TECHNICAL REPORTS
ALCOM SYMPOSIUM

Volume III

Liquid Crystals: Surfaces and Finite Size Effects

March 12–13, 1992
Case Western Reserve University
Cleveland, Ohio

Finite Difference Minimization of the Landau - de Gennes Free Energy for Liquid Crystals in Ellipsoidal Regions

Paul A. Farrell and Arden Ruttan

Department of Mathematics and Computer Science
Kent State University

ABSTRACT

We will describe a finite difference code for computing equilibrium configurations, of the order-parameter tensor field for nematic liquid crystals, in ellipsoidal regions, by minimization of the Landau - de Gennes Free Energy functional. Our implementation of the free energy functional includes linear and quadratic surface terms, electric and magnetic fields, all three quadratic gradient terms, and scalar bulk terms through sixth order. Boundary conditions include the effects of both strong and weak surface anchoring. Studying such problems in general ellipsoidal regions permits not only the modelling of traditional spherical regions, free from surface defects, but also the symmetry breaking effects of surface distortions and defects.

The target architectures for our implementation are SIMD machines, with 3 dimensional rectangular grids, such as the Wavetracer DTC in the department of Mathematics and Computer Science, and with hypercube networks such as the Thinking Machines Corporation CM-2.

1. Introduction: The Problem. We consider a finite difference approximation of the equilibrium configuration of liquid crystals in an ellipsoidal region,

$$\Omega = \{(ar\sin(\phi)\cos(\theta), br\sin(\phi)\sin(\theta), cr\cos(\theta)) : 0 \le r \le 1,\ 0 \le \theta < 2\pi,\ 0 \le \phi < \pi\},$$

where a, b and c are the three semi-axes of the ellipsoid. Following Gartland ([2]), we will use the Landau-de Gennes formulation, and seek a tensor order parameter field Q that minimizes the free energy of the system. In this case, the free energy can be expressed as

$$\begin{aligned} F(Q) &= F_{\text{vol}}(Q) + F_{\text{surf}}(Q) \\ &= \int_\Omega f_{\text{vol}}(Q) + \int_{\partial\Omega} f_{\text{surf}}(Q)\,, \end{aligned}$$

where Q(x) is a 3×3 symmetric, traceless tensor and where Ω and $\partial\Omega$ represent the interior and surface of the ellipsoid respectively.

The term $F_{\text{vol}}(Q)$ represents an approximation of the interior free energy. To describe $F_{\text{vol}}(Q)$, we will use the convention that summation over repeated indices is implied and that indices separated by commas represent partial derivatives. With these conventions $F_{\text{vol}}(Q)$ has the following form, (see, for instance [4]):

$$\begin{aligned} f_{\text{vol}}(Q) :=\ & \tfrac{1}{2}L_1 Q_{\alpha\beta,\gamma}Q_{\alpha\beta,\gamma} + \tfrac{1}{2}L_2 Q_{\alpha\beta,\beta}Q_{\alpha\gamma,\gamma} + \tfrac{1}{2}L_3 Q_{\alpha\beta,\gamma}Q_{\alpha\gamma,\beta} \\ +\ & \tfrac{1}{2}A\,\text{trace}(Q^2) - \tfrac{1}{3}B\,\text{trace}(Q^3) + \tfrac{1}{4}C\,\text{trace}(Q^2)^2 \\ +\ & \tfrac{1}{5}D\,\text{trace}(Q^2)\text{trace}(Q^3) + \tfrac{1}{6}E\,\text{trace}(Q^2)^3 + \tfrac{1}{6}E'\,\text{trace}(Q^3)^2 \\ -\ & \Delta\chi_{\max} H_\alpha Q_{\alpha\beta} H_\beta - \Delta\epsilon_{\max} E_\alpha Q_{\alpha\beta} E_\beta\,. \end{aligned} \tag{1}$$

where L_1, L_2, and L_3 are elastic constants; A, B, C, D, E, and E' are bulk constants; and H, $\Delta\chi_{\max}$, E, and $\Delta\epsilon_{\max}$ are the field terms and constants associated with the magnetic and electrical fields respectively.

Likewise, using the implied summation conventions, the surface free density f_{surf} has the form

$$f_{\text{surf}}(Q) := \frac{1}{2}V \cdot (\nu_\alpha Q_{\alpha\beta}\nu_\beta)^2 - W \cdot (\nu_\alpha Q_{\alpha\beta}\nu_\beta), \tag{2}$$

where ν is a unit vector field associated with the type of anchoring of the surface elements and V and W are prescribed constants.

For $P \in \Omega$, the tensor $Q(P)$ will be represented in the form,

$$\begin{aligned} Q(P) &= (Q_{\alpha\beta})_{\alpha,\beta=1}^{3} \\ &= q_1(P)E_1 + q_2(P)E_2 + q_3(P)E_3 + q_4(P)E_4 + q_5(P)E_5 \\ &= q_1(P)\begin{pmatrix} \frac{\sqrt{3}-3}{6} & 0 & 0 \\ 0 & \frac{\sqrt{3}+3}{6} & 0 \\ 0 & 0 & \frac{-\sqrt{3}}{3} \end{pmatrix} + q_2(x)\begin{pmatrix} \frac{\sqrt{3}+3}{6} & 0 & 0 \\ 0 & \frac{\sqrt{3}-3}{6} & 0 \\ 0 & 0 & \frac{-\sqrt{3}}{3} \end{pmatrix} \\ &\quad + q_3(x)\begin{pmatrix} 0 & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + q_4(x)\begin{pmatrix} 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 \end{pmatrix} \\ &\quad + q_5(x)\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & \frac{-\sqrt{2}}{2} & 0 \end{pmatrix}, \end{aligned}$$

similar to that in Gartland [3], where $\{q_\ell(P)\}_{\ell=1}^{5}$ are real-valued functions on Ω.

2. The Implementation . To discretize our problem we begin by dividing the ellipsoid Ω into $(I-1) \times J \times K$ regions

$$v(i,j,k) = \{(ar\sin(\phi)\cos(\theta), br\sin(\phi)\sin(\theta), cr\cos(\theta))\},$$

for $1 \le i < I$, $0 \le j < J$, and $0 \le k < K$, where for given i, j, k,

$$\begin{aligned} &r_i - \tfrac{\Delta r}{2} \le r \le r_i + \tfrac{\Delta r}{2}, \\ &\phi_j - \tfrac{\Delta\phi}{2} \le \phi \le \phi_j + \tfrac{\Delta\phi}{2}, \\ &\theta_k - \tfrac{\Delta\theta}{2} \le \phi \le \theta_k + \tfrac{\Delta\theta}{2}, \\ &r_i = i\Delta r,\ \phi_j = j\Delta\phi,\ \theta_k = k\Delta\theta \\ &\Delta r = \tfrac{1}{I+\frac{1}{2}},\ \Delta\phi = \tfrac{\pi}{J},\ \text{and}\ \Delta\theta = \tfrac{2\pi}{K}, \end{aligned}$$

and one region centered at the origin

$$\begin{aligned} v(0,0,0) = \{&(ar\sin(\phi)\cos(\theta), br\sin(\phi)\sin(\theta), cr\cos(\theta)) : 0 \le r \le \Delta r,\ 0 \le \theta < 2\pi, \\ &0 \le \phi < \pi\}. \end{aligned}$$

We represent the discrete interior free energy integral by

$$\int_\Omega f_{\text{vol}}(Q) \approx \sum_{i,j,k} f_{\text{vol}}(Q(r_i, \phi_j, \theta_k)) \times volume(v(i,j,k)). \tag{3}$$

Note that (1) involves derivatives of the functions $\{q_\ell(P)\}_{\ell=1}^5$ with respect to the x, y, and z coordinates. The spherical coordinate discretization makes the representation of these derivatives slightly complicated. For most interior points P, these derivatives can be approximated using the chain rule and central difference approximations of the derivatives of $\{q_\ell(P)\}_{\ell=1}^5$ with respect to r, θ and ϕ. Since central difference approximations use data from only immediately adjacent points, for interior points at least, $f_{\rm vol}(Q(P))$ can be approximated using the values of $\{q_\ell(P)\}_{\ell=1}^5$ and the values of those functions at the 6 points immediately adjacent to P.

When $r = 0$ or $\phi = \pm\pi$, the derivatives with respect to r, θ and ϕ do not exist, so these cases require special handling. However, if we require that $\pi/2$ be an integer multiple of both $\Delta\theta$ and $\Delta\phi$ (say $\pi/2 = k_0\Delta\theta$ and $\pi/2 = j_0\Delta\phi$), then, for instance,

$$\frac{\partial q_\ell(r_0,\phi_0,\theta_0)}{\partial x} \approx \frac{q_\ell(r_1,\phi_{j_0},\theta_0) - q_\ell(r_1,\phi_{j_0},\theta_{k_0})}{2\Delta r}.$$

In a similar manner, one can approximate the other partial derivatives of $q_\ell(r_0,\phi_0,\theta_0)$ as well as those of $q_\ell(r_i,\phi_0,\theta_k)$ and $q_\ell(r_i,\phi_{2j_0},\theta_k)$ (which lie on the positive and negative z-axis) using only data from immediately adjacent points. Therefore, even for points P which lie on the z-axis, we can approximate $f_{\rm vol}(Q(P))$ accurately using the values of $\{q_\ell(P)\}_{\ell=1}^5$ at P and at 6 points immediately adjacent to P.

By our choice of Δr , the points on the surface of the ellipsoid are

$$(r_{I+1},\phi_j,\theta_k) = ((I+\frac{1}{2})\Delta r, j\Delta\phi, k\Delta\theta),$$

for $0 < j < J$, $0 \le k < K$. Upon setting

$$a(j,k) = \{(r_{I+1},\phi_j,\theta_k) : 0 < j < J,\ 0 \le k < K\},$$

we can approximate the surface energy by

$$\int_{\partial\Omega} f_{\rm surf}(Q) \approx \sum_{j,k} f_{\rm surf}(Q(r_{I+1},\phi_j,\theta_k)) \times area(a(j,k)), \tag{4}$$

where $f_{\rm surf}(Q(r_{I+1},\phi_j,\theta_k))$ can be evaluated using only the data at the point $(r_{I+1},\phi_j,\theta_k)$.

Using (3) and (4), we have the following approximation of the **Landau-de Gennes free energy** which is essentially second order accurate[1]:

$$\begin{aligned} F(Q) &\approx \textstyle\sum_{i,j,k} f_{\rm vol}(Q(r_i,\phi_j,\theta_k)) \times volume(v(i,j,k)) \\ &+ \textstyle\sum_{j,k} f_{\rm surf}(Q(r_{I+1},\phi_j,\theta_k)) \times area(a(j,k)) \\ &= \textstyle\sum_{i,j,k} h((r_i,\phi_j,\theta_k)), \end{aligned} \tag{5}$$

where $h((r_i,\phi_j,\theta_k))$ can be evaluated using the values of $\{q_\ell\}_{\ell=1}^5$ at (r_i,ϕ_j,θ_k) and six adjacent points.

[1] Since the first order accurate approximations, which come from using the chain rule near ϕ=0 and $r = 0$ represent only a small fraction of the total, numerical experience indicates that the solution will remain second order accurate.

With the discretization (5), we have reduced the problem to one of minimizing

$$\min \sum_{i,j,k} h((r_i, \phi_j, \theta_k)) \tag{6}$$

over all choices of $\{q_\ell(r_i, \phi_j, \theta_k)\}_{\ell=1}^5$. This unconstrained discrete minimization problem can be attacked in the standard way. That is, seek a solution of the non-linear system of equations

$$g(\hat{\ell}, \hat{i}, \hat{j}, \hat{k}) := \frac{\partial \sum_{i,j,k} h((r_i, \phi_j, \theta_k))}{\partial q_\ell(r_{\hat{i}}, \phi_{\hat{j}}, \theta_{\hat{k}})} = 0, \tag{7}$$

for $0 \leq \hat{i} \leq I+1$, $0 \leq \hat{j} \leq J$, $0 \leq \hat{k} \leq K$, and $\hat{\ell} = 1 \ldots 5$. A standard approach, to solving non-linear systems such as these, is to use Newton's method (see [1]).

Each iteration of Newton's method involves solving a linear system, whose matrix is the Jacobian of (7), and then using that solution to update the iterate and the Jacobian, after which the process is repeated. For the three dimensional problem under consideration, this linear system is extremely large. A moderate sized problem would involve 10^7 unknowns. As our ultimate aim is to track the minimal energy state as the temperature varies, we need to be able to solve the minimal energy problem, as described above, rapidly and to moderate accuracy for each of a large number of successive temperatures. As such a task is probably only feasible on a massively parallel computer, we choose to implement our minimization problem on such a computer. In the next section, we will describe the type of computer architecture we use for our implementation, and sketch the details of that implementation.

3. The Architecture. Our target architecture is a massively parallel SIMD computer. This class of computers, is called Single Instruction Multiple Data stream (SIMD), since each of the processors execute the same instruction in lock-step on the data stored in its associated local memory. A facility is also provided to select a subset of processors, by masking the others processors. Only the selected processors execute the instruction, while the masked processors remain idle. This architecture is considerably simpler to implement and program than the alternative Multiple Instruction Multiple Data stream (MIMD) machines, in which each processor can execute a different instruction. The SIMD architecture is normally used for massively parallel machines, having between 4096 and 65536 processors, each with local memory. The processors are normally connected by a special purpose high-capacity communication network and controlled by a front-end processor. Early examples of this architecture included the Thinking Machines Corporation Connection Machine CM-1 and CM-2.

The platform chosen for this implementation was the Wavetracer Data Transport Computer (DTC), situated in the Department of Mathematics and Computer Science at Kent State. The front-end sends instructions and data to a control unit, which decodes these instructions and broadcasts both instructions and data to the processor array. The array processors are semi-custom 1.5 micron standard cell chips. Each chip contains 32 one-bit processors together with 2 kilobits of fast static RAM for each

processor, and associated control and memory error-detection circuitry. In addition, each processor has access to 8 or 32 kilobytes of private external dynamic memory depending on the configuration. Each circuit board consists of 128 chips. The minimal configuration, the DTC-4, has one circuit board and thus 4096 processors. Other configurations are the DTC-8, with 2 circuit boards and 8192 processors, and the DTC-16, with 4 circuit boards and 16384 processors.

The processors on each circuit board of the DTC can be configured either as a $16 \times 16 \times 16$ cube, for three dimensional applications, or as a 64×64 square, for two dimensional applications. The DTC-8, can be configured as a $16 \times 32 \times 16$ cube or as a 64×128 square, and the DTC-16 as a $32 \times 32 \times 16$ cube or as a 128×128 square. The assumption here is that most applications, correspond to physical problems in 2 or 3 dimensions, and thus a 2 and 3 dimensional interconnection network is the most efficient for their solution. This is in contrast to the Connection Machine, in which the processors are connected by a hypercube network.

The traditional mode of solution of problems on a SIMD machine involves assigning one processor of the array per node in the problem space. To provide the ability to consider problems with more nodes than are available in the array, the DTC provides the ability to partition the memory of each processor to provide a larger number of *virtual processors*. There must be the same number of virtual processors for each physical processor. The number of virtual processors per physical processor is called the *virtual processor ratio*. The controller automatically issues instructions to the array once for each partition. Thus the execution time may be expected to increase linearly with the virtual processor ratio.

For the minimization problem we are considering, each discretization point, P, of the ellipsoid is associated with a virtual processor. The spherical discretization we are using produces concentric grids of points where the ϕ and θ variables vary within each grid and changing r moves you from grid to grid. With the virtual processors arranged in a cube, each concentric grid of the spherical discretization maps onto a plane of the cube, and if we associate the z coordinate of the cube with the r coordinate of the ellipsoid, then each concentric grid of the spherical discretization will correspond to a $x - y$ plane of processors in the DTC.

At each point of the ellipsoid the tensor order parameter Q is defined in terms of the 5 unknowns $\{q_\ell(P)\}_{\ell=1,5}$. In our implementation, each set of 5 unknowns $\{q_\ell(P)\}_{\ell=1,5}$ is stored in a single virtual processor. For each $q_\ell(P)$ there is also a corresponding row of the Jacobian matrix. The nonzero constants of that row are also stored in the memory of the processor associated with the discretization point P. Each non-zero constant, in a row of the Jacobian associated with P, also corresponds to another virtual processor (which in turn corresponds to a discretization point) with which the values of $\{q_\ell(P)\}_{\ell=1,5}$ at P must be communicated when the Jacobian matrix is updated. The set of processors with which a given processor, P, must communicate in order to update its row of the Jacobian is called the stencil of P. If the stencil of any processor is large, then the process of updating the Jacobian at each step of Newton's method will be expensive. Fortunately, the finite difference approx-

imation yields a relatively small and compact stencil. In our formulation, each point of the stencil is no more than two steps away, and the total Jacobian matrix can be updated with approximately 210 communication steps (18 points two steps away and 6 points one step away, 18*2+6=42, 5*42=210) and the calculation of (24+6+1)*5 =155 row entries per virtual processor. If there are M physical processors, since each of the physical processors are working simultaneously, a SIMD computer will update M rows of the Jacobian matrix in the time it takes a sequential computer of similar speed to update 1 row.

As we mentioned above, the Jacobian matrix for this implementation is extremely large and sparse. Typical linear systems involving such matrices are solved with iterative methods the major component of which is a matrix vector multiply. The actual linear solver we will use in our implementation is still a subject of research. While the best sequential iterative methods may not be suitable for SIMD machines, there are iterative methods such as multi-colored SOR or multigrid which have convergence rates comparable to the best sequential methods for many problems and which achieve a substantial fraction of the M-fold parallelism of the SIMD computer. If we can, as we expect, achieve nearly M-fold parallelism in our linear solver along with convergence rates comparable to the best linear solvers, then our difference implementation will likewise achieve nearly an M-fold speedup over a similar finite difference approximation on a sequential machine.

REFERENCES

[1] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Engelwood, NJ, 1983.

[2] Eugene C. Gartland, On Some Mathematical and Numerical Aspects of the Landau-De Gennes Minimization Problem for Liquid Crystals, Preprint.

[3] E. C. Gartland, Jr., P. Palffy-Muhoray, and R. S. Varga, Numerical Minimization of the Landau-De Gennes free energy: Defects in cylindrical capillaries, Mol. Cryst. Liq. Cryst., 199(1991), pp. 249-452.

[4] E. B. Priestley, P. J. Wojyowicz, and P. Sheng, eds., *Introduction to Liquid Crystals*, Plenum Press, New York, London, 1975.