PREPRINT

# A Distributed Computing Approach to the Simulation of Liquid Crystal Lattice Models

Cesare Chiccoli, Paolo Pasini, Franco Semeria
*INFN Sezione di Bologna, Via Irnerio 46, 40126 Bologna, Italy*
*Chiccoli@bo.infn.it - Pasini@bo.infn.it - Semeria@bo.infn.it*

Claudio Zannoni
*Dipartimento di Chimica Fisica e Inorganica and INSTM, Universitá, Viale Risorgimento 4, 40136 Bologna, Italy*
*Claudio.Zannoni@unibo.it*

Received Day Month Year
Revised Day Month Year

We have studied a lattice spin model of nematic liquid crystal-polymer composite films by means of extensive Monte Carlo simulations over a distributed computing network. The Condor processing system installed on the Italian Nuclear Physics Institute computer network was used. The use of several geometries and boundary conditions allowed us to investigate a wide number of different realistic or speculative models. Many of the simulations differ only by a small number of parameters and they can be effectively performed in parallel. The results of the simulations can be analyzed globally when all the computations are completed and then for example, employed to extrapolate Phase Diagrams or other complex physical quantities, They provide an effective example of wide area distributed computing applications which could be also implemented in future GRID approaches.

*Keywords*: Distributed Computing, Simulations, Monte Carlo, Liquid Crystals

PACS Nos.: 2.70.Uu, 07.05.Tp, 61.30.-v

## 1. Introduction

Liquid crystals (LC) are states of condensed matter intermediate between solids and liquids and consist of various phases with different molecular organizations [1]. The main characteristic of liquid crystals at molecular level is that they possess long range orientational order, together with a translational mobility that is similar to that of liquids in nematic phases and reduced in other layered or smectic, types. A theoretical investigation of LC can be undertaken, as for any other complex fluids, by means of approximate theories or by computer simulations [2]. In particular, Monte Carlo simulations, amongst the foremost techniques used in studying phase transitions and critical phenomena, have been widely applied to investigate

liquid crystals [3]. The first implemented simulation models were lattice ones [4-6] where the molecules, or tightly ordered cluster of molecules, are represented by three dimensional unit vectors ("spins") and considered to have a fixed position at the lattice sites. The spins possess full rotational freedom, subject to a certain potential, so that this restriction does not affect the long range orientational behavior that constitutes the essential feature of LC phases. These models are still very important in studying liquid crystal systems, although off-lattice potentials, like Gay-Berne [7,8] or even realistic atomistic ones [9] have been proposed. The main advantage in using lattice models, particular close to a transition, as is often the case for LC, lies in the very large number of particles which can be treated in comparison with off-lattice systems, and is further justified by the fact that local chemical details are not essential in studying phase transitions and their neighborhood. A detailed investigation of spin models still requires, however, a very significant amount of computing power which can require using parallel computing, typically when studying large systems, or suggests employing distributed resources for the case of smaller simulation samples where many different values of parameters, corresponding to different physical conditions, need to be studied. Here we wish to present a distributed approach for studying liquid crystal lattice models using the software CONDOR [10]. This software is a system of distributed computing tools which is similar to GRID but generally used at a lower scale. It has been developed at the Computer Science Department at the University of Wisconsin - Madison and is implemented on the network of the Italian National Institute for Nuclear Physics. The paper is organized as follows: first we briefly summarize the main features of the Monte Carlo simulation models; then we describe the CONDOR software and finally we discuss how we performed the simulations of the liquid crystal systems.

## 2. Lattice Models

Lattice spin models consist of systems of interacting centers ("spins") placed at the sites of a certain regular lattice and appear at first sight the very antithesis of liquid crystals, if only because they do not allow flow! Despite this, they have also been successfully employed in investigating the orientational properties of nematics since the pioneering work of Lebwohl and Lasher [4,5]. This prototype model is indeed the simplest one with the correct symmetry for nematic liquid crystals (in particular the potential is invariant for an head-tail flip of the molecules). The particles, assumed to have uniaxial symmetry and represented by three dimensional spins located at the sites of a $L \times L \times L$ cubic lattice, interact through a pair potential of the form:

$$U_{i,j} = -\epsilon_{i,j} P_2(\cos \beta_{i,j}). \tag{1}$$

where $\epsilon_{i,j}$ is a positive constant, $\epsilon$, for nearest neighbor spins $i$ and $j$ and zero otherwise, $P_2$ is the second Legendre polynomial and $\beta_{i,j}$ is the angle between the spins. This interaction tends to bring molecules parallel to one another and effectively models whatever underlying intermolecular interaction either attractive

or repulsive that does that. The LL model has been studied and generalized, also to non-uniaxial particles [6], by many authors for bulk and confined systems [3]. In particular confined systems require several kinds of Boundary Conditions depending on the model to be simulated. Here we wish to present a case as an example of the utilization of Condor.

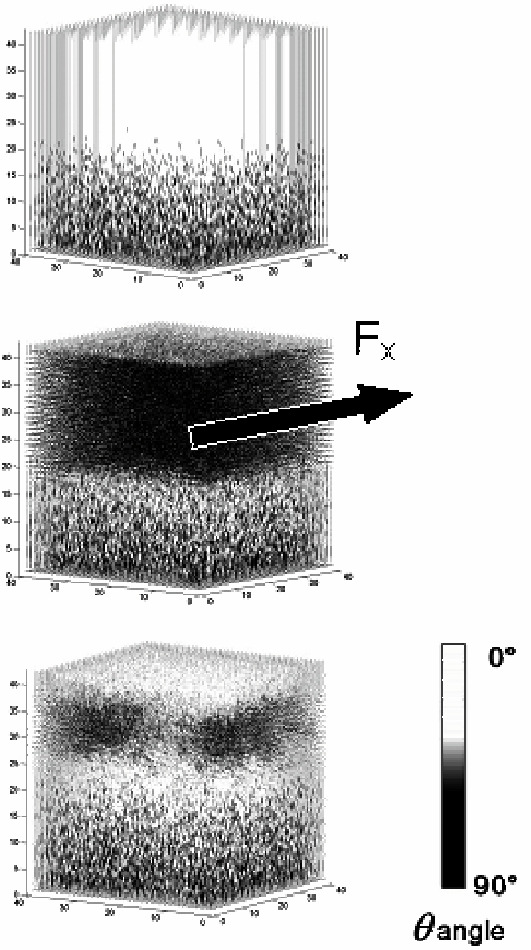


Fig. 1. An example of the molecular organization of the system as obtained from the Monte Carlo simulation. The alignment along z ($\theta = 0^o$) or perpendicular to it ($\theta = 90^o$) are coded with a gray scale from white and black respectively. (Top) starting configuration, (Middle) after the application of the external field along x (Fx); (Bottom) after the switching off of the field.

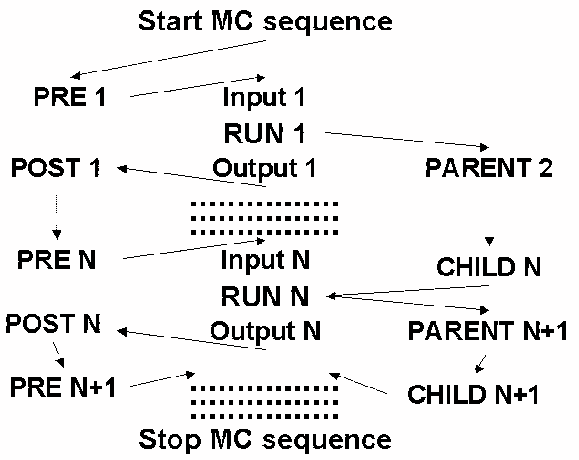


Fig. 2. Scheme of usage of the DAGman

## 3. The POLICase

The model we wish to consider sets out to mimic some of the essential features of a composite system consisting of a very thin film of nematic liquid crystal in contact with a layer of polymer. This system has attracted a lot of interest both from the basic and applied research point of view[11,12]. The polymer surface is assumed to be porous and soaked by the nematic, while the nematic layer floating above the polymer film can be aligned by a second surface or by a suitably applied external field. Here we adopt a spin lattice model as in eq. 1 to simulate both nematic and polymer particles. We assume, however, nematic molecules to be represented by mobile spins whose orientation is updated during the simulation and polymer chain units as frozen spins. Moreover we simulate the presence of the polymer chains protruding from the bottom of the sample and the consequent decreasing concentration of polymer units moving away from the bottom surface by assuming in the first few bottom layers of the system a decreasing percentage of frozen particles spins. Monte Carlo simulations are then employed to investigate the creation of structures in the nematic planar layers across the sample with different concentration gradient of frozen oriented particles inside the system. For example, the percentage of frozen randomly oriented particles can change from 50% to 5%, with a 5% variation every 2 layers, going from the bottom to the middle layer of the system. In our system we have arbitrarily chosen the boundary conditions at the top and bottom to consist of spins all aligned along the Z and X axis respectively. Of particular interest is the case where an external field is applied to the system and induces an alignment of the molecules along the field direction (the x axis).

The observables needed to describe the thermodynamic state of the systems are mainly the internal energy and the orientational order parameter. The energy

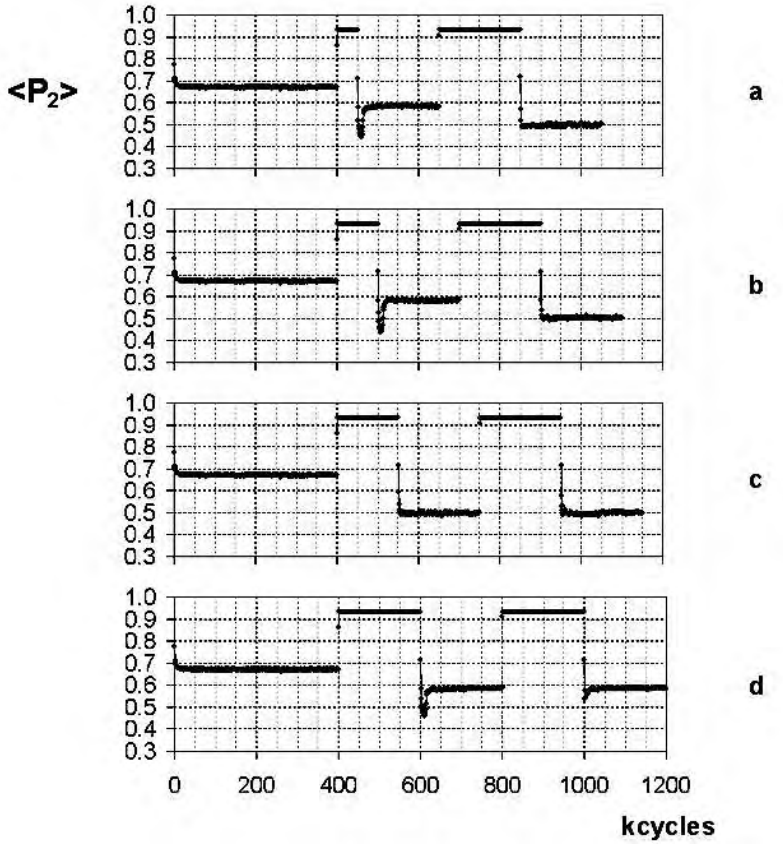


Fig. 3. Behaviour of the ¡P2¿ order parameter for a typical simulation. Apart from the first 400 *kcycles* the plates (a-d) represent curves with sequences of different cycle lengths when the external field is applied for the first time and successively switched OFF, ON and OFF respectively.

evaluation is an essential part of the Monte Carlo procedure and proceeds directly from the intermolecular potential in eq. 1. The second rank order parameter $\langle P_2 \rangle$, which is calculated from the diagonalisation of the ordering matrix [3], denotes the nematic ordering of the system and ranges from 1 (complete order) to 0 (complete disorder). Moreover some more qualitative but informative results, such as the plots of snapshots of the particles and the simulations of polarized microscopy images often used to characterize this type of systems [13] are obtained off-line. Also these images are analyzed for different field strengths and when these fields are switched ON and OFF, after several cycles of application while the field is applied (the ON state). The successive application of field pulses allows us to investigate if memory effects are induced in the system when the field is turned off. In Fig. 1 we report an example of the molecular organization induced by the boundary conditions and

the effect resulting from the application of an external field. A color coding has been employed to better appreciate the different orientations of the molecules. The main results for the physics are presented elsewhere [14] while the present paper is devoted to discussing the distributed computing used. Typically the simulation runs are performed on a $N = 40 \times 40 \times 42$ system containing 67200 spins.

Table 1. Example of a complete simulation computational effort needed for the Policase

| Computational STEPS | TOTAL RUNs |
|---|---|
| Equilibration effort for Each Interesting Temperature | 400 |
| Switching ON of 15 different Field strengths | 15 * 400 = 6000 |
| Switching OFF of 15 Fields * 4 different Application Time of field ON * 400 RUNs | 4 * 15 * 400 = 24000 |
| Analysis of Fields again ON and OFF after each application time, with an increasing request of runs... | [4*4] * 4 * 15 * 400 = 384000 |

The MC sequence is broken up in runs of 1000 cycles, where a cycle corresponds to a N attempted particle updates, according to the Metropolis procedure [2,3] and each cycle starts from the final state of the previous one. This segmentation has the advantage of always making available the latest produced data snapshot for further analysis; of spreading the computation over a very large number of hosts and, finally, of avoiding the use of the less performing hosts for too long a time. The first part of any new simulation consists of equilibration runs, typically at least 400, that are then discarded, before starting production runs and accumulating the data for the calculation of observable averages. These MC runs are synchronized with DAGMan (Directed Acyclic Graph Manager) [6], a meta-scheduler for Condor that manages dependencies between jobs at a higher level than the Condor Scheduler itself. This is realized in practice by saving and then using as input data the spins configuration of the system when restarting the computation again, possibly on a new host. Thus each run is a parent of the next one, in turn the child of the previous one. Also PRE and POST operations are performed in order to save partial data and to analyze the evolution of computing parameters. In Fig. 2 we report a scheme of the DAGMan synchronizing procedure.

The use of DAGman is then essential for the synchronization of the whole MC sequences once they are segmented in smaller subsequences for analysis purposes and for the splitting and distribution of the calculations over the wider range of computing nodes.

We should stress that the use of Condor is particularly useful for simulations of the present type, where a very large number of runs are needed to complete the study. For example, in the present case, the minimum requirements to complete each temperature analysis was about 384000 runs (1 Run = 1000 MC cycles) as

results from the scheme reported in Table 1.

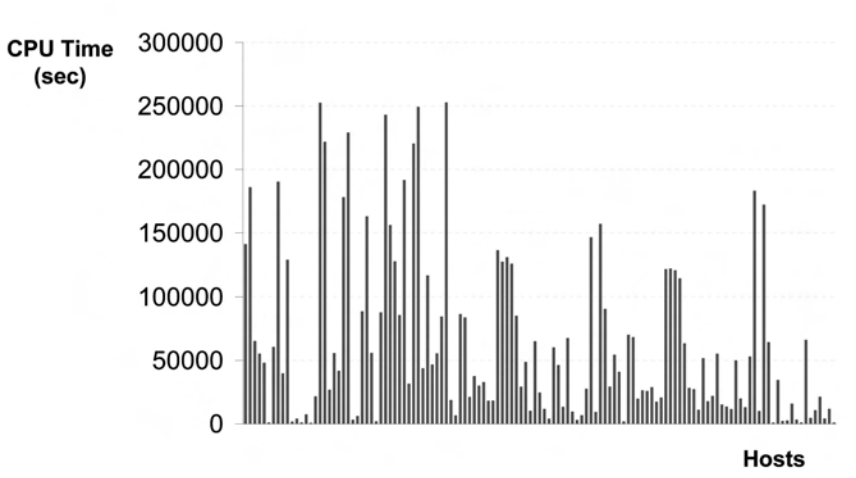


Fig. 4. Total CPU time (sec) per host.

Figure 3 shows the behaviour of the $\langle P_2 \rangle$ order parameter for a typical simulation sequence. In the first 400 kcycles of the simulation the system is equilibrated, as required for a reliable and reproducible computation of physical quantities of interest, such as order parameters and energy. After the system has relaxed to equilibrium, an electric field **F** is applied to the system for a certain number of cycles, with the aim of reorienting on average the particles along the field direction, thus obtaining a more ordered system as the increased value of the $\langle P_2 \rangle$ parameter shows. We analyze the influence of the switching OFF and ON of the applied field, starting a OFF-ON-OFF switching sequence after different timings from the first field application. This field sequence is applied to the system after 50 (a), 100 (b), 150 (c), and 200 (d) *kcycles* after the first switching ON of the field, with the simulation continuing for 200 *kcycles* in each of the OFF or ON state of the field. This procedure is employed for a chosen fraction of the frozen particles, their geometrical distribution and spin orientations inside the layers. In a full study, the same should be done for different combinations of percentages and boundary conditions.

The large number of jobs necessary for the investigation can only be performed spreading the calculation over a large number of hosts of the Condor pool. In Figs. 4-7 we report some statistics of the usage of about 130 computers used for the simulations. In Fig. 4 the total CPU usage for each of the 130 nodes is shown. As it can be seen the CPU time used is very different for each of the nodes because some of these nodes were free, and then accessible to us, for more time in comparison with some others. The total computing time on a node is of course due to different competing factors, first of all the number of accepted jobs and computing performances of the node.

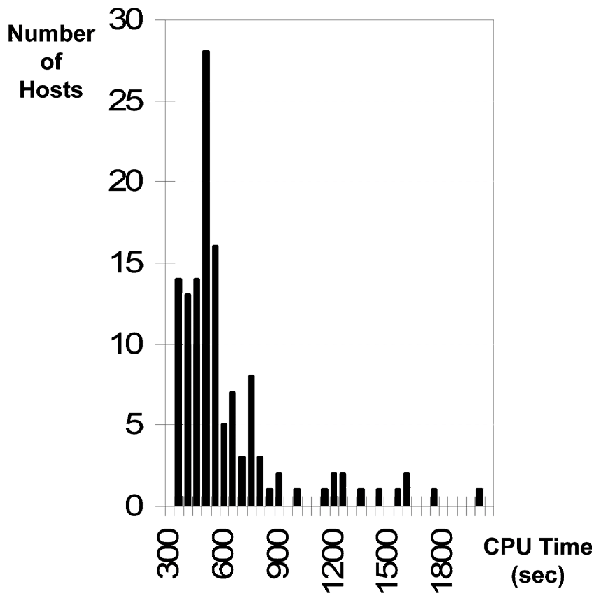


Fig. 5. Histogram of the distribution of the number of hosts on the average CPU time (sec) employed.

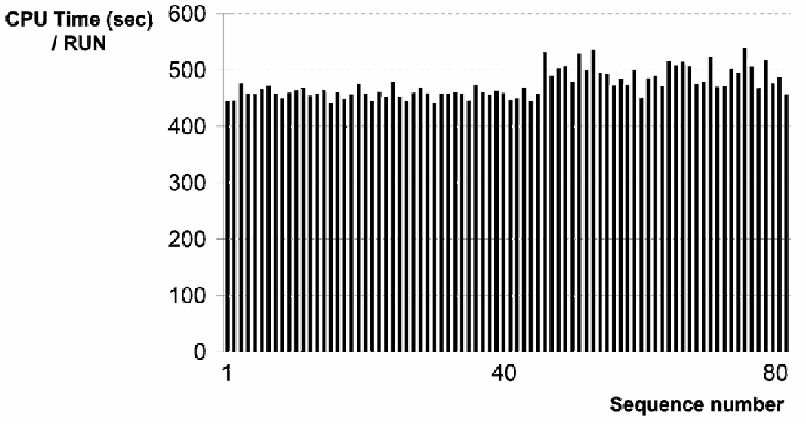


Fig. 6. Average CPU time (sec) / RUN per simulation sequence. The CPU time / RUN is about constant because the sequences are computed using the first available host of the pool. So differences in performance of the used HOSTS are spread all over the simulation sequences.

In our calculations we observed that the power of the hosts was comparable as can be seen in Fig. 5 where the average CPU time per run is shown. Only some of the 130 hosts were slower with respect to the others, but the assignment for

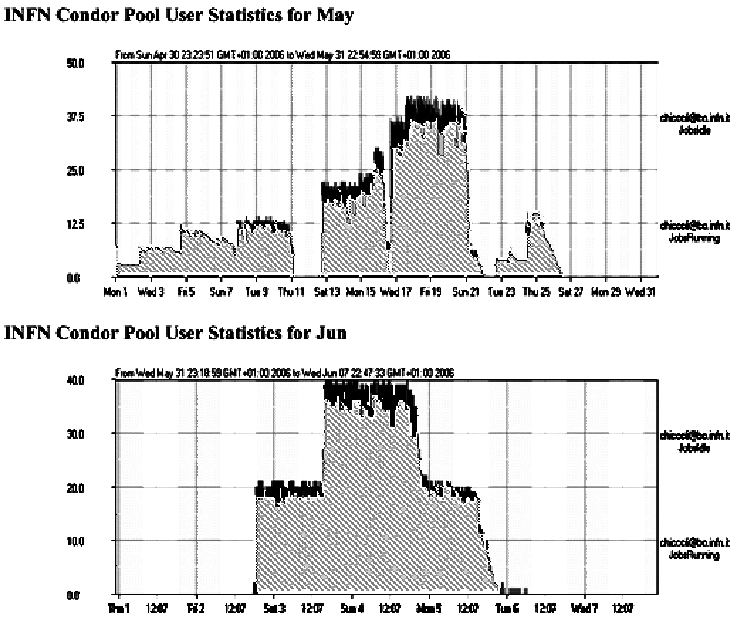


Fig. 7. Total number of idle (black) and running (grey) jobs submitted during May and June 2006, when about 10000 hours of computing time were used.

some of the sequences to the slowest nodes should not be allowed, to avoid waiting for their completion before being able to analyzing the full sequences of results. In general, our computational strategy is to spread each sequence over the maximum number of different nodes. Once each RUN (with INPUT and OUTPUT operations) is completed the computational sequence proceeds by submitting a new job having assigned the first available node of the Condor pool, without doing any explicit choice for this. This avoids a unique allocation for the fastest available resources, thus not penalizing the other users of the pool.

The results of this strategy are shown in Fig 6. The simulation sequences have different computational length because of the different kind of analysis to be performed, but the average computing time for each RUN is similar for each sequence as can be seen in Fig. 6. As mentioned before, this kind of simulations requires a very large amount of calculations. In Fig. 7 we show, as an example, the total number of running and idle jobs over two months, required for part of the study, for which a total of 10000 hours of computing time were used.

## 4. Conclusions

We have shown that the use of distributed computing, as already preliminarily tested on a much smaller scale [15], can be essential when a very large number of

jobs is required to pursue a systematic investigation of a physical system by means of computer simulations. In the case presented here we have studied a system under different conditions for which a significant effort is needed for completing the simulations. In particular we have modeled a Liquid Crystal-Polymer composite lattice system for: i) different compositions, requiring a change in the percentage of frozen particles mimicking the polymer chains, ii) different geometrical distributions and particle orientations inside the polymer layers, iii) various boundary conditions, iv) some field strengths and v) various temperatures. The calculations were distributed over about 130 nodes of a CONDOR pool of the National Institute for Nuclear Physics with the only constraint to have at least 200 MIPS of computational power for each host. We found this arrangement a satisfactory one and we plan to further extend the present approach to other, more complex, systems.

## References


1. P. G. de Gennes and J. Prost, *The Physics of Liquid Crystals*, 2nd ed., (Oxford University Press, 1993).
2. D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, 2000).
3. P. Pasini and C. Zannoni (eds.), *Advances in the Computer Simulations of Liquid Crystals*, (Kluwer, Dordrecht, 2000).
4. P.A. Lebwohl and G. Lasher, *Phys. Rev. A* **6** , 426 (1972).
5. U. Fabbri and C. Zannoni, *Mol. Phys.* **58**, 763 (1986).
6. F. Biscarini, C. Chiccoli, P. Pasini, F. Semeria and C. Zannoni, *Phys. Rev. Lett.* **75**, 1803 (1995).
7. C. Zannoni, *J. Mater. Chem.* **11**, 2637 (2001).
8. C. M. Care and D. J. Cleaver, *Rep. Progr. Phys.* **68**, 2665 (2005).
9. M. R. Wilson, *Internat. Rev. Phys. Chem.* **24**, 421 (2005).
10. http://www.cs.wisc.edu/condor/
11. Tiezheng Qian, Jae-Hoon Kim, S. Kumar and P.L Taylor, *Phys. Rev. E* **61**, 4007 (2000).
12. A. Marino, F. Vita, V. Tkachenko, R. Caputo, C. Umeton, A. Veltri and G. Abbate, *Eur. Phys. J. E* **15**, 47 (2004).
13. C. Chiccoli, O. D. Lavrentovich, P. Pasini and C. Zannoni, *Phys. Rev. Lett.* **79**, 4401 (1997).
14. C. Chiccoli et. al. *Monte Carlo simulations of a composite liquid crystal polymer system* (in preparation).
15. C. Chiccoli, P. Pasini, F. Semeria, C. Zannoni, *Lecture Notes in Computer Science, Springer* **2110**, 555-560 (2001).