

## AGENT-BASED MICRO-SIMULATIONS IN IMMUNOLOGY AND FINANCE

### *Mikrosimulēšana ar āģentu izmantošanu imunoloģijā un ekonomikā*

F. Castiglione

Istituto Applicazioni del Calcolo (IAC) "M. Picone"

Consiglio Nazionale delle Ricerche (CNR)

Viale del Policlinico, 137 - 00161 Rome, Italy

filippo@iac.rm.cnr.it

#### Abstract

*In the search for computational models that help to understand the dynamics of Complex Systems, one can take a great advantage from the impressive acceleration of computer tools and techniques. In fact the very structure of computation on digital computers has inspired the introduction of new class of models (algorithms), where interaction among degrees of freedom are expressed by logical rules acting over a discrete state space – something much closer to "biological language" than to standard (floating point) physical models.*

*Starting from the definitions of spin systems, with little changes we reach a definition a new model that is well suited to describe different simulation systems. Such class of models is can be considered a subclass of the Agent-Based systems in vogue nowadays.*

*Moreover, we shortly describe two microscopic simulators of this type, which are being used to study microscopic phenomena in two completely different fields of application, namely immunology and finance.*

*As a final remark, given the lattice representation of space, such computational-modeling paradigm is well suited for efficient and "relatively simple" parallelization. Indeed, both models have been implemented to run on parallel computers adopting the Message Passing paradigm for Distributed Memory machines.*

**Keywords:** *micro-simulation, agents, cellular automata, immunology, finance.*

#### Introduction

Since the advent of digital computers, the way research proceeds has dramatically changed. The study of physical systems for example, is traditionally investigated by means of mathematical models. These models are often very difficult to be solved analytically and approximations are necessary to reach a solution.

The use of computers has brought great advantages in handling complex models in two different although complementary ways: (1) the mathematical formulation of a model can be solved numerically using sophisticated *algorithms* to find a good "numerical" solution; (2) a system can be analysed in term of its constituents, i.e., the overall dynamics can be *simulated* by its very microscopic elements and global quantities can be compared with experimental data.

Among the two, the first field is by no means more mature. In fact, since the fifties, a huge amount of methods have been developed and many more are currently under study. Books and articles describing what is considered "standard literature" on *Numerical Calculus* are largely available nowadays.

Traditionally, also the term "simulation" refers to numerical methods to compute, for example, the solution of a system of partial differential equations. In contrast, the definition that will be used throughout this manuscript will point to a different narrower meaning.

*To simulate a system means to reproduce the "functional behaviour" of the constituents under particular laws which rule the global dynamic of the system itself. These laws are not known in general and are exactly the target of the investigation. Even the variables defining a single system-constituents are not known in general.*

This approach is much more valuable as the computing power of today's computer increases. In fact the number of *micro-constituents* of the system should be, by definition, large [1].

*Complex behaviour can occur in any system made up of large numbers of interacting constituents with non-linear coupling, be they atoms in a solid, cells in a living organism, or traders in a financial market.*

It is the availability of digital computers that makes possible to solve sophisticated models and, in so doing, to reveal the micro-dynamics of some complicated natural phenomena.

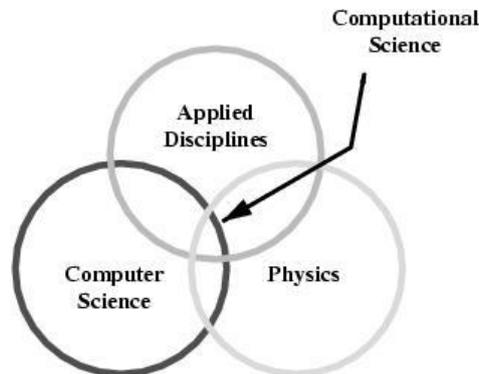


Fig. 1 Microsimulation (MS [2]) or Agent-Based methods belong to the Computational Sciences and mainly refers to methods similar to those used by computational statistical physics that are being applied to other disciplines [3]. The Computational Sciences are identified as the intersection of physics methods and applied disciplines, with the use of computers as combining element.

From the mere technical point of view, the Moore's law (computer power doubles every 1.5 years) assures increasing memory and CPU speed to simulate larger and larger systems. Although many problems are now solvable by common personal computers, there are problems for which even the largest parallel machine is not able to find a “real” solution. These problems are called *Grand Challenges*. A Grand Challenge is a large-scale science or engineering computational problem. Examples can be found in Physics, Biology, Chemistry, Materials Sciences, Fluid and Plasma Dynamics, Finance, Environment and Earth Sciences, and so on.

The reason of the intractability of such problems lies in the level of details one wants to take into accounts. For example one may think of a biological cell as a single element ruled by very simple dynamics which brings the cell at most into one excitatory state. In this scenario the amount of memory needed to represent a cell is reduced to the minimum (a bit) and the number of operations to test and possibly switch its state is negligible. Stated like this, even if we wanted to take into account millions of cells at one time we would not run into troubles today if we could access a reasonable workstation. Problems arise when we want to go into the details of the cell. In fact, a cell is a whole universe for its own, with an unthinkable level of details. Even top supercomputers would not be able to represent all that information. So, what chances do we have? The possibilities stay in between: we cannot take all the details at once but we can add them one after another as long as the computing power required is available. This philosophy has gone long enough today to allow sufficiently-detailed simulations of complex phenomena.

The present manuscript deals with the author experience with complex systems composed by many interacting elements. In particular it deals with the definition of a particular class of models derived from the spin or lattice models of physics and the presents two examples with regards to problems from biology and finance. In both fields the rules governing the micro-behaviour of the constituents (cells, molecules but also traders and

brokerage agencies) are mostly unknown. All that is given is the macro-behaviour that can be observed empirically either by experiments (this is the case in biology) or by applying statistical methods to the already given data (this is the case in finance where the use of databases allows to track any transaction worldwide).

The joining element between the two MS models described herein is not only conceptual but also technical in certain respects. In fact they are both being coded following a precise architectural schema. Such schema is, according to the author opinion, just the natural extension of the well known in the physicists community spin or lattice models.

### **From spin to automata computational models**

Nowadays, the scientific study of a phenomenon in general consists of three major approaches: theoretical, experimental and computational. The computational aspect becomes more and more important. Computational science has the flavor of both theoretical and experimental science. One must have a very good theoretical background to study a subject by means of computational methods. A good computational method often comes from a thorough theoretical analysis. On the other hand, the analyses of results are not much different from analysing experimental data. Computational methods in science become advantageous when (1) the problem at hand is too difficult to do analytically; (2) an approximate theoretical result may not be reliable, and it is necessary to check with a different method; (3) an experiment is expensive or not feasible at all.

As already mentioned in the introduction, computational methods can be roughly divided in two areas, that of numerical analysis and that of computer simulation. Numerical methods include solving linear equations, eigenvalue problems, solving differential equations and partial differential equations, etc. In contrast, computer simulations are methods that try to model the physical world *directly*, rather than solving the equations governing the physical processes.

A complex systems (physical, biological, chemical or financial, just to mention a few) can be defined as a system with a large number of degrees of freedom. Thus, microsimulation (MS) is a method to mimic a complex phenomenon through the description of its micro-components. That is, leaving the system free to evolve without too many constraints and simplifying assumptions.

In the following we present techniques belonging to the class of MS methods. These have been applied with success in the field of statistical mechanics [4]. Some of them are being also used in biology and recently in finance [2, 3]. At the end we define a new class of MS models that we use throughout this manuscript to describe two different simulation algorithms dealing respectively with problems from immunology and finance.

The very first definitions are those of “spin” and “lattice”. A ferromagnet can be regarded as a system composed by a large number of elementary magnets placed on the sites of a crystal lattice. To model and to understand the magnet properties of solids, various types of *lattice spin models* have been proposed. Such models are defined by

- (i) a lattice type (dimension 1,2,3..., and topology, i.e., cubic, triangular and so on);
- (ii) the possible values of the random variable, called “spin” at each lattice site, that is, the number of possible states a spin can take (these may be either discrete or continuous; also, a spin can be in generalized sense, a single value, a vector or a tensor, although some representations may lack of a physical meaning);
- (iii) the interactions among spins, in terms of rules determining the way the value of the spins are coupled and how they change with time.

Iterating the interaction rules, one gets a discrete dynamical system.

The following overview of spin systems is by no means complete. Moreover, we voluntarily avoided talking about the interaction rules, which, together with the topological

definition of a spin system, is the most important element to distinguish one application from another.

If the total number of lattice sites is  $L$ , we identify a spin with a stochastic variable  $s_n$ , for  $n=1,2,\dots,L$ . According to (ii) one deals with different spin models. The most popular are the enunciated below.

### Ising model

A spin may take on just two values; ‘‘up’’ or ‘‘down’’, usually  $s_n=+1$  or  $s_n=-1$ .

### Potts model of the $K^{\text{th}}$ order

It is a generalization of the Ising model with the spins taking one of  $K$  possible values, i.e.,  $s_n=1,2,3,\dots,K$ .

### XY model

Each spin is a complex number of absolute value 1, i.e.,  $s_n = e^{i\phi_n}$ .

Later, other models have been derived from these definitions.

### Cellular Automata (CA)

Inspired by the early work of J. von Neumann [5] on self-replicating machines, they are discrete dynamical systems where each spin per lattice site is updated according to the state of the spin in its neighborhood [6]. CA are being used to model many physical systems but seem more suited to model biological systems.

### Lattice Gas Automata (LGA)

Lattice gas automata were introduced by Frisch, Hasslacher and Pomeau as a means to solve the Navier-Stokes equations of fluid dynamics. The two dimensional triangular lattice gas (fig. 2) is indeed called FHP lattice [7]. A lattice gas is like the Potts model in which the states of particles represent velocities. The FHP model is associated to particular types of lattices with peculiar interaction rules (collisions conserving mass and momentum). The FHP model is a two-dimensional triangular lattice, thus the number of velocities  $K$  is equal to six.

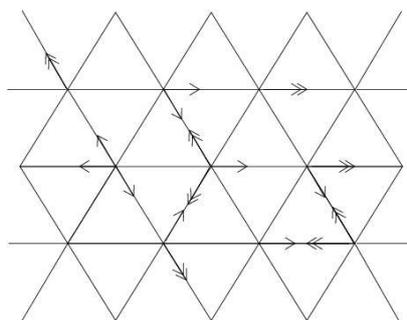


Fig. 2 Triangular lattice. Particles at time  $t$  and  $t+1$  are marked by single and double arrows, respectively.

### Integer Lattice Gas Automata (ILGA)

A generalization of lattice gas has been proposed with the name of Integer Lattice Gas Automata [8]. The generalization is given allowing more than a single particle per each direction to stay on a lattice site, e.g.,  $s_n \in \{1,\dots,K\}^r$  with fixed number of particles per site equal to  $r$ .

Through this step-wise definition of spin models we come to the definition of a spin system in which each site of the lattice contains exactly  $r$  particles, each in one of  $K$  different states-velocities. We still need, for our purpose, two further generalizations:

- (1) we want to represent different particles types each having their own micro-state space;
- (2) we do not want a fixed number of particles on each lattice site.

The first of these requirements leads to a definition of spin with many components  $s_n = \bar{s}_n = (s_n^1, s_n^2, \dots, s_n^E)$  with  $E$  the number of different types of entities (we also use the term *entity* in place of particle; entity fits better the meaning of (e.g.) agents in a stock market or cells in the immune system). Each agent belonging to class  $e = 1, \dots, E$ , can be found in a micro-state taken from a *discrete* set of states  $1, \dots, K(e)$ , whose number depends on  $e$ .

The second point requires to consider  $r \equiv r(x, t)$  that is, the number of entities on each lattice site  $x$  is a function of time  $t$  and position  $x$ . In general, the number of entities in a given lattice site depends on the diffusion process we choose. But in practice it is sufficient to choose  $r$  as the maximum number of particles on the lattice during the whole simulation to recover the definition of integer lattice gas.

Because we allow  $r(x, t)$  growing without constraints we set apart this particular case of *unbounded capacity* and call it “unbounded” lattice gas.

### Unbounded Lattice Gas (ULG)

It is a lattice gas with unlimited number of particles on each lattice site. Particles belong to different classes  $e = 1, \dots, E$ . They may take on one micro-state from a set  $1, \dots, K(e)$  which in turn depends on the class  $e$  to which they belong to.

Summarizing, we are able now to define models where the different entities belongs to different classes. They occupy the lattice sites with no constraint on the number. They interact *locally* instead of interacting with the neighborhood as in CA models. Finally, eliminating the constraint on the occupation number we allow the particles to diffuse freely on the lattice grid. At this time we consider the particles to follow the classical *Brownian* motion. General non-uniform diffusion schema is also well defined thanks to the unconstrained capacity  $r(x, t)$ .

As already anticipated above, the use of ULG as formal definition of the microsimulation systems developed and discussed thereafter is justified by the availability of large-memory computers. Instead of storing a single bit like in the Ising model or at most few bytes as in the Potts model to keep the memory consumption at minimum, we can now represent *particles* (cells, atoms, molecules, traders etc.) as a collection of information or *attributes*. Thus, the informative structure representing a single particle is heterogeneous as we allow mixing binary information, integer numbers or even arrays of more complicated records. Note that we intentionally restricted ourself to the use of integer numbers to represent the internal states. The reason is to avoid floating points operations to assure unconditional *numerical stability* to the simulation algorithm.

Moreover, given the unbounded capacity of the lattice, the choice of a *static* data structure is clearly wrong. Indeed, a *dynamic* memory allocation is much more appropriate. In practice, all we need to represent a  $d=2$  ULG is a *pointer* to a list of “records” containing the information structure of the entities *for each* lattice site.

In our models the complex behaviour of the entities is subjected to precise *state-changes* upon interaction. Every single entity can be thought as a *Stochastic Finite State Machine* (SFSM) [9] which processes information and changes its state according to the result of the interaction with other entities, or with external fields.

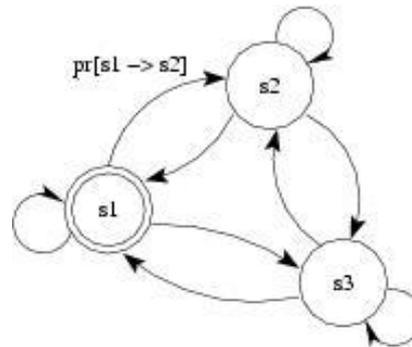


Fig. 3. Stochastic Finite State Automata. The probability  $pr[s1 \rightarrow s2]$  to switch from state one to state two, for example, can be given or can be the outcome of a more complicated procedure. In our case these probabilities are computed by complex rules.

Probabilistic or stochastic models should not be confused with non *deterministic* models in theoretical computer science [9]. A typical example of stochastic system is a Markov chain where each state transition is subject to a given probability.

A typical diagram showing a stochastic finite state (automata) is given in fig. 3. The transition between the states  $s_1$ ,  $s_2$  and  $s_3$  is stochastic. The transition probabilities can be fixed or changing in time. In our case they depend on the outcome of *more or less* complicated interaction rules between entities. They can also depend on some global quantities or external fields.

Finally, because the particles interact locally (i.e., inside each lattice site) and only after they diffuse to adjacent sites, we can “easily” divide the CPU-load distributing the lattice grid to different processors of a parallel machine [10]. Message passing among processors is needed only during the diffusion phase. This allows us to simulate a large number of interacting entities with a high level of details.

### Application I: Immunology

The Immune System response has been modeled by means of a generalized cellular automaton:

- (1) the CA is defined on a triangular 2D lattice with periodic boundary conditions (toroidal geometry).
- (2) the dynamics is probabilistic
- (3) the evolution of each site depends just on the site itself (internal dynamics)
- (4) entities move from site to site (diffusion process)
- (5) each time step corresponds to about 8 hours of “real life”.

From the viewpoint of statistical mechanics, the CA can be considered as an extended lattice gas. There is neither a partition function nor any “classic” observable but the entropy.

In this model we map a single lymph node (or generically a small portion of a secondary lymphoid organ) of a vertebrate animal onto a two-dimension hexagonal lattice  $L \times L$ , with periodic boundary conditions in both directions. The primary lymphoid organs thymus and bone marrow are modeled apart: the thymus is implicitly represented by the positive and negative selection of immature thymocytes before they get into the lymphatic system, whereas the bone marrow generates already mature B lymphocytes. Hence, on the lattice we find only immunocompetent lymphocytes. In this model the cells (and to a certain extend also the molecules) represents the agents of the simulation, sometimes also referred to as entities. Agents interact and cooperate to build the main functions of the immune system, namely, recognition, defense and memory.

This model belongs to the class of *bit string models* [11]: the affinity among the entities is described in terms of *complementarity* between binary strings having a fixed directional reading frame. Bit strings represent the “binding site” of cells and molecules as for example lymphocyte receptors (T lymphocytes receptor TCR, B lymphocytes receptor BCR), Major Histocompatibility Complexes MHC, antigen peptides and epitopes, immunocomplexes, etc. The model includes the major classes of cells of the lymphoid lineage (T helper lymphocytes or TH, cytotoxic T lymphocytes or CTL, B lymphocytes and antibody-producer plasma cells, PLB) and some of the myeloid lineage (macrophages, MA, and dendritic cells, DC). The interactions among these cells define their functional behaviour. Rules accounting for phagocytosis, digestion and presentation of the antigen, opsonization, lymphocyte recognition and cooperation, viral infection, cytokine secretion etc. have been defined.

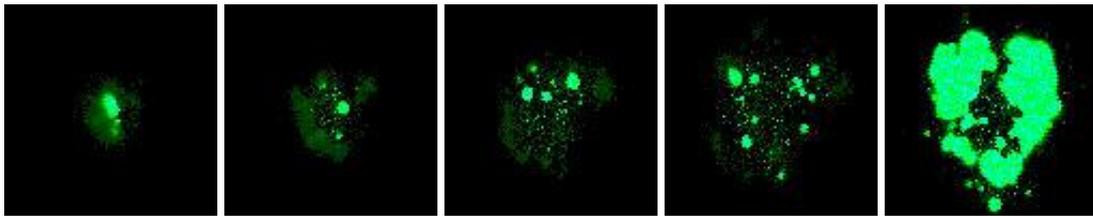


Fig. 4. Virus diffusion in a 100x100 lattice at various instants (time goes from left to right).

Figure 4 shows five snapshots of a viral infection and spreading on the lattice representing the lymphnode. Formation of areas of low antigen density corresponding to the presence of anti-virus immunoglobulins are visible.

The model allows for the study of various phenomena and the validation of different immune mechanism [12,13].

## Application II: Financial Markets

We have constructed an agent-based model to simulate the dynamic interactions between market price and the decisions of different kind of traders. The present model incorporates many features of other known models and is meant to be the first step toward the construction of an agent based model that uses more realistic markets rules, strategies, and information structures.

Agents represent traders in a “virtual” stock market. They can group together to form coalitions. Each coalition follows a strategy chosen from a proportional voting “dominated” by a leader's decision. The interplay of the different kind of trading strategies gives rise to complex price dynamics that is consistent with the main stylized facts of financial time series. The main goal is to give an easy way to implement different key issues in modeling the stock market, to understand the relevance and the mutual influence of certain factors that other models have treated separately, and to investigate the necessary and sufficient conditions determining the factors, which actually drive the empirical observed facts in real markets.

Recalling the introduction, the computational model described herein is formally equivalent to an *unbounded lattice gas*.

The model represents [14] three kinds of agents (or agents' strategies) trading for a set of  $N$  assets (stocks): *fundamentalists*, *noisy* and *technical traders* [15, SteiglitzHonigCohen:96]. The former consider a reference (or “fundamental”) value to determine the “right” price of an asset. The noisy are most of the “small” traders, which do not follow any reference value and do not look at charts. Their behaviour is mostly random. Lastly, technical traders are those who take into account information about the evolution of

the price, (in our case the *moving average* over certain horizons of time [16]). These are the traders that we name *chartists*.

At each time step each agent decides, if to be active or not to trade. For active traders, a different decision path is followed depending on the trading strategy. A different decision is taken for each stock on the market.

To model how the decisions of agents are influenced by their mutual interaction, we assume that the decision process undergoes a proportional voting where agents occupying the same lattice site express their preference. The single agent's decision is weighted by its *influence strength* to form the collective decision.

Finally, the model implements a quasi-realistic book-of-orders as lists of buy-orders and sell-orders are added. Matching orders on the two lists are satisfied (filled) while the remainings wait for a pair matching order to be inserted.

The model reproduce realistic crossover of the price change of an asset to Gaussian behaviour for increasing activity of traders corresponding to a larger time resolution of the simulation (figure 5).

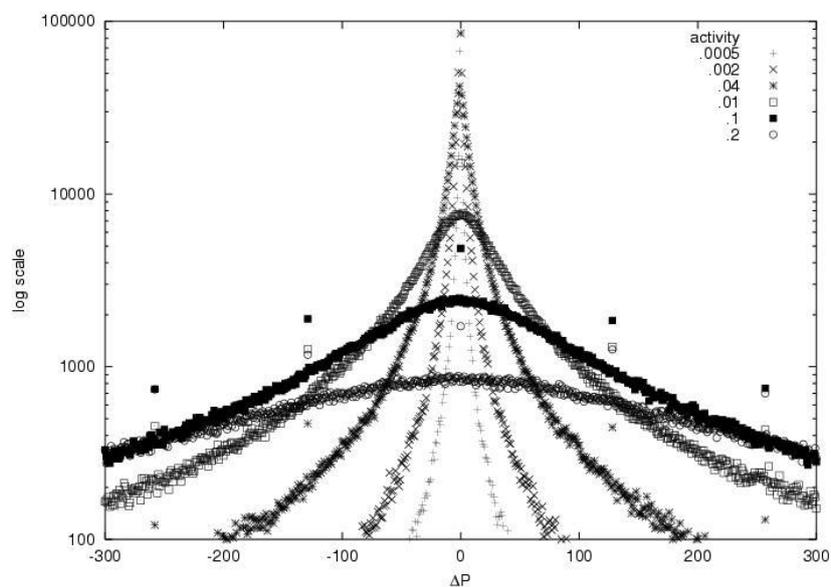


Fig. 5. Crossover to Gaussian regime of the price difference for increasing activity of the traders, i.e., decreasing time resolution of the simulation step.

It applies to an index defined on all simulated stock prices. Again, the model is consistent with fat tails of histogram of the normalised price change (the returns, see fig. 6), log-normal distribution and clustering of volatility (not shown).

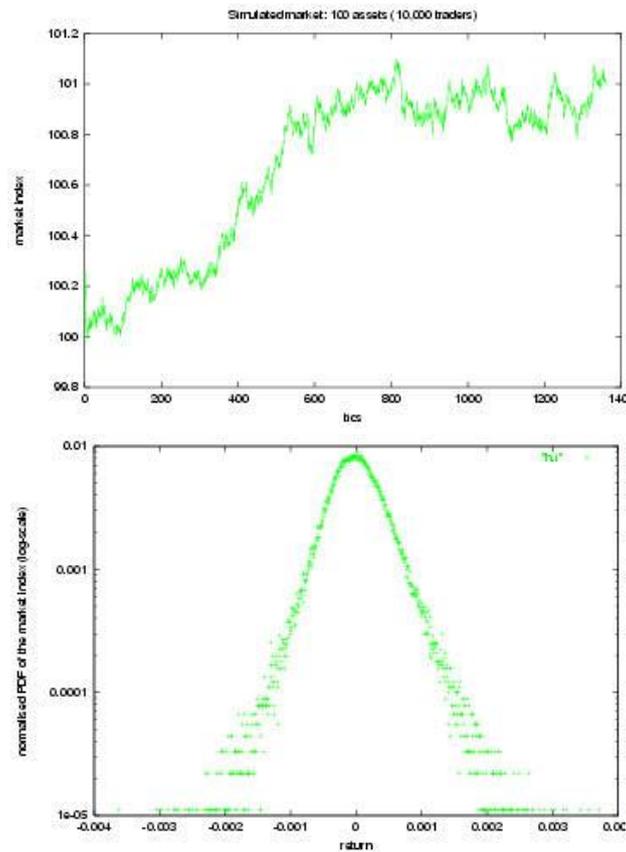


Fig. 6. Simulated market index considering 100 assets. Index dynamics (left) and its return distribution (right).

### Parallel programming

The approach to the parallelization of both the above computational models is very similar. We have used MPI [17] mainly to exploit the better support for Collective Communication Primitives (CCP's) that MPI offers. In the present code, CCP's are used to perform many *reduce* operations in parallel instead of collecting all data on a single node and then process them sequentially.

The scheme, which is shown in fig. 7, can be summarized as follows. Each task of a parallel run is in charge of a subset of the total number of agents or cells according if we are talking of the financial model or the immunological model. The distribution is made according to the distribution of agents on the lattice.

All phases of the simulation are executed in parallel and there is no dependency on the total number of tasks. There are two main phases in which the tasks interact and diffuse. Then the output phase, which is performed by a single processor that we identify as the master.

During the diffusion, agents may migrate from a task to another and the communication is point-to-point. All receive operations are *posted* in advance, to avoid any dependency on the internal buffering mechanisms of MPI. To evaluate global quantities required by all tasks (e.g., the price change or the total number of cells) CCP's are employed. The same technique is applied when data are collected from all tasks before writing results to the files.

The efficiency of the parallel code depends strongly on the number of agents. This is not surprising since the overhead of the implicit synchronization required by the CCP's is, for few agents, greater or equal to the speedup due to the parallel processing.

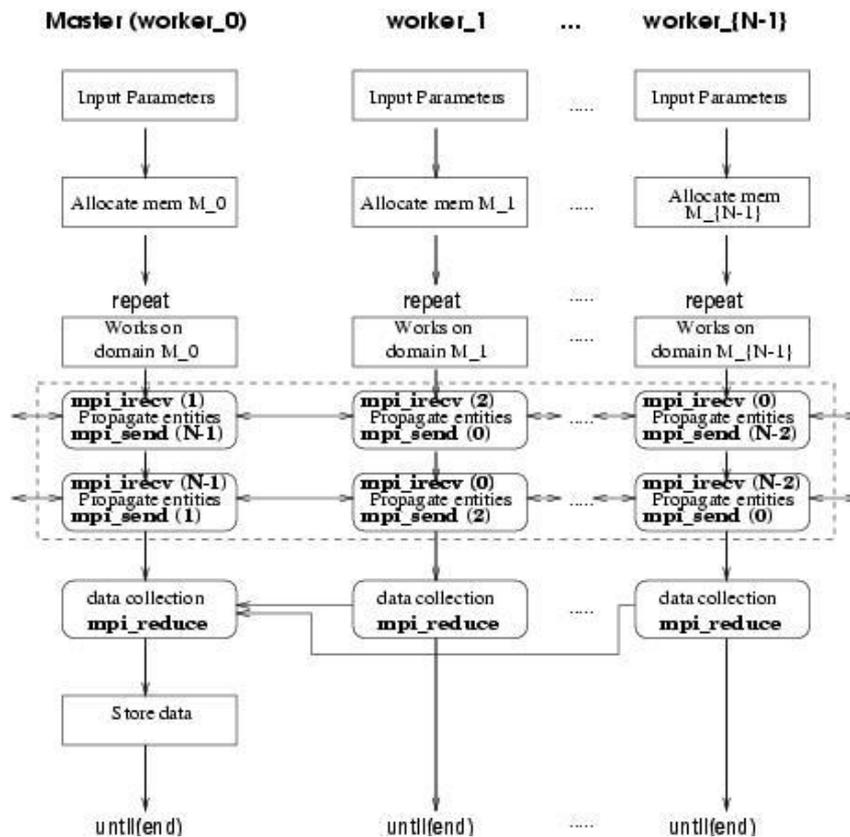


Fig. 7. Communication scheme. The numbers in parentheses represent the source, for the `mpi_irecv`, or the target, for the `mpi_send`, of the point-to-point communication operations (e.g., `mpi_irecv(1)` means receive message from worker<sub>1</sub>, `mpi_send(N-1)` means send message to worker<sub>(N-1)</sub>).

Owing to the MPI portability, exactly the same code runs on pretty different platforms like the Sun Enterprise 10000, the Cray T3E and the IBM SP2.

### Conclusions

In the present manuscript we describe a new way of thinking at the Agent-based paradigm of modern simulation. This paradigm can be used to model systems when the knowledge of the rules governing the dynamics are mostly unknown.

Through a step-wise definition of spin models, we have derived a general definition of Unbounded Lattice Gas. The definition itself has no practical use but presenting the common denominators of two different models to simulate biological and financial processes. Some technical issues as well as the schema of the inter-process communication in the parallel version of the two models has been briefly described as an example of how computing and communication may be distributed among processors to improve the performance of the code. Something, which is needed to solve realistic-size problems.

### References

1. Auyang, S. *Foundations of Complex-System Theories in Economics, Evolutionary Biology and Statistical Physics*. Cambridge Univ. Press, Cambridge, UK, 1998.
2. Levy, H., Levy, M., and Solomon, S. *Microscopic Simulations of Financial Markets*. Academic Press, New York, NY, 2000.
3. De Oliveira, S., De Oliveira, P., and Stauffer, D. *Evolution, Money, War, and Computers - Non-Traditional Applications of Computational Statistical Physics*. Teubner, Stuttgart-Leipzig, 1999.

4. Beck, C., and Schlogl, F. *Thermodynamics of chaotic systems*. Cambridge Univ. Press, Nonlinear Science Series, Cambridge, UK, 1993.
5. Von Neumann, J., and Burks, A. *Theory of Self-Reproducing Automata*. Urbana: U. Ill. Press, 1966.
6. Wolfram, S. *Cellular Automata and Complexity*. Addison Wesley, New York, NY, 1994.
7. Frisch, U., Hasslacher, B., and Y.Pomeau. *Lattice-gas automata for the Navier-Stokes equation*. Phys. Rev. Lett. **56**:1505-1508, 1986.
8. Boghosian, B., Yepez, J., Alexander, F., and Margolus, N. *Integer lattice gases*. Phys. Rev. E **55**:4137-4147, 1997.
9. Mandrioli, D., and Ghezzi, C. *Theoretical Foundations of Computer Science*. Krieger Publishing Company, Melbourne, Florida, 1993.
10. Quinn, M. *Parallel Computing : Theory and Practice*. McGraw-Hill, New York, NY, 1993.
11. Farmer, J. D., Packard, N., and Perelson, A. *The immune system, adaptation and machine learning*. Physica D **22**:187-204, 1986.
12. Celada, F., and Seiden, P. *A computer model of cellular interaction in the immune system*. Immunology Today **13**:56-62, 1992.
13. Celada, F., and Seiden, P. *Affinity maturation and hypermutation in a simulation of the humoral immune response*. Eur. J. Immunol. **26**:1350, 1996.
14. Castiglione, F. *Diffusion and aggregation in an agent based model of stock market fluctuations*. Int. J. Mod. Phys. C **11**:865-880, 2000.
15. Lux, T., and Marchesi, M. *Scaling and criticality in a stochastic multi-agent model of financial market*. Nature **397**:499-500, 1999.
16. Bernaschi, M., and Castiglione, F. *Effect of technical traders in a synthetic stock market*. Int. J. Mod. Phys. C **11**:1437, 2000.
17. The Message Passing Interface standard. [www-unix.mcs.anl.gov/mpi/](http://www-unix.mcs.anl.gov/mpi/).