

1 **A Linux cluster of personal computers for the numerical simulation of**  
2 **natural airflows in greenhouses using a lattice model**

3  
4 J.E. Jiménez-Hornero <sup>a</sup>, F.J. Jiménez-Hornero <sup>b,\*</sup>, J.V. Giráldez <sup>b</sup>

5  
6 <sup>a</sup>Dept. of Computing and Numerical Analysis, University of Cordoba  
7 C2 Building, Campus Rabanales, 14071 Cordoba, Spain

8  
9 <sup>b</sup>Dept. of Agronomy, University of Cordoba  
10 P.O. Box 3048, 14080 Cordoba, Spain

11  
12 \*e-mail of corresponding author: ir2jihof@uco.es  
13  
14

15 **Abstract**

16  
17 Some numerical models need a considerable computational effort to run their  
18 simulations. This fact is translated into long execution times delaying the decisions that  
19 have to be taken by the modeller in order to obtain the best phenomena description. This  
20 situation is also frequent in agronomical modelling, particularly when the subject of  
21 study is the fluid flow pattern as happens when simulating the natural airflows in a  
22 greenhouse. The knowledge of the effects of different ventilator configurations allows  
23 the crop manager to improve the greenhouse's natural ventilation conditions. In this  
24 work, a Linux cluster of ten personal computers (PC) is proposed with the aim of  
25 reducing the execution time of the lattice model simulations by means of parallel  
26 computing. This model has been used for describing fluid flow patterns in the presence  
27 of solid obstacles since the 1990s. The lattice model structure makes them suitable for a  
28 straightforward parallel computing implementation. As Jiménez-Hornero et al. (2005)  
29 show, this model is suitable for simulating the two-dimensional natural airflow in the  
30 vertical cross-section of a tropical crop protection structure described by Montero et al.

1 (2001). The performance metrics clearly show the benefit of using the proposed Linux  
2 PC cluster in terms of execution time reduction and speedup with respect to the  
3 sequential running in a single PC.

4

5 *Keywords:* Linux PC cluster; Parallel computing; Lattice model; Natural airflows in  
6 greenhouses

7

## 8 **1. Introduction**

9

10 Natural airflow in greenhouses is a common method for cooling and controlling the  
11 CO<sub>2</sub> concentration and it has a direct influence on plant growing conditions. Several  
12 works have been devoted to describing the natural airflow velocity field in a greenhouse  
13 using computational fluid dynamics (CFD) models (e.g. Mistriotis et al., 1997; Montero  
14 et al., 2001; Bartzanas et al., 2002; Mistriotis and Briassoulis, 2002) or lattice models  
15 (e.g. Jiménez-Hornero et al., 2005). Both methods provide useful information to crop  
16 managers about the ventilation conditions in the greenhouse with much less time and  
17 cost than in the case of laboratory testing. Lattice models have been developed in the  
18 last decade of the 20th century, while CFD models have been used since the late 1960s.  
19 For this reason, commercial CFD codes (Xia and Sun, 2002) are more extended and  
20 better known than those corresponding to the lattice model approach. However, both  
21 alternatives present the same drawback from the modelers' point of view when they  
22 wish to answer the question 'what if' in a short time when large problem domains are  
23 analyzed or strict convergence conditions are considered. In that situation it is necessary  
24 to simulate a considerable amount of time-steps. One solution for reducing the  
25 simulation time is to use parallel computing, adapting the code to be executed on

1 several processors. This alternative has been applied as a method to speed up the CFD  
2 computations (e.g. Henderson et al., 2003). In this aspect, the lattice model is an  
3 appropriate approach because of its ease in producing the required parallel code due to  
4 its simplicity, flexibility, intrinsic parallelism and the use of simple and regular meshes  
5 (Chopard and Hoekstra, 2005).

6 Traditionally, parallel computing has been associated with the use of expensive  
7 supercomputers that belong to universities or research centers. However, the past few  
8 years have seen significant advances in microprocessor and network technologies  
9 (Grama et al., 2003, pp. 1-8; Dongarra et al., 2003). These advances have improved,  
10 among other features, the communications efficiency between single microprocessors  
11 allowing the use of personal computers (PC) clusters for parallel computing. According  
12 to Dongarra et al. (2005), Linux PC clusters are now used more often than other types  
13 of parallel computers because of their low cost, flexibility and accessibility. One of the  
14 most widely used libraries for parallel implementation with a Linux PC cluster is the  
15 message passing interface, MPI (e.g., Snir et al., 1998; Gropp et al., 1998). This  
16 interface has been developed for parallel programming with FORTRAN and C++ (e.g.  
17 Gropp et al., 1999; Karniadakis and Kirby, 2003). The aspects previously mentioned  
18 enable us to consider the Linux PC cluster as an appropriate alternative for doing  
19 simulations with lattice models by means of parallel computing.

20 Jiménez-Hornero et al. (2005) showed that the lattice model is a valid tool for  
21 describing the steady-state natural airflows in greenhouses. Their results are in  
22 acceptable agreement with the flow patterns reported by Montero et al. (2001) for a  
23 tropical crop protection structure with four ventilator configurations. The lattice model  
24 simulations were run using parallel computing on a SGI Origin 2000 server with eight  
25 MIPS R10000 processors (200 MHz). This multiprocessor computer is rather obsolete

1 and it had to be shared among other users. This fact notably reduced its computational  
2 power and increased its execution time. All the simulations took 15-20 minutes with a  
3 converge criterion applied for the velocity value between two consecutive time-steps of  
4 less than or equal to  $1 \times 10^{-5} \text{ m s}^{-1}$ . As an alternative, a homogeneous Linux PC cluster  
5 consisting of ten nodes or computers with identical features was proposed in this work.  
6 Several performances were achieved using a different number of nodes of the PC cluster  
7 to show the benefit of using parallel computing for the lattice model simulations.

8

## 9 **2. The lattice model**

10

11 Lattice models are numerical approaches developed for simulating fluid dynamics  
12 (Chen and Doolen, 1998; Wolf-Gladrow, 2000, pp. 7-14). These models use a work  
13 scale, called mesoscopic, which is between the microscopic and macroscopic levels. At  
14 this work scale, the reality is transformed into a regular lattice in which simple particles  
15 interact with each other following simple rules. As is shown in Fig. 1, partly based on  
16 Bernsdorf (2001), there are two main steps for these models: propagation and collision.  
17 In the propagation stage, the particles move from a lattice site to its neighbour in each  
18 time-step. When two or more particles move to the same site the collision stage occurs  
19 according to the conservation of particles masses and momentum (Chopard and Droz,  
20 1998, pp. 122-135; Rothman and Zaleski, 2004, pp. 73-75).

21 The lattice model known as the Bathnagar, Groos and Krook (BGK) model (Chen et  
22 al., 1992; Qian et al., 1992) was used in this work. This model is based on a  
23 simplification of the lattice Boltzmann model using the relaxation time parameter  
24 proposed by Bathnagar et al. (1954) in the context of gas molecule collisions. In the  
25 lattice BGK model the probability of finding a particle that is moving in the direction of

1 link  $i$ , that connects a node  $r$  with one of its neighbours, is represented by the  
 2 independent variable  $f_i$ , which varies continuously between 0 and 1 according to  
 3 Boltzmann's molecular chaos hypothesis. The main mesoscopic equation of this lattice  
 4 model for a node  $r$  at time  $t$  (Chen and Doolen, 1998; Chopard and Droz, 1998, pp. 122-  
 5 135; Wolf-Gladrow, 2000, pp. 139-144; Succi, 2001, pp. 65-73; Rothman and Zaleski,  
 6 2004, pp. 73-79) is:

$$7 \quad f_i(\mathbf{r} + \Delta t \mathbf{c}_i, t + \Delta t) = f_i(\mathbf{r}, t) - \frac{1}{\tau} (f_i(\mathbf{r}, t) - f_i^{\text{eq}}(\mathbf{r}, t)) \quad (1)$$

8 where  $\Delta t$  is the time-step duration,  $\mathbf{c}_i$  is the velocity of a particle in the link  $i$ ,  $f_i^{\text{eq}}$  is the  
 9 local equilibrium function, and  $\tau$  is the relaxation time parameter that is an indicator of  
 10 the difference between  $f_i$  and  $f_i^{\text{eq}}$ . Eq. (1) describes the propagation step where the  
 11 right hand side term is a collision operator, derived from a first order linearization.  
 12 From the main equation of the lattice BGK model it is possible to derive the Navier-  
 13 Stokes equation using the Chapman-Enskog expansion if the local equilibrium function  
 14 is chosen as follows (Chen and Doolen, 1998; Chopard and Droz, 1998, pp. 71-74;  
 15 Wolf-Gladrow, 2000, pp. 145-152; Rothman and Zaleski, 2004, pp. 78-79, Succi, 2001,  
 16 pp. 66-68):

$$17 \quad f_i^{\text{eq}} = \rho t_p \left[ 1 + \frac{c_{i\beta} u_\beta}{c_s^2} + \frac{1}{2} \left( \frac{c_{i\beta} u_\beta}{c_s^2} \right)^2 - \frac{u_\beta u_\beta}{2c_s^2} \right] \quad (2)$$

18 where the macroscopic variables density and velocity are represented by  $\rho$  and  $\mathbf{u}$ ,  
 19 respectively. The Einstein's summation convention has been adopted for index  
 20  $\beta \in [1, 2]$  that denotes the components of velocities  $\mathbf{c}_i$  and  $\mathbf{u}$  in the  $\beta$  dimension. The  $t_p$   
 21 are weighting factors in Eq. (2). In a two-dimensional squared vicinity model,  
 22  $p \in [0, 2]$ ,  $t_0$  weights the particles at rest,  $t_1$  weights those particles that move  
 23 horizontally and vertically and  $t_2$  weights those particles that move diagonally. The

1 parameter known as lattice sound speed,  $c_s$ , is selected according to the vicinity model  
 2 chosen. Fig. 2 shows the two-dimensional  $d2q9$  model used in this work while the  
 3 values of the Eq. (2) parameters (Succi, 2001, pp. 69) are listed in Table 1.

4 The macroscopic density  $\rho(\mathbf{r}, t)$  and velocity vector  $\mathbf{u}(\mathbf{r}, t)$  are deduced from  $f_i$  in the  
 5 following way for  $q$  particles:

$$\begin{aligned}
 \rho(\mathbf{r}, t) &= \sum_{i=1}^q f_i(\mathbf{r}, t) \\
 \mathbf{u}(\mathbf{r}, t) &= \frac{\sum_{i=1}^q f_i(\mathbf{r}, t) \mathbf{c}_i}{\rho(\mathbf{r}, t)}
 \end{aligned}
 \tag{3}$$

7 The relaxation time parameter determines the kinematic viscosity,  $\nu$

$$\nu = c_s^2 \left( \tau - \frac{1}{2} \right)
 \tag{4}$$

9 and, therefore, the Reynolds number  $Re$

$$Re = \frac{U L}{\nu}
 \tag{5}$$

11 where,  $U$  and  $L$  denote the mean velocity and the typical length of the flow,  
 12 respectively.

13 The Reynolds number is high for natural airflows in greenhouses due to the low  
 14 value of the air kinematic viscosity. For this reason, the relaxation time is only a little  
 15 greater than 0.5. This fact can lead to lattice BGK model numerical instability. This  
 16 limitation can be removed as Hou et al. (1996) suggested by applying the large eddy  
 17 simulation (LES) method for simulating eddies larger than the lattice resolution, with a  
 18 subsequent extrapolation to lower resolution scales. In this work, the LES method is  
 19 applied using the Smagorinsky eddy viscosity model (Smagorinsky, 1963; Pope, 2000,  
 20 pp. 587-601; Succi, 2001, pp. 137-141). That model is characterized by the external  
 21 parameter  $C_{smago}$  called the Smagorinsky constant, and it can be easily coupled to the

1 lattice BGK model previously described. A detailed explanation of this combination can  
2 be found in Jiménez-Hornero et al. (2005).

3 To complete the description of the lattice models, Table 2 shows the conversion rules  
4 between the magnitudes used at a mesoscopic work scale and their corresponding  
5 macroscopic values (Succi, 2001, pp. 261-262). The scale factors  $\Delta r$  and  $\Delta t$  are,  
6 respectively, the length of the lattice spacing and the time elapsed during one iteration  
7 or time-step.

8

### 9 **3. PC cluster description**

10

11 The proposed homogeneous PC cluster consists of ten nodes, one master and nine  
12 slaves; each one has a microprocessor Intel Pentium IV 3.06 GHz and 1 Gb DDR400  
13 RAM. Only the master has a hard disk (120 Gb), in which the operating system is  
14 installed and all the nodes write the output files. As shown in Fig. 3, the nodes are  
15 connected through two local networks with the aim of separating the computing  
16 operations from the management traffic, reducing in this way the execution time. Both  
17 networks are star-connected with the master node (a) acting as the main console  
18 equipped with screen (e), keyboard (f) and mouse (g).

19 The first, or low velocity, network (100 Mbps) represented by thin lines in Fig. 3,  
20 consists of a switch (d), network cards in each node and the corresponding Cat. 5 UTP  
21 network wires. The main tasks of this network are the following: 1) Slave nodes (b)  
22 boot. Boot-up is done remotely from the master, as it is necessary for the 100 Mbps  
23 network cards to be compatible with the pre-boot execution environment (PXE). In this  
24 way, the operating system is downloaded and executed by the slave nodes without a  
25 hard disk. 2) Management of the communications between the master and the slave

1 nodes. Communications are mainly generated by the network file system (NFS).  
2 Although there are file systems specifically designed for PC clusters (e.g. Wang and  
3 Zhiwei, 2002), the NFS was considered to be efficient enough for the purpose of this  
4 work. 3) Management of other automatic tasks.

5 The second, or fast velocity, network (1 Gbps) represented by thick lines in Fig. 3,  
6 consists of the corresponding switch (c), network cards and Cat. 5 UTP network wires  
7 and it is devoted to managing the computational traffic between nodes as a consequence  
8 of a parallel execution of several tasks.

9 The operating system used is Linux because of its stability. Linux distributions that  
10 include tools for controlling a parallel execution are widely used in PC clusters (e.g.  
11 Campbell, 2001, pp. 119-184; Bookman, 2002). As programming language, FORTRAN  
12 90 is selected due to its ease in the management of large arrays of typical lattice model  
13 data. In combination with the message passing interface library MPICH 1.2.6  
14 (<http://www-unix.mcs.anl.gov/mpi>), FORTRAN 90 permits code writing for parallel  
15 computing in a simple way using the single program multiple data (SPMD) approach. In  
16 SPMD programs, the code executed by different nodes is identical except for a few  
17 processes that are done by the master (e.g. the synchronizations of computing and  
18 output files writing managed by the MPI library kernel). The source code compilation  
19 and linking are done in the master including the corresponding MPI functions. The  
20 executable program obtained is launched in the same node automatically creating an  
21 identical process for each slave node.



#### 1 **4. The lattice BGK model parallel implementation**

2

3 According to Grama et al. (2003, pp. 86), dividing a computation into smaller  
4 computations and assigning them to different processors for parallel execution are the  
5 two main factors in the parallel implementation of an algorithm. For those authors, the  
6 decomposition process consisted of dividing a computation into smaller parts, some or  
7 all of which may potentially be executed in parallel. Grama et al. (2003, pp. 86) also  
8 defined *tasks* as units of computation into which the main computation is subdivided by  
9 means of decomposition. Simultaneous execution of multiple tasks reduces the  
10 execution time.

11 The decomposition method used in this work is the partitioning of the problem  
12 domain. This approach is appropriate for deriving concurrency in algorithms operating  
13 on dense matrixes (i.e. the lattice BGK model). A task is created for each domain  
14 partition and it performs the computation using the corresponding local data. After all  
15 the tasks are finished, an extra computation is needed for combining their partial results  
16 to yield the final solution. To complete the lattice model parallel implementation it is  
17 necessary to take these tasks and assign or map them onto the available processors.  
18 Following Grama et al. (2003, pp. 110-115), the selection of the mapping scheme  
19 should be done attending to the following characteristics of the tasks: 1) Task  
20 generation. The partitioning of the domain leads to static task generation, in which all  
21 the tasks are known before the algorithm execution; 2) Task sizes. In the case of lattice  
22 model parallel implementation, all the tasks are uniform, requiring the same amount of  
23 execution time; 3) Size of data associated with tasks. One of the most important factors  
24 is that the data associated with a task must be available to the processor performing that

1 task. In order to avoid excessive data-movement overheads the mapping technique has  
2 to be chosen by taking into account the size and location of these data.

3 In parallel computing, the tasks need to interact with each other in order to share data  
4 or to synchronize information. The knowledge of the inter-task interactions makes them  
5 more suitable for certain mapping schemes. For the lattice model parallel  
6 implementation developed in this work the task interactions are: 1) static, because for  
7 each task they happen at known times prior to its execution; 2) regular, as they show a  
8 spatial structure that can be exploited for an efficient implementation; 3) read-write due  
9 to multiple tasks requiring to read and write on the shared data; 4) two-way, because the  
10 data or work needed by a task is supplied by another task.

11 Fig. 4 shows the mapping scheme used for the two-dimensional lattice BGK model  
12 parallel implementation proposed in this work. The mapping scheme is a kind of static  
13 mapping in which the tasks are distributed among processors before their execution.  
14 This block distribution shares the largest array dimension ( $nx > ny$ ) and assigns  
15 contiguous portions of the matrix to each processor (P0, P1 and P2) in such a way that it  
16 can execute computations on its data. In this work, the width of each array portion is the  
17 same and equal to  $nx$  divided by the number of processors. However, it is possible to  
18 deal with different matrix portions according to the different features of the processors  
19 when the PC cluster is heterogeneous. This mapping technique is suitable when the  
20 value of a site is updated depending on its neighbours' values. However,  
21 communications are needed if neighbouring data are placed on a contiguous processor.  
22 This problem is solved using the buffer columns denoted by 0 and  $mx + 1$  in Fig. 4.  
23 They keep a copy of the first, 1, and last,  $mx$ , columns of the right and left neighbouring  
24 processors, respectively. They are used for updating the outermost columns of the  
25 corresponding portion of array without bothering whether the values of the columns 1 or

1  $mx$  belonging to the neighbouring processors have been modified yet. The message-  
2 passing programming enables the described communications to be carried out.

3 The lattice BGK model computing step scheme for the processor P1 shown in Fig. 4  
4 is detailed in Fig. 5. As can be seen, non-blocking communications between processors  
5 are used for passing the buffer columns data. According to Gropp et al. (1999, pp. 93-  
6 96), the non-blocking operations permit the use of the processors during the  
7 communication overhead to obtain performance optimisation. However, some  
8 precautions are necessary when implementing the non-blocking scheme because errors  
9 can result from unsafe access to the data being communicated (Grama et al., 2003, pp.  
10 239-240).

11 The described lattice BGK model message-passing program is loosely asynchronous  
12 because the tasks only synchronize to perform interactions. During the rest of the  
13 execution time, they are completely asynchronous.

14

## 15 **5. Results**

16

17 Benefits, mainly in the form of a faster problem solution, accrue by using parallel  
18 computing as compared to sequential implementation for executing the lattice BGK  
19 model simulations. To examine these benefits, the concept of speedup,  $Sp$ , was used.  
20 Following Grama et al. (2003, pp. 198-200), speedup is defined as the ratio of the time  
21 taken to solve a problem on a single processor to the time required to solve the same  
22 problem on a PC cluster of  $k$  identical processors.

23 To find the corresponding speedup, simulations with one (sequential execution), two,  
24 four, six, eight and ten nodes, were done for determining the steady state natural airflow  
25 in the vertical cross-section of a tropical crop protection structure described by Montero

1 et al. (2001). As Jiménez-Hornero et al. (2005) show, the lattice model simulates the  
2 flow patterns in reasonable agreement with the experimental data reported by Montero  
3 et al. (2001) using a 1:15 scale model replica of this kind of greenhouse immersed in a  
4 flume tank filled with water for four ventilator configurations. Fig. 6 shows the  
5 ventilator configuration considered in this work.

6 The lattice model simulations were done on a rectangular domain with  $n_x$  of 400 and  
7  $n_y$  of 150 lattice units setting  $\Delta r$  and  $\Delta t$  to  $2.666 \times 10^{-3}$  m and  $2.222 \times 10^{-3}$  s, respectively.  
8 The horizontal mean velocity not influenced by the greenhouse structure (not reported  
9 by Montero et al., 2001) was fitted to  $0.066 \text{ m s}^{-1}$  while the relaxation time parameter,  $\tau$   
10 of 0.5009378, corresponded to the water kinematic viscosity,  $10^{-6} \text{ m}^2 \text{ s}^{-1}$ . The  
11 Smagorinsky constant,  $C_{smago}$ , was set to 0.15. The periodic boundary condition was  
12 applied in the flow direction while the free slip boundary condition was used in the  
13 upper margin of the domain. The steady state was considered to be reached when the  
14 flow velocity difference in its mean value was kept less than or equal to  $1 \times 10^{-6} \text{ m s}^{-1}$   
15 between two consecutive time-steps in all the nodes of a sampling area of 200 and 100  
16 lattice units in width and height, respectively, centered on the greenhouse cross-section.  
17 This happened for a maximum time-steps number of  $205 \times 10^3$ , taking into account that a  
18 warm up stage of 1500 time-steps was firstly run with  $\tau = 1$  to avoid the blowing up  
19 risk.

20 The greenhouse structure, roof and sidewalls, as well as the ventilator configuration,  
21 were introduced as data in the lattice BGK model using the marker and cell technique  
22 by means of a computer aid design (CAD) tool. In the lattice nodes marked as obstacles  
23 (roof and sidewalls of the greenhouse and ground), the bounce-back rule was applied to  
24 simulate the interaction between the fluid particles and the solid sites, obtaining a zero  
25 fluid velocity.

1 Fig. 7 shows the steady state flow pattern simulated with the lattice model. A jet flow  
 2 is created at the height of the sidewall openings that is deflected by the leeward wall,  
 3 creating a counter clockwise circulation in the center of the greenhouse. The maximum  
 4 jet velocity,  $0.0985 \text{ m s}^{-1}$ , is reached near the windward opening inside the crop  
 5 protection structure while at the near roof, the velocity is  $0.0552 \text{ m s}^{-1}$ . Taking as a  
 6 reference the velocity at the greenhouse eaves,  $0.1193 \text{ m s}^{-1}$ , the calculated ratio  
 7 between the mean velocity at the sidewall opening,  $0.0662 \text{ m s}^{-1}$ , and the former is  
 8 0.555, similar to that estimated by Montero et al. (2001), 0.55.

9 Table 3 shows the execution time reduction and, consequently, a high speedup as the  
 10 number of processors increases. As can be checked, the speedup is not equal to  $k$  in any  
 11 case. This non-ideal behaviour is due to the time devoted by the processors to  
 12 communicating between with each other instead of computing. Efficiency,  $E$ , is a  
 13 measure of the fraction of time for which processors are usefully employed. According  
 14 to Grama et al. (2003, pp. 202-203) it is defined as the ratio of speedup  $Sp$  to the  
 15 number of processors,  $k$ .

$$16 \quad E = \frac{Sp}{k} \quad (6)$$

17 It can be noted in Table 3 that efficiency diminishes as  $k$  increases due to a higher  
 18 communications overhead.

19

## 20 **6. Conclusions**

21

22 The execution time reduction and performance speedup that can be obtained with the  
 23 proposed Linux PC cluster make parallel computing an useful tool for simulating fluid  
 24 flow patterns with lattice models. Thus, the modeller can adopt appropriate decisions  
 25 when fitting parameters to obtain the correct phenomena description in less time.

1 The two-dimensional analysis of the natural airflows in a tropical greenhouse  
2 performed in this work was a relatively simple problem in which the domain  
3 decomposition in strips could be used without having excessive communications  
4 overhead. The same method can be used to simulate other similar two-dimensional  
5 phenomena with a lattice model, such as the wind velocity reduction due to the effect of  
6 different kinds of windbreaks. However, for more complex problems in two and three  
7 dimensions, a detailed study for selecting the suitable decomposition technique is  
8 required to avoid low performance efficiencies. This situation is frequent when  
9 simulating the dispersion of pollutants in soils or flows in which the thermal and  
10 turbulent diffusion are involved in the presence of intricate configurations of obstacles.  
11 The time and effort expended on the pre-modelling study is the main drawback that the  
12 user has to overcome when parallel implementation is applied to the lattice model with  
13 the aim of being run on a Linux PC cluster.

14

## 15 **Acknowledgements**

16

17 The authors gratefully acknowledge the support of the DGIFAP of the Junta de  
18 Andalucía through project CAO01-001-C4-01. F.J. Jiménez-Hornero wishes to thank  
19 the support of Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía  
20 (Ayudas para facilitar el Retorno de Investigadores a Centros de Investigación y  
21 Universidades de Andalucía).

## 1 **References**

- 2 Bartzanas, T., Boulard, T., Kittas, C., 2002. Numerical simulation of the airflow and  
3 temperature distribution in a tunnel greenhouse equipped with insect-proof screen in  
4 the openings. *Comput. Electron. Agric.* 34, 207-221.
- 5 Bathnagar, P., Gross, E.P., Krook, M.K., 1954. A model for collision processes in  
6 gases. I. Small amplitude processes in charged and neutral one-component systems.  
7 *Phys. Rev. B* 94, 511-525.
- 8 Bernsdorf, J., 2001. Introduction to lattice gas and lattice Boltzmann methods. DFG and  
9 KONWIHR-Workshop. Lattice Boltzmann methods: theory and applications in fluid  
10 mechanics. Institute of Fluid Mechanics, University of Erlangen-Nuremberg,  
11 Nuremberg, Germany.
- 12 Bookman, C., 2002. *Linux Clustering: Building and Maintaining Linux Clusters*. New  
13 Riders, Indianapolis, IN, USA.
- 14 Campbell, I., 2001. *Reliable Linux: Assuring High Availability*. John Wiley & Sons,  
15 New York, NY, USA.
- 16 Chen, S., Doolen, G., 1998. Lattice Boltzmann method for fluid flows. *Annu. Rev.*  
17 *Fluid Mech.* 30, 329-364.
- 18 Chen, S., Wang, Z., Shan, X., Doolen, G., 1992. Lattice Boltzmann computational fluid  
19 dynamics in three dimensions. *J. Stat. Phys.* 68, 379-400.
- 20 Chopard, B., Droz, M., 1998. *Cellular Automata Modeling of Physical Systems*.  
21 Cambridge University Press, Cambridge, UK.
- 22 Chopard, B., Hoekstra, A., 2005. Computational science of lattice Boltzmann  
23 modelling. *Futur. Gener. Comp. Syst.* 20, 907-908.

- 1 Dongarra, J., Kennedy, K., White, A., 2003. Introduction. In: Dongarra, J., Foster, I.,  
2 Fox, G., Kennedy, K., White, A., Torczon, L., Gropp, W. (Eds.), *The Sourcebook of*  
3 *Parallel Computing*. Morgan Kaufmann, San Francisco, CA, USA, pp. 3-14.
- 4 Dongarra, J., Sterling, T., Simon, H., Strohmaier, E., 2005. High-Performance  
5 Computing: Clusters, Constellations, MPPs, and Future Directions. *Comput. Sci.*  
6 *Eng.* 7, 51-59.
- 7 Grama, A., Karypis, G., Kumar, V., Gupta, A., 2003. *An Introduction to Parallel*  
8 *Computing: Design and Analysis of Algorithms* (2nd ed.). Addison-Wesley, Harlow,  
9 UK.
- 10 Gropp, W., Huss-Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., Snir,  
11 M., 1998. *MPI: the Complete Reference*. Vol. 2. *The MPI Extensions*. MIT Press,  
12 Cambridge, MA, USA.
- 13 Gropp, W., Lusk, E., Skjellum, A., 1999. *Using MPI: Portable Parallel Programming*  
14 *with the Message-Passing Interface* (2nd ed.). MIT Press, Cambridge, MA, USA.
- 15 Henderson, R., Meiron, D., Parashar, M., Samtaney, R., 2003. Parallel computing in  
16 computational fluid dynamics. In: Dongarra, J., Foster, I., Fox, G., Kennedy, K.,  
17 White, A., Torczon, L., Gropp, W. (Eds.), *The Sourcebook of Parallel Computing*.  
18 Morgan Kaufmann, San Francisco, CA, USA, pp. 93-144.
- 19 Hou, S., Sterling, J., Chen, S., Doolen, G.D., 1996. A lattice subgrid model for high  
20 Reynolds number flows. *Fields Institute Communications* 6, 151-166.
- 21 Jiménez-Hornero, F.J., Gutierrez de Ravé, E., Hidalgo, R., Giráldez, J.V., 2005.  
22 Numerical study of the natural airflow in greenhouses using a two-dimensional  
23 lattice model. *Biosystems Eng.* 91, 219-228.



- 1 Karniadakis, G.E., Kirby, R.M., 2003. *Parallel Scientific Computing in C++ and MPI:*  
2 *A seamless Approach to Parallel Algorithms and their Implementation.* Cambridge  
3 *University Press, Cambridge, UK.*
- 4 Mistriotis, A., Bot, G.P.A., Picuno, P., Scarascia-Mugnozza, G. 1997. Analysis of the  
5 efficiency of greenhouse ventilation using computational fluid dynamics. *Agric. For.*  
6 *Meteorol.* 85, 217-228.
- 7 Mistriotis, A., Briassoulis, D., 2002. Numerical estimation of the internal and external  
8 aerodynamic coefficients of a tunnel greenhouse structure with openings. *Comput.*  
9 *Electron. Agric.* 34, 191-205.
- 10 Montero, J.I., Hunt, G.R., Kamaruddin, R., Antón, A., Bailey, B.J., 2001. Effect of  
11 ventilator configuration on wind-driven ventilation in a crop protection structure for  
12 the tropics. *J. Agr. Eng. Res.* 80, 99-107.
- 13 Pope, S.B., 2000. *Turbulent Flows.* Cambridge University Press, Cambridge, UK.
- 14 Qian, Y.H., D’Humières, D., Lallemand, P., 1992. Lattice BGK models for Navier-  
15 Stokes equation. *Europhys. Lett.* 17, 479-484.
- 16 Rothman, D.H., Zaleski, S., 2004. *Lattice-gas Cellular Automata. Simple Models of*  
17 *Complex Hydrodynamics (1st paperback ed.).* Cambridge University Press,  
18 *Cambridge, UK.*
- 19 Smagorinsky, J., 1963. General circulation experiments with the primitive equations: I.  
20 the basic equations. *Mon. Weather Rev.*, 91, 99-164.
- 21 Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J., 1998. *MPI: the*  
22 *Complete Reference. Vol. 1. The MPI Core (2nd ed.).* MIT Press, Cambridge, MA,  
23 *USA.*

- 1 Succi, S., 2001. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.*
- 2 *Numerical Mathematics and Scientific Computation.* Oxford University Press,
- 3 Oxford, UK.
- 4 Wang, J., Zhiwei, X., 2002. Cluster file systems: a case study. *Futur. Gener. Comp.*
- 5 *Syst.* 18, 373-387.
- 6 Wolf-Gladrow, D.A., 2000. *Lattice-gas Automata and Lattice Boltzmann Models.*
- 7 *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, Germany.
- 8 Xia, B., Sun, D.-W., 2002. Applications of computational fluid dynamics (CFD) in the
- 9 food industry: a review. *Comput. Electron. Agric.* 34, 5-24.

## 1 **Figure captions**

2

3 Fig. 1. Lattice model stages completed in each time-step according to Bernsdorf (2001).

4 In the propagation phase, the particles move from a lattice site to its neighbour. When  
5 two or more particles move to the same site, they collide according to the conservation  
6 of particles masses and momentum.

7

8 Fig. 2. The vicinity model  $d2q9$  used in simulations done with the lattice model;  $d$  and  $q$   
9 mean the number of neighbours considered, respectively. The variable  $f_i$  represents the  
10 probability of finding a particle that is moving in the direction of link  $i$ .

11

12 Fig. 3. Diagram of the proposed Linux PC cluster. The nodes are star-connected by  
13 means of 100 Mbps and 1 Gbps networks represented by thin and thick lines,  
14 respectively. The meanings of the reference numbers are the following: (a) Master node;  
15 (b) Slave nodes; (c) 100 Mbps network switch; (d) 1 Gbps network switch; (e) Master  
16 node screen; (f) Master node keyboard; (g) Master node mouse.

17

18 Fig. 4. Mapping scheme used for the lattice model parallel implementation. Each  
19 processor (P0, P1 and P2) has an array of  $n_y \times m_x$ . The communications between  
20 processors are done using the buffer columns denoted by 0 and  $m_x + 1$ .

21

22 Fig. 5. Parallel computing step scheme for the processor P1 shown in Fig.4.

23

24 Fig. 6. Cross-section of the crop protection structure model used in the experiments  
25 reported by Montero et al. (2001); the dimensions are in m.

1

2 Fig. 7. Cross-section flow obtained with the lattice model for the crop protection

3 structure model reported by Montero et al. (2001) with a side and roof ventilator

4 configuration.

1 **Tables**

2

3

4

5

6

7

Table 1

8

Parameters for the  $d2q9$  vicinity model

Weighting factors			Lattice sound speed
$t_0$	$t_1$	$t_2$	$c_s$
4/9	1/9	1/36	$1/\sqrt{3}$

9

10

11

1

2

3

4

5

6

7

8

Table 2

9

Conversion rules between the lattice model magnitudes and their physical

10

values

Magnitude	Lattice model	Physical value
Time	$t$	$t^{\text{real}} = \Delta t t$
Space	$r$	$r^{\text{real}} = \Delta r r$
Velocity	$u$	$u^{\text{real}} = (\Delta r / \Delta t) u$
Kinematic viscosity	$\nu$	$\nu^{\text{real}} = (\Delta r^2 / \Delta t) \nu$

11

1

2

3

4

5

6

7 Table 3

8 Performance metrics for the lattice model parallel computing done with different

9 number of processors,  $k$ .  $Sp$  and  $E$  stand for speedup and efficiency, respectively.

Number of processors ( $k$ )	Execution time (s)	$Sp$	$E$
1	9351	-	-
2	4724	1.979	0.985
4	2390	3.912	0.978
6	1607	5.819	0.969
8	1252	7.468	0.933
10	1042	8.974	0.897

10

11

1 **Figures**

2

3

4

Propagation

5

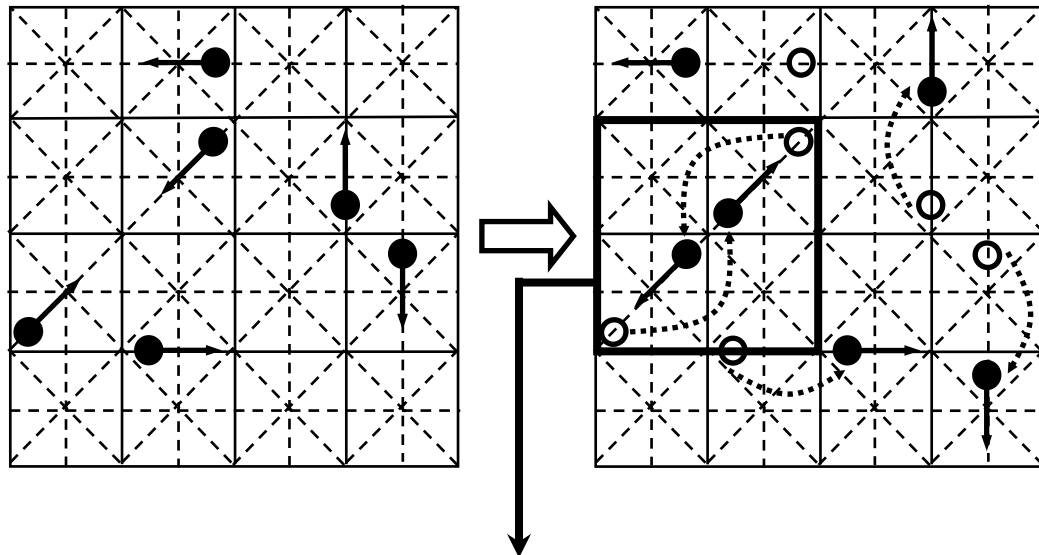
6

7

8

9

10



11

12

Collision

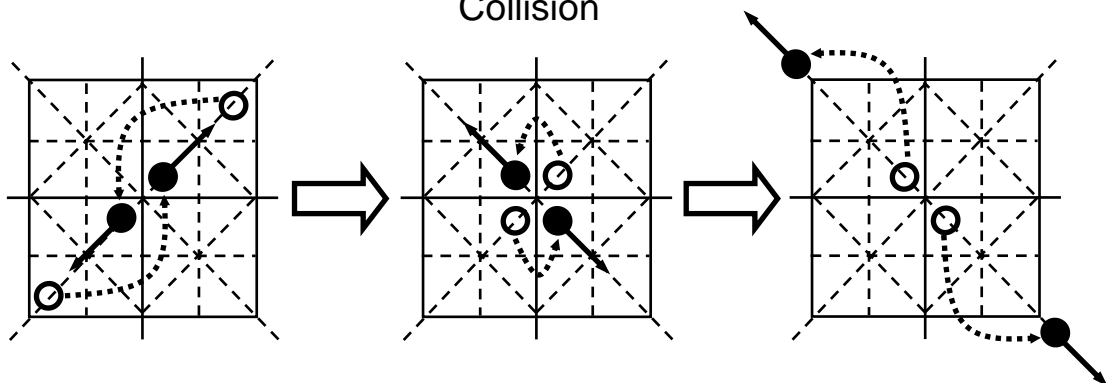
13

14

15

16

17



18

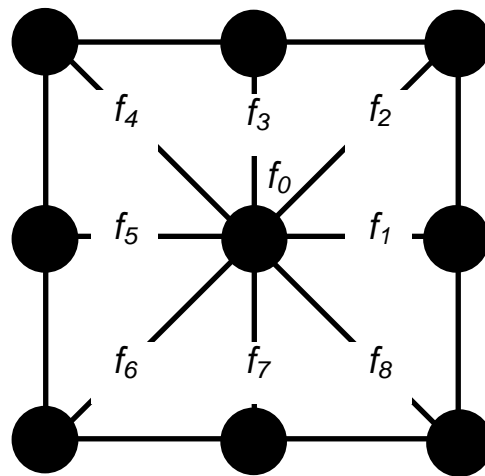
19 Fig. 1. Lattice model stages completed in each time-step according to Bernsdorf (2001).

20 In the propagation phase, the particles move from a lattice site to its neighbour.

21 When two or more particles move to the same site, they collide according to the conservation

22 of particles masses and momentum.





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

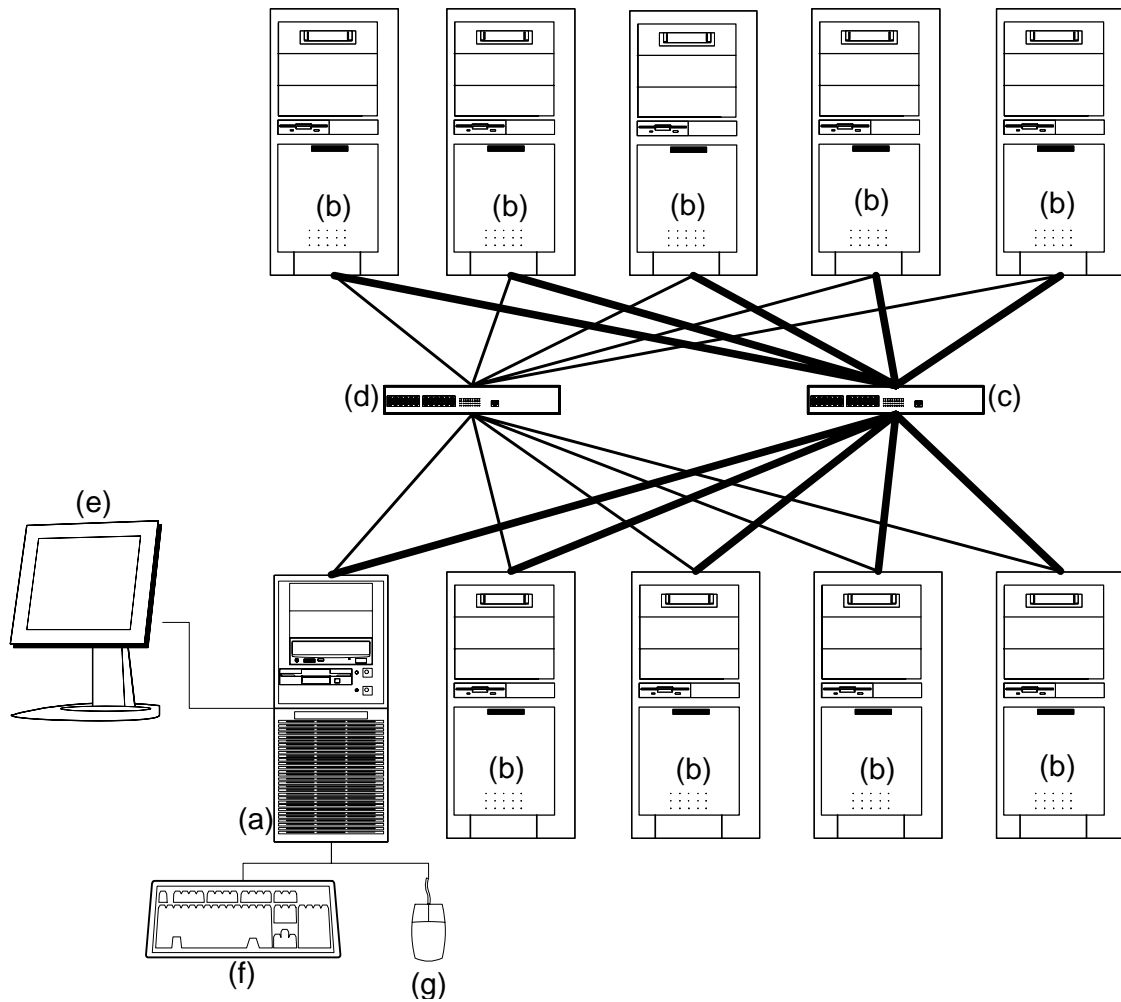
Fig. 2. The vicinity model  $d2q9$  used in simulations done with the lattice model;  $d$  and  $q$  mean the number of neighbours considered, respectively. The variable  $f_i$  represents the probability of finding a particle that is moving in the direction of link  $i$ .

1

2

3

4



5

6

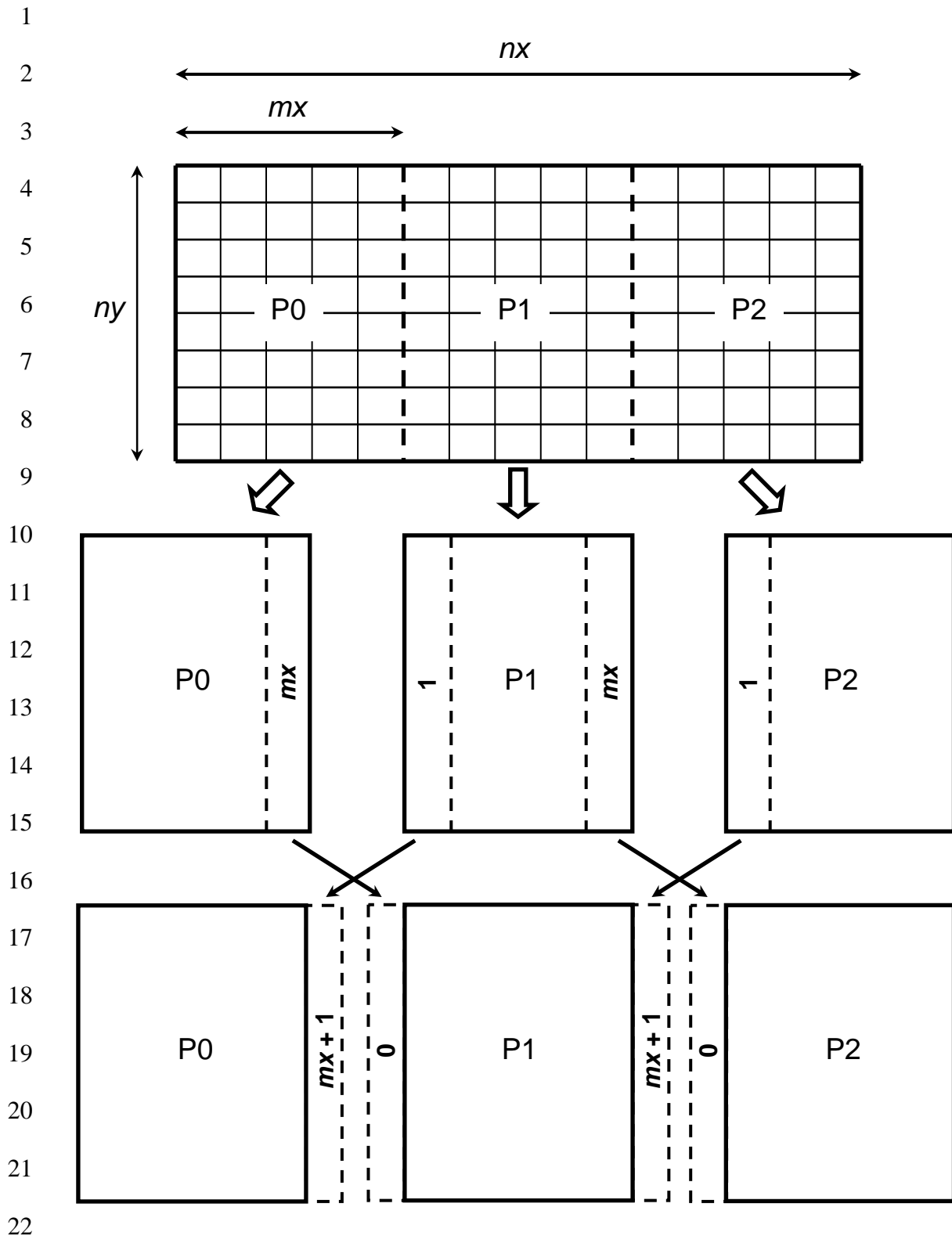
7 Fig. 3. Diagram of the proposed Linux PC cluster. The nodes are star-connected by

8 means of 100 Mbps and 1 Gbps networks represented by thin and thick lines,

9 respectively. The meanings of the reference numbers are the following: (a) Master node;

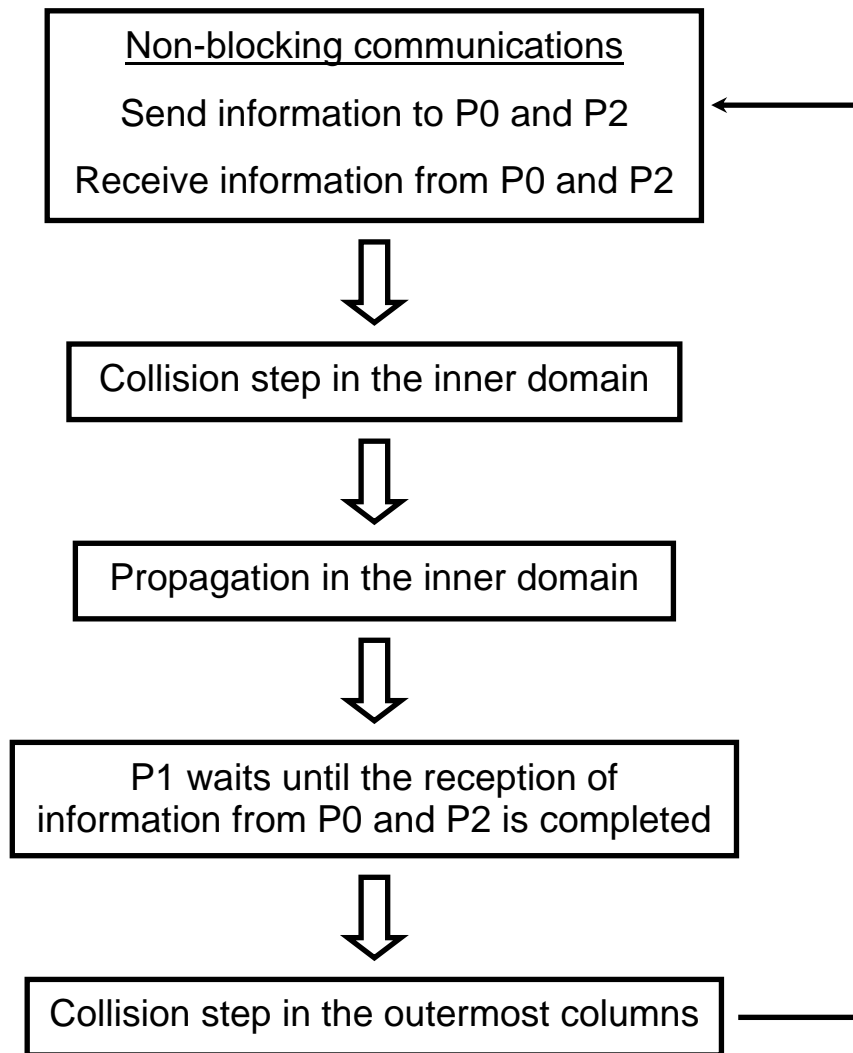
10 (b) Slave nodes; (c) 100 Mbps network switch; (d) 1 Gbps network switch; (e) Master

11 node screen; (f) Master node keyboard; (g) Master node mouse.



23 Fig. 4. Mapping scheme used for the lattice model parallel implementation. Each  
 24 processor ( $P_0$ ,  $P_1$  and  $P_2$ ) has an array of  $ny \times mx$ . The communications between  
 25 processors are done using the buffer columns denoted by  $0$  and  $mx + 1$ .

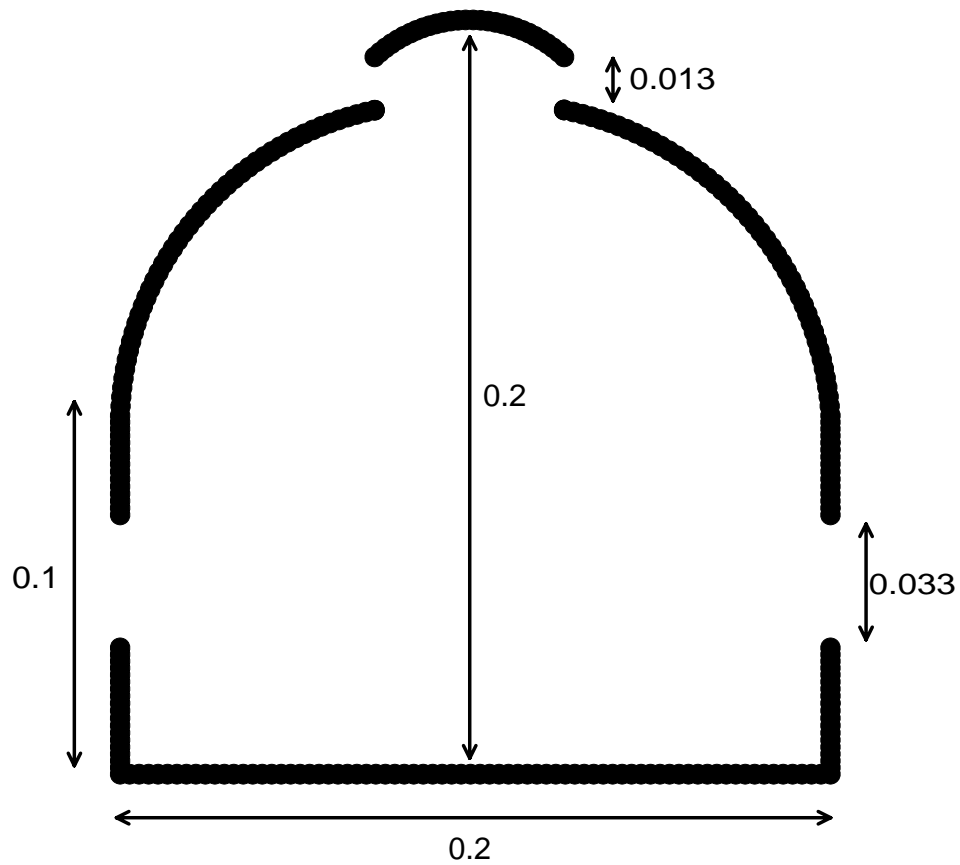
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19



20 Fig. 5. Parallel computing step scheme for the processor P1 shown in Fig.4.

21

1  
2  
3  
4  
5

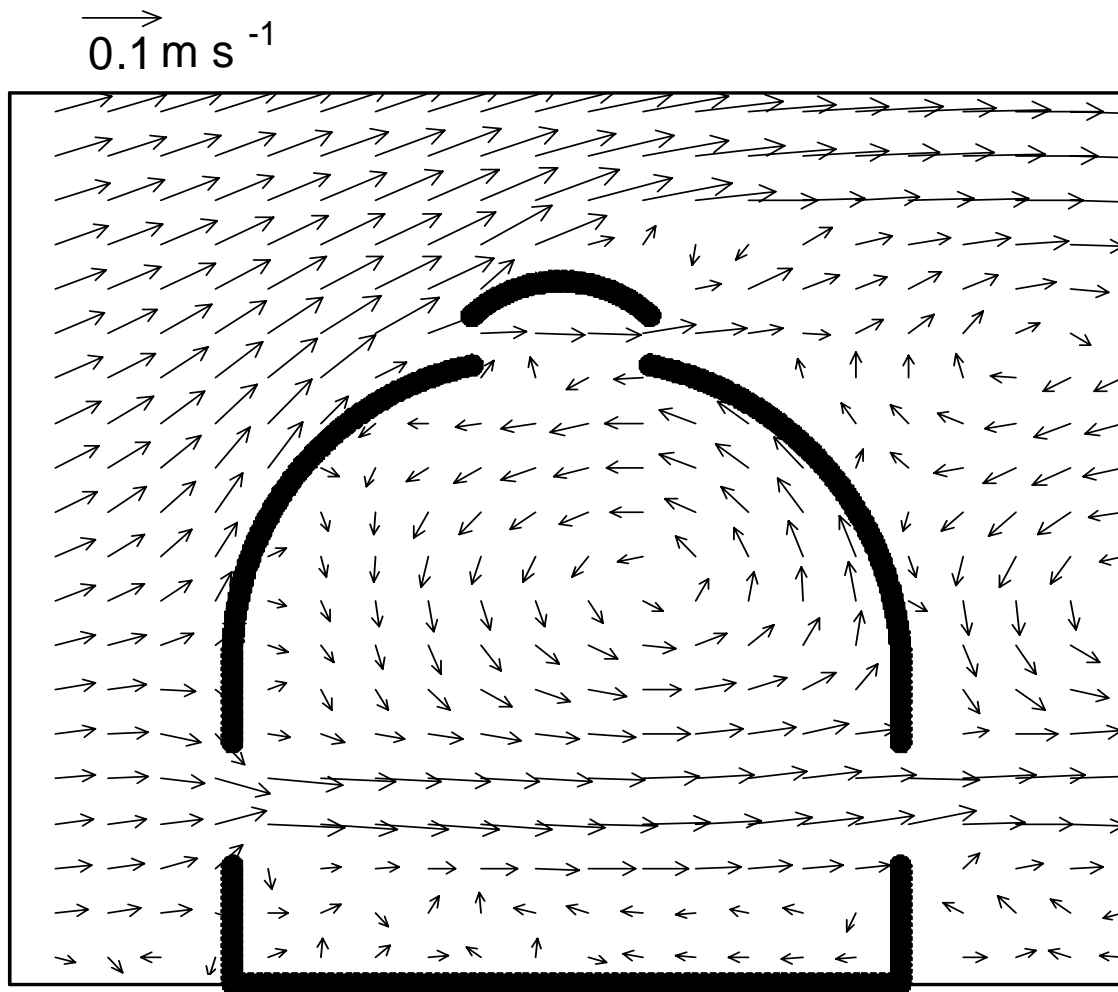


6  
7

8 Fig. 6. Cross-section of the crop protection structure model used in the experiments  
9 reported by Montero et al. (2001); the dimensions are in m.

1

2



3

4

5 Fig. 7. Cross-section flow obtained with the lattice model for the crop protection  
6 structure model reported by Montero et al. (2001) with a side and roof ventilator  
7 configuration.

8