\,0, n_A\,) \to [\,0, n_G\,) \to \mathbb{R}\ .$$

We say that $d\, u\, g\, p\, y\, i\, j$ is the demand of the j-th good of the i-th agent and require d to be positive. Before discussing further specifications for the demand function, let's see how d enters the definition of a demand-based elementary cooperative trade. In such elementary trade (16.1) is satisfied as equality:

$$(-\delta y\, i\,(g\, i))\,(p\, i\,(g\, i)) \;=\; (\delta y\, i\,(g\, i'))\,(p\, i\,(g\, i'))\ . \tag{18}$$

Let $\delta y\, i\,(g\, i') \neq 0$. Then equations (15), (18) and (16.2) imply

$$\frac{p\, i'\,(g\, i')}{p\, i'\,(g\, i)} \;\leq\; \frac{p\, i\,(g\, i')}{p\, i\,(g\, i)}\ . \tag{19}$$

Conversely, if this inequality is not satisfied, $\delta y\, i\, (g\, i')$ has to be equal to zero. Given a demand function d, equation (18) and the specification (13)-(17) fully determine a demand-based elementary cooperative trade $h1\, d$:

$$h1\, 6\, u\, g\, p\, y\, (i, i') \;=\; y \;+\; \delta y \;\Rightarrow\; \delta y\, i\, (g\, i') \;=\; \delta' \;\wedge\; -\delta y\, i\, (g\, i) \;=\; \delta \tag{20}$$

where

$$(\delta', \delta) \;=\; \begin{cases} 0 & \text{if } \frac{p\, i\, (g\, i')}{p\, i\, (g\, i)} \;<\; \frac{p\, i'\, (g\, i')}{p\, i'\, (g\, i)} \\ (\mu 2, \nu 2) & \text{otherwise} \end{cases} \tag{21}$$

$$(\mu 0, \nu 0) \;=\; (\Delta, \Omega) \tag{22}$$

$$(\mu 1, \nu 1) \;=\; \text{if } \mu 0 \leq y\, i'\, (g\, i') \text{ then } (\mu 0, \nu 0) \text{ else } \left(y\, i'\, (g\, i'), y\, i'\, (g\, i') \frac{p\, i\, (g\, i')}{p\, i\, (g\, i)} \right) \tag{23}$$

$$(\mu 2, \nu 2) \;=\; \text{if } \nu 1 \leq\; y\, i\, (g\, i) \text{then } (\mu 1, \nu 1) \text{ else } \left(\nu 1 \frac{p\, i\, (g\, i)}{p\, i\, (g\, i')}, \nu 1 \right) \tag{24}$$

$$\Delta \;=\; d\, u\, g\, p\, y\, i\, (g\, i') \tag{25}$$

$$\Omega \;=\; \Delta \frac{p\, i\, (g\, i')}{p\, i\, (g\, i)} \tag{26}$$

$$\Delta' \;=\; d\, u\, g\, p\, y\, i'\, (g\, i) \tag{27}$$

$$\Omega' \;=\; \Delta' \frac{p\, i'\, (g\, i)}{p\, i'\, (g\, i')}\ . \tag{28}$$

$$\tag{29}$$

In the above equations, Δ, Ω, Δ' and Ω' are the demand and the offer of the i-th and of the i'-th agents, respectively. In (27) and (29), offers are computed from demands to ensure that, for each agent, the value of the own-good given away equals the value of good received, see equation (18).

Demand and offer of the i-th agent are upper bounds for the amounts of the $(g\, i')$-th and of the $(g\, i)$-th goods exchanged in a trade. In particular

$$d\, u\, g\, p\, y\, i\, (g\, i') = 0 \;\Rightarrow\; h1\, d\, u\, g\, p\, y\, (i, i') = y \tag{30}$$

The "default" exchange (Δ, Ω) is limited by two budget constraints. These are expressed in (24)-(25) through *if-then-else* rules. These rules guarantee that the fluxes of the $(g\, i')$-th and of the $(g\, i)$-th goods do not exceed the stocks of own-good of the i'-th and of the i-th agent, respectively.

The fluxes of $h1$ in the $(g\, i')$-th and in the $(g\, i)$-th good for the i'-th agent are determined by (15):

$$\begin{aligned} \delta y\, i'\, (g\, i') \;&=\; -\delta y\, i\, (g\, i') \\ \delta y\, i'\, (g\, i) \;&=\; -\delta y\, i\, (g\, i)\ . \end{aligned} \tag{31}$$

Because of (13) and (14), the fluxes of $h1$ in goods different from the $(g\, i')$-th and from the $(g\, i)$-th are zero for all agents:

$$\forall k \in [\, 0, n_A\,) \;\forall j \in [\, 0, n_G\,) \;\; j \neq g\, i \;\wedge\; j \neq g\, i',\; \delta y\, k\, j \;=\; 0\ , \tag{32}$$

$$\forall k \in [\, 0, n_A\,) \;\; k \neq i \;\wedge\; k \neq i'\; \delta y\, k \;=\; 0\ . \tag{33}$$

Demand-based cooperative elementary trades are not symmetric. Because of (18), demand-based cooperative elementary trades are stock value preserving if the prices are same for all agents.

Demand-based limited cooperative elementary trade. In the elementary trade $h1$ presented above there is nothing that prevents the i'-th agent from receiving an amount of the $(g\,i)$-th good which exceeds its demand Δ' or from giving away more than its offer Ω'. Notice that this can never happen to the i-th agent because of equation (30).

If demand and offer are designed to move one agent's stocks toward some target, this might lead to exchanges which overshoot or undershoot the target of the i'-th agent. This can be avoided by *limiting* the flux by taking into account demand and offer of the i'-th agent. We call the limited scheme $h2$. It is defined by

$$(\delta', \delta) \;=\; \begin{cases} 0 & \text{if } \frac{p\,i\,(g\,i')}{p\,i\,(g\,i)} \;<\; \frac{p\,i'\,(g\,i')}{p\,i'\,(g\,i)} \\ (\mu 4, \nu 4) & \text{otherwise} \end{cases} \tag{34}$$

$$(\mu 3, \nu 3) \;=\; \text{if } \mu 2 \leq \Omega' \text{ then } (\mu 2, \nu 2) \text{ else } \left(\Omega', \Omega' \frac{p\,i\,(g\,i')}{p\,i\,(g\,i)}\right) \tag{35}$$

$$(\mu 4, \nu 4) \;=\; \text{if } \nu 3 \leq \Delta' \text{ then } (\mu 3, \nu 3) \text{ else } \left(\Delta' \frac{p\,i\,(g\,i)}{p\,i\,(g\,i')}, \Delta'\right) \tag{36}$$

$$\tag{37}$$

where $\mu 2$, $\nu 2$, Δ' and Ω' are computed as in the demand-based cooperative elementary trade $h1$, see equations (25), (28) and (29). As for $h1$ one has

$$d\,u\,g\,p\,y\,i\,(g\,i') = 0 \;\Rightarrow\; h2\,d\,u\,g\,p\,y\,(i, i') = y \tag{38}$$

Demand functions The last two examples of elementary trades show that demand-based cooperative elementary trades are essentially defined in terms of the demand function d. It is therefore important to provide minimal specifications for the demand function. The first one is, as already mentioned:

$$d\,u\,g\,p\,y\,i\,j \;\geq\; 0\;.$$

This condition is necessary for $h1$ to be an elementary trade (to fulfill (13)-(17)), in particular, for its flux to be positivity preserving.

In the next paragraph we discuss how elementary trades and, in particular, demand-based cooperative elementary trades, can be combined to build models of exchange economies in which stocks evolve in time upon given fixed prices. As mentioned in section 3.2, we are interested in the particular class of models in which stocks which are in equilibrium with the initial stocks are stationary.

Remember that the flux of the transition function has to be zero at stationary states, see equation 12. A sufficient condition for the flux of demand-based cooperative elementary trades to be zero at equilibrium is that the demand function is zero (at equilibrium). Therefore, a minimal specification for d is

$$\begin{array}{l} y\,i \;\in\; \underset{z\cdot(p\,i)\leq(y\,i)\cdot(p\,i)}{\operatorname{argmax}} \; u\,i\,z \\ \Rightarrow \\ d\,u\,g\,p\,y\,i \;=\; 0\;. \end{array} \tag{39}$$

For the case in which argmax contains exactly one element, the demand function

$$d\,u\,g\,p\,y\,i\,j \;=\; \begin{cases} 0 & \text{if } j = g\,i \\ \max\,0\,\left(\left(\underset{z\cdot(p\,i)\leq(y\,i)\cdot(p\,i)}{\operatorname{argmax}} \; u\,i\,z\right) j - y\,i\,j\right) & \text{otherwise} \end{cases} \tag{40}$$

naturally fulfill the specification above. We call (40) the *natural* demand function.

4.5 Elementary trade based transition function

The most straightforward way of combining two or more elementary trades in a transition function is to apply them in sequence. Sequentializing elementary trades requires a *trade schedule*. This is a *list* of agent index pairs ts such that

$$\text{elem}\,(i,i')\,ts \;\Rightarrow\; g\,i \neq g\,i' \;. \tag{41}$$

Lists are recursively defined data types [3]. We denote an empty list with the symbol []. A list consisting of an $a \in A$ put on the top of a list $as \in ListA$ is denoted by $(a : as)$. We say that $as!!k$ is the k-th element of as. In (41), elem is the boolean function

$$\begin{aligned}&\text{elem} \;\in\; A \to ListA \to Bool\\ &\text{elem}\;a\,as \;=\; \begin{cases}\text{true} & \text{if } \exists\; k \in \mathbb{N} \text{ such that } as!!k = a\\ \text{false} & \text{otherwise .}\end{cases}\end{aligned}$$

Given an elementary trade $h\,u\,g\,p \in (\mathbb{R}^{n_G})^{n_A} \to [\,0, n_A\,) \times [\,0, n_A\,) \to (\mathbb{R}^{n_G})^{n_A}$ and a trade schedule $ts \in List\;[\,0, n_A\,) \times [\,0, n_A\,)$, the transition function that sequentializes $h\,u\,g\,p$ on ts is

$$\begin{aligned}&tr \in \mathbb{R}^{n_G \times n_A} \to \mathbb{R}^{n_G \times n_A}\\ &tr\,y \;=\; \text{fold}\,(h\,u\,g\,p)\,y\,ts \;.\end{aligned} \tag{42}$$

Like elem, fold is a *polymorphic* function. Its signature depends on two parameters X and Y:

$$\text{fold} \in (X \to Y \to X) \to X \to List\;Y \to X \;.$$

Like many functions that operate with lists, see [3], fold is defined recursively by *pattern matching* the case in which the list argument is empty

$$\text{fold}\;f\,x\,[\,] \;=\; x$$

and the case in which the list argument consists of a $y \in Y$ on the top of a (possibly empty) list $ys \in List\;Y$:

$$\text{fold}\;f\,x\,(y : ys) \;=\; f\,(\text{fold}\;f\,x\,ys)\,y \;.$$

As an example, consider the transition function obtained by folding $g = h1\,u\,g\,p$ on the trade schedule $[(0,1),(2,1),(0,2)]$:

$$\begin{aligned}&\text{fold}\;g\,y\,[(0,1),(2,1),(0,2)]\\ &=\\ &\text{fold}\;g\,y\,((0,1) : [(2,1),(0,2)])\\ &=\\ &g\;(\text{fold}\;g\,y\,((2,1) : [(0,2)]))\;(0,1)\\ &=\\ &g\;(g\;(\text{fold}\;g\,y\,((0,2) : []))\;(2,1))\;(0,1)\\ &=\\ &g\;(g\;(g\;(\text{fold}\;g\,y\,[])\;(0,2))\;(2,1))\;(0,1)\\ &=\\ &g\;(g\;(g\,y\,(0,2))\;(2,1))\;(0,1)\;.\end{aligned}$$

Notice that the trade schedule is "executed" from the right to the left. First, one accounts for the effect of the elementary trade between the 0-th and the second agent on y. This results in a new set of stocks, say y'. Then, the elementary trade between the second and the first agent yields new stocks y''. Finally, the interaction between the 0-th and the first agent completes the transition.
For this simple example we can in fact rewrite the computation in a more explicit form as:

$$\begin{aligned}
&\text{fold } g\, y\, [(0,1),(2,1),(0,2)] \;=\; y''' \\
&\text{where :} \\
&y' \;=\; g\, y\, (0,2) \\
&y'' \;=\; g\, y'\, (2,1) \\
&y''' \;=\; g\, y''\, (0,1)\ .
\end{aligned}$$

We conclude this paragraph with a number of remarks. First, elementary trade based transition functions inherit the properties of elementary trades: their flux is conservative, positivity preserving, own-good non-increasing and non-own-good non-decreasing. The flux of transition functions based on stocks value preserving elementary trades is itself stocks value preserving.
Second, elementary trade based transition functions depend on the sequence of interactions defined by the trade schedule. The trade schedule signals the presence of an hidden time-scale. As the hidden (short) time advances, new agent pairs interact according to the schedule. During a time step an agent is either idle or engaged in an elementary trade with exactly another agent. There is no notion of simultaneous trades here. Thus, the 0-th agent of the example above trades both with the first and with the second agent. However, it does so in a well-defined sequence. It first trades with the second agent; then, after the second agent has traded with the first, it trades with the first agent.
Third, a trade schedule can be seen as a relation between agents belonging to different sectors together with a time plan. The relation defines a network of elementary interactions. The time plan their sequence. Depending on the economic system to be modeled, different kind of networks may be more or less appropriate. In *social* networks, for instance, the number of agents a given agent interacts with is bounded from above independently of the sector sizes. We do not consider the problem of defining suitable networks in this paper. In the next paragraph, however, we discuss how to build a random schedule on the basis of given random events an of a sector-to-sector trades upper bound $c \in (\mathbb{N}^{n_G})^{n_G}$. This is the kind of schedule used in the dynamic model of exchange economies presented in sections 4.7 and 5.

4.6 Random events, random schedule

A simple yet general way of construction a random trade schedule for the transition function tr from equation (42) is in terms of a random event $\omega \in \Omega$, see equation (11).

In this framework, sector or agent specific trading preferences can only be modeled via the probability distributions associated to ω. We do not discuss the issue of modeling sector or agent specific preferences here. Instead, we show how to compute a trading schedule ts from a given random event ω and a given sector-to-sector trades upper bound c. We follow an algorithm originally proposed in [8].

In this context, ω is a tuple of random events: $\omega = (\gamma, \alpha, \eta)$. The first element of ω is a

random inverse numbering[2] of $[\,0, n_G\,)$:

$$\gamma \;\in\; [\,0, n_G\,) \to [\,0, n_G\,) \;.$$

There are $n_G!$ such numberings. The second element of ω is a vector of random inverse numbering, one for each sector:

$$\alpha \;\in\; [\,0, n_G\,) \to [\,0, n_A\,) \to [\,0, n_A\,) \;.$$

The j-th component of α is an inverse numberings of the j-th sector A_j. Therefore $\alpha\, j$ are partial functions. They fulfill:

$$\begin{aligned} &\forall j \in [\,0, n_G\,)\;\; \mathrm{dom}\,(\alpha\, j) \;=\; \big[\,0, n_{A_j}\,\big) \\ &\forall j \in [\,0, n_G\,)\;\; \mathrm{ran}\,(\alpha\, j) \;=\; A_j \end{aligned}$$

There are $\prod_{j=0}^{j<n_G} n_{A_j}!$ such numberings. The third element of the tuple ω is, for each pair of sector indexes j and j' and for each offerer in A_j, a random draw of $c\, j\, j'$ offerers in $A_{j'}$:

$$\eta \;\in\; [\,0, n_G\,) \to [\,0, n_G\,) \to [\,0, n_A\,) \to \mathbb{N} \to [\,0, n_A\,) \;.$$

It has to fulfill the following specification:

$$\begin{aligned} &\forall j, j' \in [\,0, n_G\,)\;\; \mathrm{dom}\,(\eta\, j\, j') \;=\; A_j \;, \\ &\forall j, j' \in [\,0, n_G\,)\;\; \mathrm{ran}\,(\eta\, j\, j') \;=\; [\,0, (c\, j\, j')\,) \to A_{j'} \;. \end{aligned}$$

There are $\prod_{j=0}^{j<n_G} \prod_{j'=0}^{j'<n_G} n_{A_{j'}}^{(c\, j\, j')}!$. Thus, Ω is a finite space of events of size

$$|\Omega| \;=\; n_G! \prod_{j=0}^{j<n_G} n_{A_j}! \prod_{j'=0}^{j'<n_G} \prod_{j''=0}^{j''<n_G} n_{A_{j''}}^{(c\, j'\, j'')}!$$

Given $\omega = (\gamma, \alpha, \eta)$ and $c \in (\mathbb{N}^{n_G})^{n_G}$, a random trade schedule ts can be computed as follows:

$$\begin{aligned} ts \;=\; [\; (\alpha\,(\gamma\, j)\, i, \eta\,(\gamma\, j)\,(\gamma\, j')\,(\alpha\,(\gamma\, j)\, i)\, k) \;\mid\; & j \leftarrow [\,0, n_G\,)\,, \\ & j' \leftarrow [\,0, n_G\,)\,, \\ & i \leftarrow \big[\,0, n_{A_{(\gamma\, j)}}\,\big)\,, \\ & k \leftarrow [\,0, c\,(\gamma\, j)\,(\gamma\, j')\,)\;] \end{aligned}$$

The equation defines ts through a list *comprehension*. The expression $a \leftarrow as$ is read "for a drawn from as". It represents the action of iterating over the elements of as. Thus, ts is the list of pairs $(\alpha\,(\gamma\, j)\, i, \eta\,(\gamma\, j)\,(\gamma\, j')\,(\alpha\,(\gamma\, j)\, i)\, k)$ obtained by drawing j and j' from $[\,0, n_G\,)$, i from $[\,0, n_{A_{(\gamma\, j)}}\,)$ and k from $[\,0, c\,(\gamma\, j)\,(\gamma\, j')\,)$. It consists of $\sum_{j=0}^{j<n_G} (n_{A_{(\gamma\, j)}} \sum_{j'=0}^{j'<n_G} c\,(\gamma\, j)\,(\gamma\, j'))$ pairs.

4.7 Trade-driven stocks dynamics

In this section we study the dynamics of stocks under constant prices that is, prices which are the same for all agents: $p\, i = p_0$. We are interested in sequences of stocks as defined by the following iteration:

$$\begin{aligned} &(s\,(t+1), ts\,(t+1)) = \mathrm{randomize}\,(s\, t, c) \\ &y\,(t+1) = \mathrm{fold}\,(h\, u\, g\, p)\,(y\, t)\,(ts\,(t+1)) \end{aligned} \tag{43}$$

[2] a numbering of a finite set A is a bijective function in $A \to [\,0, |A|\,)$. It associates to each element of A exactly one number between 0 and $|A| - 1$. There are $|A|!$ such functions.

starting from an initial seed $s\,0 = s_0$ and from initial stocks $y\,0 = y_0$. We do not discuss here the notion of a seed and the implementation of randomize and we assume that $ts\,(t+1)$ is computed from $s\,t$ and from the sector-to-sector trades upper bound c in terms of uniformly distributed random inverse numberings as discussed in the previous paragraph. Instead, we focus on the dynamics of stocks. We start the iteration with

$$y_0\,i\,j \;=\; \begin{cases} \frac{1+j}{|A_j|} & \text{if } j = g\,i \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

where $|A_j|$ is the number of agents belonging to the j-th sector, see equation (10). Thus, the total quantity of the j-th good in the system is $1 + j$.

For studying the dynamics of stocks governed by (43) when h is a demand-based elementary trade, e.g., $h = h1\,d$ or $h = h2\,d$, it is useful to introduce the notion of *total demand*:

$$\begin{aligned} &td\,d\,u\,g\,p \in \mathbb{R}^{n_G \times n_A} \to \mathbb{R} \\ &td\,d\,u\,g\,p\,z \;=\; \sum_{i=0}^{i<n_A} \sum_{j=0}^{j<n_G} d\,u\,g\,p\,z\,i\,j\ . \end{aligned} \tag{45}$$

For these transition functions, the total demand $td\,d\,u\,g\,p$ is an upper bound for the total amount of (non-own) goods which can be exchanged during a transition.

If the utility profile u and the initial stocks y_0 support, for given prices p_0, unique stocks y_e, the natural demand function (40) guarantees that the transition function tr of equation (42) is *total demand non-increasing*, see appendix A:

$$z' = tr\,z \;\Rightarrow\; td\,d\,u\,g\,p\,z' \leq td\,d\,u\,g\,p\,z\ . \tag{46}$$

This implies that the iteration (43) is total demand non-increasing:

$$td\,d\,u\,g\,p\,y\,(t+1) \leq td\,d\,u\,g\,p\,y\,t\ . \tag{47}$$

Under these conditions, one would like to ensure that, if the total demand of $y\,t$ converges towards zero, $y\,t$ is stationary and equal to y_e.

The first property is guaranteed by positivity of demands and, for $h = h1\,d$ or $h = h2\,d$, by equations (30), (38), see appendix B. A sufficient condition for $y\,t$ to be in equilibrium if its total demand is zero is that the second argument of max in equation (40) is positive. This is the case if the demand based elementary trade h is limited i.e. $h = h2\,d$. Limiting the flux between any two agents by demand and offer of both agent is also necessary for the iteration (43) to converge.

We use this property to setup two experiments. These are designed to validate our implementation of (43) and to understand how the dynamics of stocks for fixed constant prices depends on the number of agents and on the sector-to-sector trades upper bound.

In both experiments, the number of goods is equal to 3 and the utility function is the Scarf utility function with utility weights w equal to the total initial stocks:

$$w\,j = 1 + j \tag{48}$$

Under these conditions, any vector of prices p_0 defines equilibrium stocks

$$y_e\,i = \frac{(y_0\,i)\cdot p_0}{w \cdot p_0} w \tag{49}$$

and the total utility at equilibrium is

$$\sum_{i=0}^{i<n_A} u\,i\,y_e \;=\; 1\;, \tag{50}$$

see appendix C. We consider random prices in $(0, 1]$. In the first experiment we fix the number of agents per sector to 1000 and let the sector-to-sector trades upper bound c vary. In the limited case $h = h2\,d$, we expect the total demand to converge towards zero and the total utility to converge towards one as the stocks converge towards the equilibrium stocks (49). The results shown in figure 1 do not contradict this hypothesis.

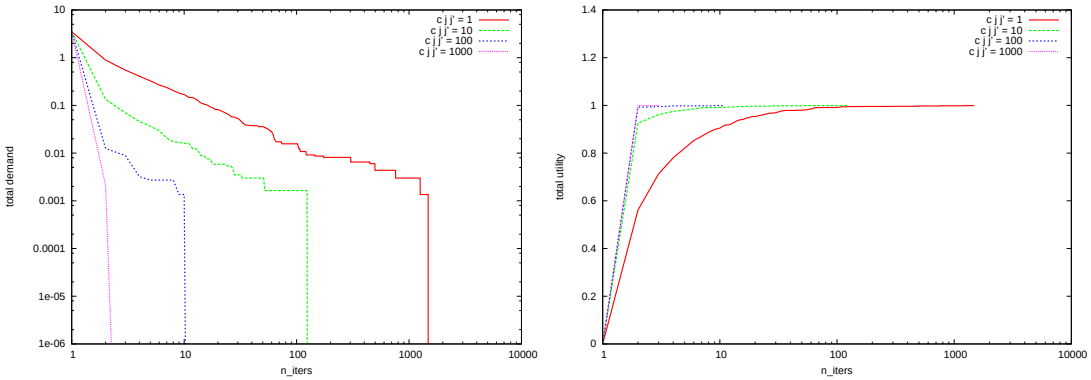

Figure 1: Transition function based on **limited** cooperative elementary trade: total demand (left) and total utility (right) versus number of iterations for sector-to-sector trades upper bounds of 1, 10, 100 and 1000.

In the unlimited case $h = h1\,d$, we expect the iteration to converge towards stationary, sub-optimal stocks[3]. The results shown in figure 2 are not in contradiction with such expectation. In this experiment the utility profile u and the initial stocks y_0 allow unique equilibrium stocks y_e for any vector of (strictly positive) prices p_0. Under these conditions time history of the total demand (on the left of the figures above) should be essentially the same as the time history of the distance

$$\left(\sum_{i=0}^{i<n_A}\sum_{j=0}^{j<n_G}(y\,t\,i\,j - y_e\,i\,j)^2\right)^{\frac{1}{2}}$$

between the actual stocks $y\,t$ and the equilibrium stocks y_e. Figure 3 shows that this is the case.

In the second experiment we consider the number of iterations needed for the total demand to be less than 10^{-6}. First we fix the ratio between the sector-to-sector trades upper bound and the number of agents per sector to 0.05 and let the number of agents vary. Then we fix the sector-to-sector trades upper bound to 10 and vary the number of agents. The results are reported in figure 4.

[3]remember that, in an unlimited trade between any two agents of indexes i and i', nothing prevents the i'-th agent from receiving an amount of the $(g\,i)$-th good which exceeds its demand or from giving away an amount of the $(g\,i')$-th good larger than its offer. This leads to sub-optimal allocations that cannot be reversed in subsequent trades. This is because elementary trades are non-decreasing (in the quantities of non-own-goods) and non-increasing (in the quantity of own-good).

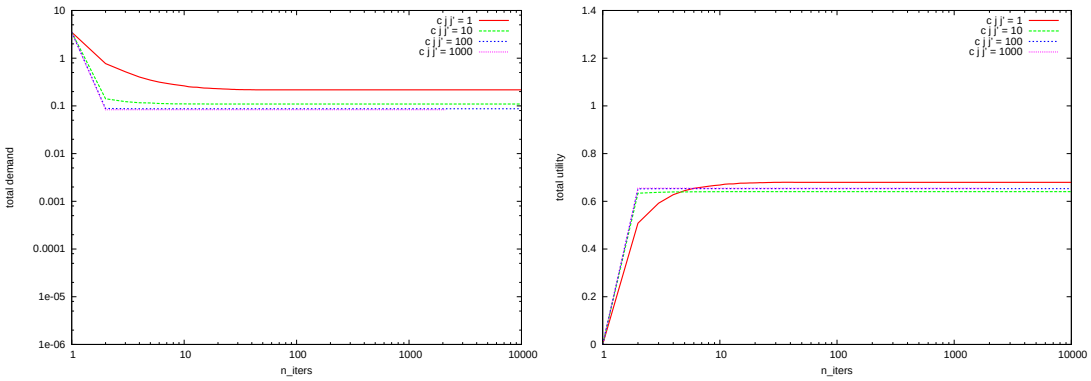

Figure 2: Transition function based on **unlimited** cooperative elementary trade: total demand (left) and total utility (right) versus number of iterations for sector-to-sector trades upper bounds of 1, 10, 100 and 1000.

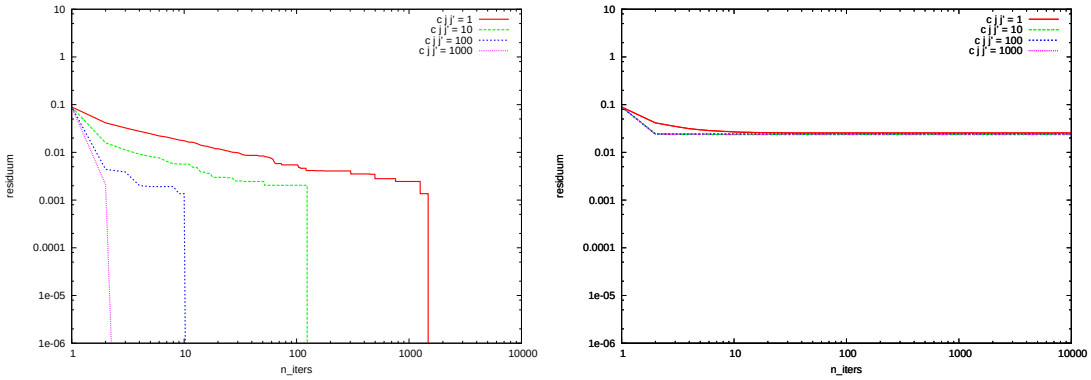

Figure 3: Transition function based on cooperative elementary trade: residuum versus number of iterations for sector-to-sector trades upper bounds of 1, 10, 100 and 1000; limited (left) and unlimited (right) case.

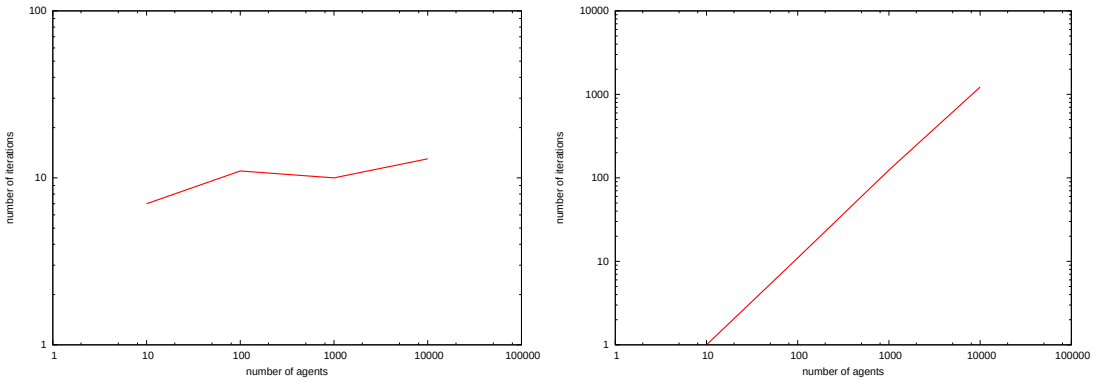

Figure 4: Elementary trade based transition function, number of iterations needed for convergence versus number of agents: constant ratio between sector-to-sector trades upper bound and number of agents per sector (equal to 0.05, left) and constant sector-to-sector trades upper bound (equal to 10, right).

TODO: discuss the adequacy of the monotonicity conditions and introduce the next section

5 The dynamics of prices driven by stocks utilities

6 Conclusions, outlook

A

Because tr is obtained by folding demand-based cooperative trades and p is constant, its flux is stocks value preserving. This means that

$$z' = tr\, z \;\Rightarrow\; \forall i \in [\,0, n_A\,) \quad (z'\, i) \cdot (p\, i) \;=\; (z\, i) \cdot (p\, i)\;.$$

For the same reason, the flux of tr is also non-own-good non-decreasing:

$$z' = tr\, z \;\Rightarrow\; \forall i \in [\,0, n_A\,), j \in [\,0, n_G\,), j \neq g\, i \quad z'\, i\, j \geq z\, i\, j\;.$$

These two equations together with the definition of natural demand (40) imply

$$z' = tr\, z \;\Rightarrow\; \forall i \in [\,0, n_A\,), j \in [\,0, n_G\,), j \neq g\, i \quad d\, u\, g\, p\, z'\, i\, j \;\leq\; d\, u\, g\, p\, z\, i\, j$$

and, by definition of total demand (45), (47).

B

One has:

$$
\begin{array}{l}
td\,d\,u\,g\,p\,(y\,t) \;=\; 0 \\
= \{\text{equation (45)}\} \\
\sum_{i=0}^{i<n_A} \sum_{j=0}^{j<n_G} d\,u\,g\,p\,(y\,t)\,i\,j \;=\; 0 \\
= \{\text{positivity of } d\} \\
\forall i \in [\,0, n_A\,), \forall j \in [\,0, n_G\,), \quad d\,u\,g\,p\,(y\,t)\,i\,j \;=\; 0 \\
= \{\text{take } j = g\,i'\} \\
\forall i \in [\,0, n_A\,), \forall i' \in [\,0, n_A\,), \quad d\,u\,g\,p\,(y\,t)\,i\,(g\,i') \;=\; 0 \\
= \{\text{equations (30), (38)}\} \\
\forall i \in [\,0, n_A\,), \forall i' \in [\,0, n_A\,), \quad h\,d\,u\,g\,p\,(y\,t)\,(i, i') \;=\; y\,t \\
= \{\text{equation (43)}\} \\
y\,(t+1) \;=\; y\,t
\end{array}
$$

C

$$
\begin{array}{l}
\sum_{i=0}^{i<n_A} u\,i\,y_e \\
= \{\text{equation (7)}\} \\
\sum_{i=0}^{i<n_A} \min_{j \in [\,0, n_G\,)} \frac{y_e\,i\,j}{w\,j} \\
= \{\text{equation (49)}\} \\
\sum_{i=0}^{i<n_A} \min_{j \in [\,0, n_G\,)} \frac{(y_0\,i)\cdot p_0}{w \cdot p_0} (w\,j) \frac{1}{w\,j} \\
= \{\text{equation (44)}\} \\
\sum_{i=0}^{i<n_A} \frac{1}{w \cdot p_0} \frac{1+g\,i}{|A_{g\,i}|}\,(p_0\,(g\,i)) \\
= \{\text{counting}\} \\
\frac{1}{w \cdot p_0} \sum_{j=0}^{j<n_G} (1+j)(p_0\,j) \\
= \{w\,j = 1 + j \text{ (hypothesis)}\} \\
1
\end{array}
$$

References

[1] A. Aho, J. Ullman, and J. Hopcroft. *Data Structures and Algorithms*. Addison-Wesley, 1982.

[2] W.B Arthur, V. Morrison, and S.N. Durlauf, editors. *The Economy as an Evolving Complex System II*. Santa Fe Institute Studies in the Sciences of Complexity Lecture Notes, 1997.

[3] R. Bird. *Introduction to Functional Programming using Haskell*. International Series in Computer Science. Prentice Hall, second edition edition, 1998.

[4] G. Debreu. *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. Cowles Foundation Monographs Series, Yale University Press, 1959.

[5] S.J. DeCanio. *Economic Models of Climate Change: A Critique*. Palgrave Macmillan, New York, 2003.

[6] S. L. Peyton Jones et al. *Haskell 98 Language and Libraries: the Revised Report*. Cambridge University Press, 2003.

[7] D. Gale. Bargaining and competition part i: Characterization. *Econometrica*, 54:785–806, 1986.

[8] H. Gintis. The emergence of a price system from decentralized bilateral exchange. *B. E. Journal of Theoretical Economics*, 6:1302–1322, 2006.

[9] H. Gintis. The dynamics of general equilibrium. *Economic Journal*, 17:1280–1309, 2007.

[10] C. Jaeger, A. Mandel, S.Fürst, W. Lass, and F. Meissner. Lagom generic: A minimal description from an economic point of view. *submitted to the ECF-GSD Modeling Workshop on Agent-Based Modeling for Sustainable Development, Venice 2-4 April 2009*, 2009.

[11] D. Kahneman and A. Tversky, editors. *Choices, values and frames*. New York: Cambridge University Press., 2000.

[12] F. W. Lawvere and R. Rosebrugh. *Sets for Mathematics*. Cambridge University Press, 2003.

[13] L.Walras. *Éléments d'économie politique pure, ou théorie de la richesse sociale (Elements of Pure Economics, or the theory of social wealth)*. Corbaz, Lausanne, 1874.

[14] A. Mandel, S. Fürst, W. Lass, F. Meissner, and C.Jaeger. Lagom generiC: an agent-based model of growing economies. *ECF working paper*, 1, 2009.

[15] A. Mas-Collel. *The Theory of General Economic Equilibrium: A Differential Approach*. Cambridge University Press, Cambridge, 1985.

[16] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.

[17] H. Scarf. Some examples of global instability of competitive equilibrium. *International Economic Review*, 1:157–172, 1960.

[18] I. Sommerville. *Software Engineering*. Addison-Wesley, 8th edition edition, 1992.

[19] H. Sonnenschein. Do walras identity and continuity characterize the class of community excess demand functions? *Journal of Economic Theory*, 6:345–354, 1973.

[20] L. Tesfastion and K. Judd, editors. *Handbook of Computational Economics II: Agent-Based Computational Economics*. North-Holland, 2006.