Parallel Monte Carlo simulation of Lebwohl-Lasher systems

S. Boschi
High Performance System Division, CINECA

C. Zannoni
Dipartimento di Chimica Fisica ed Inorganica, Università di Bologna

C. Chiccoli, P. Pasini
INFN, Bologna

M. C. P. Brunelli
Istituto Nazionale Fisica della Materia, Modena

Abstract

I sistemi Lebwohl-Lasher sono stati studiati approfonditamente in passato come modello molto semplificato di cristalli liquidi. Nonostante la semplicità, il modello riproduce molto bene l'ordine orientazionale dei cristalli liquidi nematici reali e molte proprietá della transizione di fase isotropo-nematico, incluso il suo carattere del primo ordine "debole". Con il calcolo ad alte prestazione su calcolatori paralleli é stato possibile lanciare una simulazione di grandi dimensioni, con N = 1728000 particelle, eseguendo la scansione della transizione di fase mostrata da questo sistema, ed analizzarne il comportamento in dettaglio.

Lebwohl-Lasher systems have been extensively studied in the past as very simple models for liquid crystals. Although the model is very simple, it reproduces very well the orientational ordering of real nematic liquid crystals and many of the characteristics of the isotropic-nematic phase transition, including its weak first order character and pretransitional properties. With high performance computation on parallel computers it has been possible to run a large simulation, with N = 1728000 particles, scanning the phase transition exposed by this system and analyse its behavior with high detail.

1 Introduction

Lebwohl-Lasher [1,2] systems have been extensively studied in the past as very simple models for liquid crystals [3]. In this model a system of uniaxial particles placed at the sites of a cubic lattice interacts through the nearest-neighbor pair potential:

$$U_{ij} = -\varepsilon_{ij}\left(\frac{3}{2}\cos^2\beta_{ij} - \frac{1}{2}\right) \tag{1}$$

where β_{ij} is the angle between the axis of the i-th and j-th particle, and ε_{ij} is a positive constant, ε, for nearest-neighbour particles, and zero otherwise.

Although the model is very simple, it reproduces very well the orientational ordering of real nematic liquid crystals and many of the characteristics of the isotropic-nematic phase transition, including its weak first order character and pretransitional properties. On the other hand the precise location of the transition and its exponents are still not available, making it very interesting to investigate the behaviours of the model in very large systems.

2 Monte Carlo simulations

From the interaction potential presented above, it is possible to determine the equilibrium state of a system of interacting particles at a certain temperature with the Monte Carlo method [4, 5].

The implementation of the Monte Carlo method with the Lebwohl-Lasher potential is very simple: each particle is represented using its director cosines. The relationship between them and the Euler angles is:

$$\begin{aligned} v_x &= \cos\alpha\sin\beta \\ v_y &= \sin\alpha\sin\beta \\ v_z &= \cos\beta \end{aligned} \tag{2}$$

And the cosine of the angle between two orientations is the scalar product. So eq. 1 becomes:

$$U_{ij} = -\varepsilon_{ij}\left(\frac{3}{2}\left(v_i \cdot v_j\right)^2 - \frac{1}{2}\right) \tag{3}$$

In order to perform the simulations, all the involved quantities are dimensionless.

3 The parallelization of the Monte Carlo

One of the requirements of the Monte Carlo method is that single random moves must be performed in sequence, in order to build up the Markov chain that finally converges to the desired distribution.

This requirement makes it impossible to evolve concurrently and independently different parts of the system.

On the other hand the method says nothing about the order in which such moves must be performed. Thus, it is with a smart choice of the order in which the moves are done that it is possible to split the system in quite independent parts without breaking the Markov chain.

Let's take a system of L×L×L particles, L even (such an assumption can be easily relaxed). Each particle needs three coordinates in order to be stored.

Usually, in order to apply the periodic boundary conditions, it is necessary to test where the particles are: if the particle is on the left boundary, its left neighbour is on the right boundary.

We can decide on the other hand to store the system in an array dimensioned to (L+2)×(L+2)×(L+2), with indexes running from 0 to L+1. In such a way the system itself is stored where all the indexes are between 1 and L, while in the frames around it, copies of the boundary particles are stored: in this way there is no need for any "if". In order to keep track of periodic boundary conditions we need just to copy the particles from the boundary to the frame on the other side, before the sys-

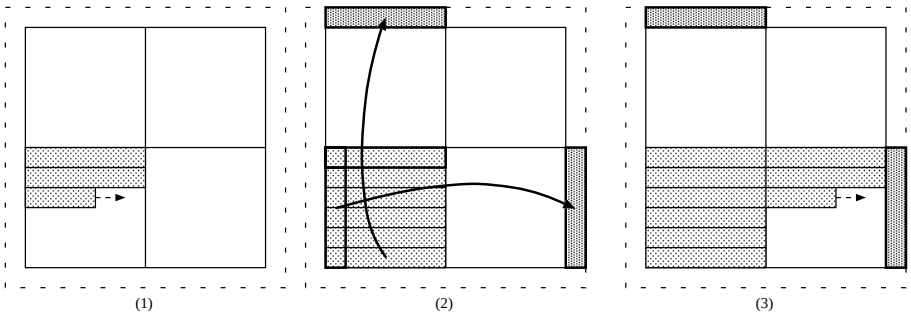

Figure 1 Subdivision in subparts and consequent frame updates in the single processor implementation. First the evolution is performed inside one of the subparts (1). There is no need of updating the corresponding frame because the rest of the system is not moved. After all the particles of the subpart have been moved, the ones on the neighbours can be copied to the corresponding frames (2), in order to be ready for the evolution of the next subparts (3). Just 2 dimensions are represented here for simplicity.

tem needs them.

Then we subdivide the system in 8 subparts, each of them L/2×L/2×L/2 large. One single subpart can be evolved separately without involving the update of the frames. In fact the corresponding particles at the boundary on the other side have not been moved. Only at the end of the subpart evolution the particles at the boundary are sent to the corresponding frames.

This is shown in fig. 1.

When such a structure is implemented, most of the work for a parallel algorithm is done. Let us suppose to have P^3 processors (with L/P even).

Now let's distribute the problem, allocating on each processor a matrix of $(L/P+2)^3$ elements: each processor owns $(L/P)^3$ particles to be moved, and a frame of $(L/P+2)^3-(L/P)^3$ fixed particles. These frames will be copied from the neighbour parts of the system, allocated on the other processors.

In order to avoid to move concurrently interacting particles, again, each of the parts of the problem is divided in 8 subparts, of $(L/(2P))^3$ particles, evolved at the same time and independently.

After each subpart is completely moved, the updated positions on the boundaries have to be communicated to the neighbouring processors frames (fig. 2).

The Markov chain is not broken since the system evolves as well as the simulation is performed on one processor choosing a particular order of the moves.

By assigning neighbours in the right way it is automatic to get the periodic boundary conditions. This can be done imposing that the leftmost processor is the right

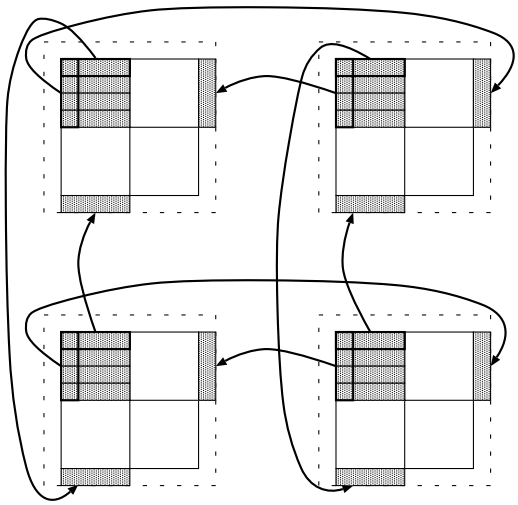

Figura 2 System distributed between 4 processors. Each processor has evolved the up-left subpart and is communicating it to all the processors. In this way the periodic boundary conditions are maintained by the topology of the communication. The communications in the x and in the y directions here are shown together but they are performed in two distinct and independent steps.

neighbour of the rightmost processor.

These arguments apply to the other directions in the same way.

4 The lagged-Fibonacci random number generator

The random number generator is probably the most delicate point in the Monte Carlo method. It is important to have numbers "as random as possible" in order for the method to converge correctly to the Boltzmann distribution. It is very difficult to provide an operational definition corresponding to this statement, and currently mathematicians are still working in order to do it rigorously [7]. What can be safely said is that there are some generators that behave well, other less, in running simulations [8].

The lagged-Fibonacci generator has a very important feature for parallel applications: it has many independent sequences, all of them as long as the maximum period of the generator. It is possible to initialize randomly $2^{31}-1$ of them starting from the Park and Miller generator [10]: such an approach is described in detail by Pryor *et al* [11].

Similarly to the other generators, the lagged-Fibonacci generator has a iterative form:

$$X_n = X_{n-l} + X_{n-k} \bmod m \quad \text{where} \quad l > k > 0 \tag{4}$$

The generator used in our simulations has $l = 17$, $k = 5$, $M = 32$ ($m = 2^M$): LFG(17,5,32).

5 Observables

The thermodynamic quantities we are interested in are the per-particle energy $\langle U \rangle$, the second rank order parameter $\langle P_2 \rangle$, the fourth rank order parameter $\langle P_4 \rangle$ and the pair correlation functions, which will be treated in detail in the next section.

The energy is evaluated at each move and the Monte Carlo engine needs it in order to work. Each move contributes to the modification of the total energy and we need just to average it over all the processors of the parallelized code each time we need to print it.

The order parameters are the coefficients of the expansion of the orientational distribution $P_L(\cos\beta)$. The inverse relationship is:

$$P(\cos\beta) = \sum_{L=0}^{\infty} \frac{2L+1}{2} \langle P_L \rangle P_L(\cos\beta) \tag{5}$$

In the Lebwohl-Lasher system the particles do not distinguish “up” and “down” and so the odd order parameters are zero.

The order parameters $\langle P_2 \rangle$ and $\langle P_4 \rangle$ and the director **n** of the system can be computed from the diagonalization of a order matrix **Q** defined in [3].

6 The pair correlation functions

The rotationally invariant pair distribution $G(r_{12},\omega_{12})$ [12] describes the probability of finding two particles at distance r_{12} with orientation ω_{12}. The coefficients of the expansion of $G(r_{12},\omega_{12})$ in terms of Legendre polynomials:

$$G(r_{12},\omega_{12}) = G_0^{00}(r_{12}) \sum_L \frac{2L+1}{64\pi^4} G_L(r_{12}) P_L(\cos\beta_{12}) \tag{6}$$

give the correlation between the orientation of two particles at distance r_{12}:

$$G_L(r_{12}) = \frac{1}{G_0^{00}(r_{12})} \int d\omega_1 d\omega_2 G(r_{12}\omega_{12}) P_L(\cos\beta_{12}) = \tag{7}$$

$$= \langle P_L(\cos\beta_{12}) \rangle_{r_{12}} \tag{8}$$

where $G_0^{00}(r_{12})$ is the positional radial distribution, which is constant in a system of fixed particles.

The pair correlation functions $G_2(r_{12})$ and the $G_4(r_{12})$ may be computed in a Monte Carlo simulation using the following procedure.

For each pair of particles *ij* in the system we should evaluate their distance r_{ij}, $P_2(\cos\beta_{ij})$ and $P_4(\cos\beta_{ij})$, averaging the values in an histogram, whom bins have length ΔR. Defining

$$d(r,\Delta r) = \begin{cases} 1 & \text{if} \quad r \in \left[r - \frac{\Delta r}{2}, r + \frac{\Delta r}{2} \right[\\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Even if we were able to reach the peak performance (1.2 GFlops) of the 64 alpha processors, the task of the full evaluation of the correlation function for N=1728000 would be very heavy, since the number of pairs in our system is about $1.5 \cdot 10^{12}$.

So we need to simplify the problem. We could think to calculate the correlation function for just one particle with respect to the rest of the sample, but this would not average spatially the correlation function. So we have chosen to select a subset $\{N\}_s$ of equally spaced particles in all the sample:

$$N_R = \sum_i^{\{N\}_s} \sum_j^N d\left(r_{ij}, \Delta R\right) \tag{10}$$

$$\left\langle P_2 \right\rangle_R = \frac{1}{N_R} \sum_i^{\{N\}_s} \sum_j^N d\left(r_{ij}, \Delta R\right) P_2\left(\cos \beta_{ij}\right) \tag{11}$$

$$\left\langle P_4 \right\rangle_R = \frac{1}{N_R} \sum_i^{\{N\}_s} \sum_j^N d\left(r_{ij}, \Delta R\right) P_4\left(\cos \beta_{ij}\right) \tag{12}$$

Even in this way the calculation of the correlation functions contributes to about the 15% of the total simulation time.

7 Scalability of the code

We have performed some timings on our target model (120 ×120×120 particles, with periodic boundary conditions) on the Cray T3E/1200 (256 Alpha EV67 processors at 600MHz) and on the SGI Origin2000 (64 R12000 processors at 300MHz) installed at CINECA.

The timings have been done on the "pure" Monte Carlo algorithm, without the calculation of both the correlation function (its algorithm require minimal communication with respect to the computation) and observables (I/O could be relevant in this task, even if it is not accomplished frequently, once every 50 cycles in the production runs).

The timings are reported in fig. 3. These show a very good scalability of the code on the T3E, with an efficiency of about 80% on 96 processors.

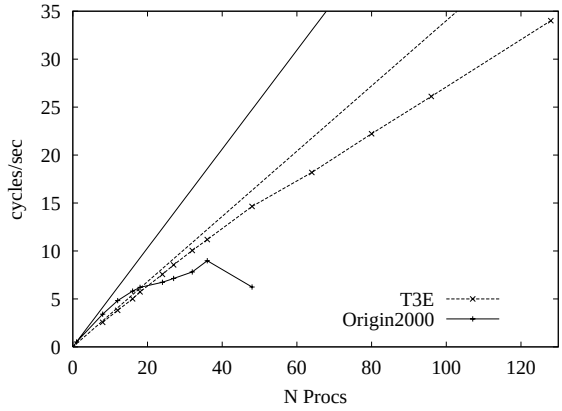

Figure 3 Scalability test of the MC code on Cray T3E/1200 and SGI Origin2000 R12000/300MHz. The straight lines represents the theoretical scale-up, calculated from the single processor runs, and the points are the cycles per second on the two machines, reported in the corresponding table. The scalability is almost linear for the T3E. On the other side the code is more efficient on the processor of the Origin2000.

The scalability is not so good on the Origin2000: even if the code is faster on the R12000 processor (about 1.5 times), with more than 16 processors it is more efficient on the T3E.

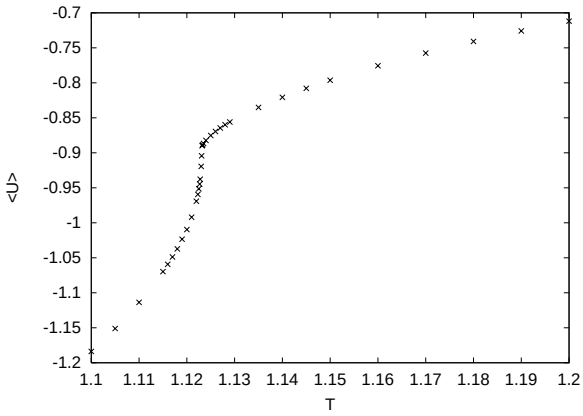

Figure 4 Averaged energy vs. temperature. Each point has been evaluated and averaged from a set of configurations coming from the Monte Carlo simulation at the temperature in abscissa.

8 Results

The main aim of this work is to determine the nematic-isotropic transition temperature, and compare other properties observed in real liquid-crystals or predicted by the theory.

In section 8.1 we describe the simulations performed in the course of this work.

In section 8.2 we show the main results coming out from the simulations.

In section 8.3 the heat capacity and the transition temperature are computed.

In section 8.4 we analyze the pair correlation functions, showing how this system exhibits pretransitional effects, as the same as real liquid-crystals do.

8.1 Performed simulations

The simulations at the first temperature have been started from a crystal structure (all the particles aligned with the z axis). Then the temperature has been increased slowly from T = 1.1 to the highest investigated temperature T = 1.2.

For each temperature we have run from 100,000 to 2,400,000 Monte Carlo cycles. Temperatures close to the transition need more cycles both to equilibrate and to average correctly the data. In all the simulations the acceptance ratio (the number of moves accepted with respect the total number of tried moves) was in between 0.3 and 0.6, showing that the Monte Carlo method was working correctly.

8.2 Observed energy and order parameters

The averaged energies performed at each temperature are reported in fig. 4. The figure fig. 5 shows the averages of the order parameters, $\langle P_2 \rangle$ and $\langle P_4 \rangle$.

From the three plots it is evident the presence of a phase transition of the first order at $T_{NI} \approx 1.123$. First order transitions are characterized by a discontinuity of the heat capacity, which is the first derivative of the energy. A characteristic feature of a phase transition is the sudden loss of the macroscopic order. At temperatures higher than T_{NI} (see fig. 5) the order parameters drop quickly to zero, showing that the phase transition is characterized by the disappearance of the order in the system.

8.3 The heat capacity

The heat capacity of a system at constant volume is the derivative of the internal energy with respect to the temperature:

A numerical derivative of the energies in fig. 4 using a direct approach like a finite difference method would not give the transition temperature to a good precision.

It is therefore necessary to fit the energies against a smoothing spline with local convexity constraints [13]. The spline has been constrained to be convex in the cold side before the transition ($T \leq 1.1228$), concave in the hot part ($T \geq 1.232$), and free on the points in between.

The fitting procedure allows to assign weights to data. In the fit of the energy in fig. 6 the

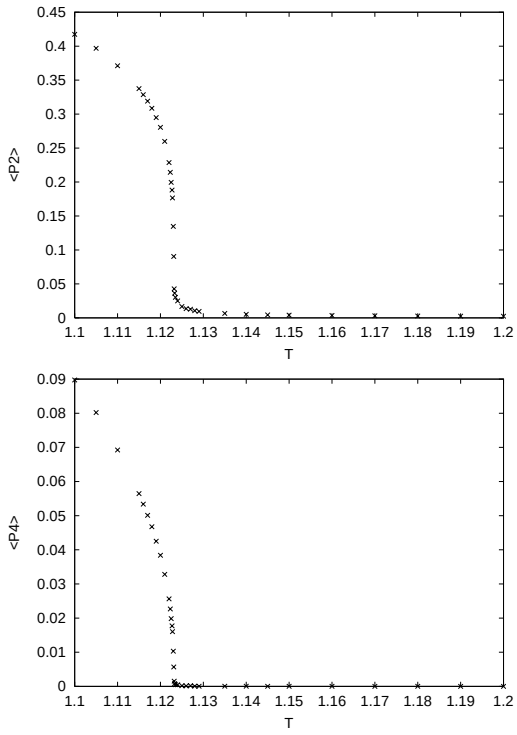

Figure 5 Averaged $\langle P_2 \rangle$ and $\langle P_4 \rangle$ order parameter vs. temperature.

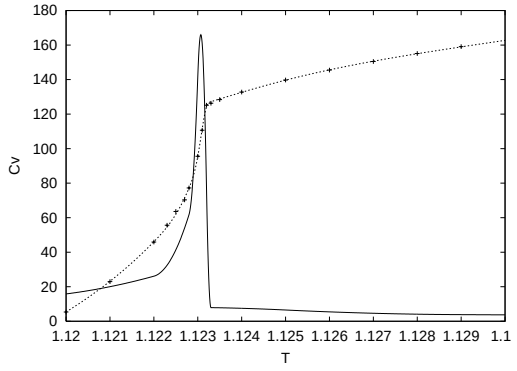

Figure 6 Heat capacity vs. temperature (continuous line), obtained by numerical derivation of a spline (dashed line, not in the same scale) used to fit the energies (points, not in the same scale).

weights are the reciprocal the statistical errors.

In order to fit the spline the data have been weighted with the reciprocal of their statistical error.

The spline which interpolates the energy and the heat capacity are reported in fig. 6.

The phase transition is located by the maximum of the heat capacity:

$$T_{NI} = 1.12307$$

8.4 Correlation functions analysis

The magnetic susceptibility κ of a Lebwohl-Lasher system of N spins with boundary conditions coupled to an external magnetic field B is:

$$\kappa = \frac{\lambda}{T} g_2 \tag{14}$$

where λ is the parameter which tunes the interaction of B with the system, and g_2 is the zero-field average of:

$$g_2 = \frac{5}{N} \sum_i \sum_j < \mathrm{P2}(\cos\beta_t) P_2(\cos\beta_j) > \mathrm{B} = 0 \tag{15}$$

In a cubic lattice system g_2 can be reformulated as a sum over lattice points of:

$$g_2 = \sum_k z_k G_2(r_k) \tag{16}$$

where G_2 is the correlation function defined in eq. 8, and z_k is the number of neighbours in the shell at distance r_k.

There is an experimental evidence of the divergence of κ at a temperature T_{NI^*} smaller and close to T_{NI} [14,15], which can be detected in the simulations as well.

In fig. 7 is depicted the inverse of the susceptibility $1/(\kappa\lambda) = T/g_2$ versus the temperature on the hot side of the transition. The fit of a linear dependence of $1/(\kappa\lambda)$ from the temperature has been found to reproduce the simulated data within an error of 5%. The divergence temperature T_{NI^*}, which can be extrapolated from the fit parameters requiring that $1/(\kappa\lambda) = 0$, has found to be:

$$T_{NI^*} = 1.12305 \tag{17}$$

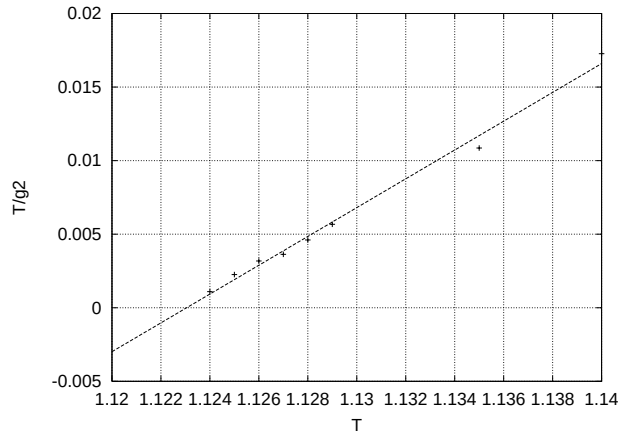

Figure 7 Simulated values of $1/(\kappa\lambda) = T/g_2$ versus T (dots), and linear fitting function (line).

9 Conclusions

The massively parallel implementation of the simulation program has allowed the full simulation of this large system.

It is quite evident the system exhibits many features common to real systems, that can be summarized in two points:

A nematic-isotropic phase transition of almost first order at

$$T_{NI} = 1.12307$$

pretransitional effects, at temperature lower than the above transition temperature, characterized by a divergence temperature

$$T_{NI^*} = 1.12305$$

The work still needs some refinements, in particular some temperatures close to the transition have to be simulated and some more analyses are needed.

10 Acknowledgments

We thank INSTM (*National Institute for the Science and Technology of Materials*) for granting part of the computing time.

Publications and References

[1] P.A.Lebwohl, G.Lasher, Phys. Rev. A, 6, 426 (1972).

[2] C.Zannoni, *The molecular Physics of Liquid Crystals*, edited by G.R.Luckhurst and G.W.Gray - Academic Press (1979).

[3] U.Fabbri, C.Zannoni, Mol. Phys., 58, 763 (1986).

[4] M.P.Allen, D.J.Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford (1987)

[5] D. Frenkel, B. Smit, *Understanding Molecular Simulations - From Algorithms to Applications*, Academic Press (1996)

[6] A.M.Ferremberg, D.P.Landau, Y.J.Wong, Phys. Rev. Lett., 69 (1992)

[7] S.Pincus, R.E.Kalman, Proc. Natl. Acad. Sci. USA, 94, 33513–3518 (1997).

[8] Vattulainen, T. Ala-Nissila, K.Kankaala, Phys. Rev. Lett, 73, 2513 (1994).

[9] M.Mascagni, S.Cuccaro, D.Pryor and M.Robinson, *A fast, high quality, reproducible, parallel, lagged-Fibonacci pseudo random number generator*, SRC-TR-94-115, Super-computing Research Center, 17100 Science Drive, Bowie, MD 20715, 1994.

[10] S.K.Park, K.W.Miller, *Random Number Generators:Good Ones are Hard to Find - Transactions of the ACM* (Nov. 1988)

[11] D.Pryor, S.Cuccaro, M.Mascagni, M.Robinson, *Implementation and usage of a portable and reproducible parallel pseudo random number generator*, SRC-TR-96-116, Super-computing Research Center, 17100 Science Drive, Bowie, MD 20715, 1994.

[12] P.G. De Gennes, *The Physics of Liquid Crystals*, Clarendon press (1974).

[13] P.Dierckx, *Curve and surface fitting with splines*, Monographs on numerical analysis, Oxford University press (1993).

[14] T.D.Gierke, W.H.Flygare, J. Chem. Phys, 61, 2231 (1974).

[15] G.S.Attard, P.A.Beckmann, P.A.Emsley, J.W.Luckhurst, D.L.Turner, Mol. Phys., 45, 1125 (1982).