Numerical challenges in the application of density functional theory to biology and nanotechnology

2002 J. Phys.: Condens. Matter 14 12167

(http://iopscience.iop.org/0953-8984/14/46/320)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS: CONDENSED MATTER

J. Phys.: Condens. Matter **14** (2002) 12167–12187 PII: S0953-8984(02)37775-0

Numerical challenges in the application of density functional theory to biology and nanotechnology

L J D Frink[1], A G Salinger, M P Sears, J D Weinhold[2] and A L Frischknecht

Sandia National Laboratories, Albuquerque, NM 87185, USA

E-mail: ljfrink@sandia.gov

Received 7 June 2002
Published 8 November 2002
Online at stacks.iop.org/JPhysCM/14/12167

Abstract

This paper summarizes our efforts to develop fast algorithms for density functional theory (DFT) calculations of inhomogeneous fluids. Our goal is to apply DFTs to a variety of problems in nanotechnology and biology. To this end we have developed DFT codes to treat both atomic fluid models and polymeric fluids. We have developed both three-dimensional real space and Fourier space algorithms. The former rely on a matrix-based Newton's method while the latter couple fast Fourier transforms with a matrix-free Newton's method. Efficient computation of phase diagrams and investigation of multiple solutions is facilitated with phase transition tracking algorithms and arclength continuation algorithms. We have explored the performance that can be obtained by application of massively parallel computing, and have begun application of the codes to a variety of two- and three-dimensional systems. In this paper, we summarize our algorithm development work as well as briefly discuss a few applications including adsorption and transport in ion channel proteins, capillary condensation in disordered porous media and confinement effects in a diblock copolymer fluid.

1. Introduction

The behaviour of fluids at interfaces is fundamental to a wide range of natural and industrial processes. Well studied examples include the formation of double layers at charged interfaces (important to colloidal stabilization) [1], capillary condensation of liquids in small pores (important to the dessicant properties of zeolites and other nanoporous materials) [2, 3] and the wetting of surfaces by fluids [4]. Density functional theories (DFTs) have played an important

[1] Author to whom any correspondence should be addressed.
[2] Current address: Dow Chemical Corporation, Houston, TX 77090, USA.

role in understanding these phenomena. Over the past 20 years, several groups have worked toward developing accurate functionals to capture the behaviour of atomic fluid models at solid interfaces [5]. A variety of approaches have also been developed to treat molecular and polymeric fluids [6, 7]. While development of the underlying functionals will continue to be an important area for research, our efforts have been focused on developing the numerical tools needed to solve existing functionals for large two- and three-dimensional problems [8, 9]. This focus arises from a desire to apply DFTs to treat a wide range of problems in the areas of biology and nanotechnology.

Clearly, nanotechnologies seek to tailor materials and function on nanoscopic length scales. It is important to note that a variety of nanotechnologies have been developed to both measure surface forces to ångström resolution and manipulate individual atoms, molecules or macromolecules. These nanotechnologies include the surface forces apparatus [10a], the atomic force microscope [11] and the optical tweezers [12]. Other *in situ* measures of surface forces include osmotic stress measurements [13] and beam bending stress measurements [14]. These surface force measurements are often made in a solvated environment. It is therefore possible to apply molecular simulation and DFT approaches to compute these surface (or solvation) forces. A variety of detailed DFT and molecular simulation studies have been performed to both demonstrate the underlying statistical mechanical ensembles [15] involved in the experiments, and to interpret specific experimental results [16]. More recent developments in nanotechnology seek not only to measure the properties of atoms, but to control their arrangement and properties in complex assemblies. These assemblies are often formed in a fluid state, and there are a myriad of interfacial fluid problems that arise. A few examples of interfacial fluid problems in modern nanotechnology applications include patterned assembly via soft-lithography [17], self-assembly of silica based nanostructured materials via solvent evaporation [18] and controlled switching of the properties of fluid lipid bilayers via chemical recognition events [19].

Many nanotechnologies are, in fact, based on biomimetic principles. While highly tailored porous materials will be extremely useful for selective separations, efficient adsorption and the next generation of sensors, it is important to recognize that there are biological systems, ion channels, that already possess exquisite selectivity and sensitivity [20]. Ion channels are protein assemblies that span the lipid bilayers of cells. They control the ionic content of the cell, and in that way influence all of the processes inside the cell. Ion channels are highly selective for transport of a particular type of ion as a result of the specific three-dimensional structure and charge distribution of the proteins that make up the channel. These ion channels are very narrow, 0.5 nm, at the selectivity filter, and so the fluid in and around the pore is expected to be inhomogeneous. This is an ideal application for DFT approaches, but it is only one of many possible applications to biological systems. Other examples where there is a clear opportunity for the application of DFT include solvated interactions of proteins and drug (or toxin) targets, solvated interactions of biological macromolecules (DNA, RNA, proteins), the structure of lipid bilayers, permeation of lipid bilayers by small molecules, penetration of lipid bilayers by macromolecules (e.g. viruses) and routes to the assembly of protein complexes. In all of these cases, the behaviour of a fluid in the presence of *surfaces* must be understood. In the examples above the fluid is either an electrolyte or a lipid bilayer; the surfaces are now the proteins. The key complexity in studying biological systems is that proteins have specific geometries and charge distributions that correlate with their functions. As a result a three-dimensional DFT calculation is required.

While DFTs have not yet been broadly applied to biological systems, we note that liquid state theories have been applied to ion channel proteins, structure and small molecule permeation of lipid bilayers, and protein assembly in lipid bilayers [21]. In addition we note

that many calculations of protein solvation [22] and ion channel transport [23] have been performed at the level of a Poisson–Boltzmann (PB) electrolyte. While the PB model (point charges in a dielectric continuum water solvent) is crude, it does capture the basic feature of a double layer at a charged interface and the requirement of charge neutrality. One important result that can come from application of DFT to these problems is to quantify the magnitude of other physical effects (e.g. volume exclusion, dispersion, polarization, hydrogen bonding, etc) relative to the electrostatics. In addition, DFT provides direct access to a potential of mean force which may be used to perform simulations on proteins with an accurate but implicit solvent [24]. Since a large fraction of the computational effort in molecular simulation of biological systems involves simulation of the solvent, such a coupling could be tremendously useful.

Clearly, DFT is an attractive way to approach interfacial fluid problems in nanotechnology and biology. However, three-dimensional (3D) solutions will be required, and will even need to become rapid and robust enough to be incorporated inside of analysis or molecular simulation loops. The integral equations on which DFTs are based make 3D calculations challenging. In this paper, we will summarize our efforts to overcome the numerical challenges to solving DFTs for 3D systems on the scale of a protein solvation problem. We will show advances that have been made in analysis capabilities. We discuss a few functional approaches for both atomistic and polymer fluids from a computational perspective. We also briefly present several recent applications of our algorithms to demonstrate the feasibility of the methods.

2. Description of functionals

Before proceeding to a comparison of different numerical approaches, we summarize the fluid models we have studied to date. We note that extension of the algorithms to treat many other types of model fluids is straightforward. The description presented here is quite brief; a more detailed description of the DFT approach, the connection between DFT and correlation function hierarchies, sum rules and approximations can be found elsewhere [25].

2.1. Atomic fluids

Since DFTs are based on finding the extrema of the grand free energy functional, Ω, we first write this free energy as a perturbation expansion around a hard sphere system as

$$\Omega = F_{id} + F_{hs} + F_p - \sum_i \int \mathrm{d}\boldsymbol{r}\, \rho_i(\boldsymbol{r})[V_i(\boldsymbol{r}) - \mu_i], \tag{1}$$

where F_{id} is the ideal gas contribution to the Helmholtz free energy, F_{hs} and F_p are excess free energies of the hard sphere and perturbation (e.g. Lennard-Jones or Coulomb) contributions, V_i is the external field acting on species i and μ_i is the chemical potential of species i. In our work, the various terms are

$$F_{id} = kT \sum_i \int \mathrm{d}\boldsymbol{r}\, \rho_i(\boldsymbol{r})\{\ln(\Lambda_i^3 \rho_i(\boldsymbol{r})) - 1\}, \tag{2}$$

$$F_{hs} = kT \int \mathrm{d}\boldsymbol{r}\, \Phi(\{\bar{\rho}_\gamma(\boldsymbol{r})\}), \tag{3}$$

and

$$F_p = \tfrac{1}{2} \sum_i \sum_j \int \mathrm{d}\boldsymbol{r} \int \mathrm{d}\boldsymbol{r}'\, \rho_i(\boldsymbol{r})\rho_j(\boldsymbol{r}')u_{ij}(|\boldsymbol{r} - \boldsymbol{r}'|), \tag{4}$$

where Φ is a nonlocal free energy density, $\{\bar{\rho}_\gamma\}$ are the set of nonlocal densities and u_{ij} is a pairwise interaction potential. Note that equation (4) implies that the perturbation potential is treated in a strict mean field approximation where the perturbation contribution is independent of the integration between the hard sphere potential and the potential of interest [5].

In our work, we have applied the hard sphere free energy densities derived by Rosenfeld *et al* [26]. In this approach, the nonlocal densities are

$$\bar{\rho}_\gamma(\boldsymbol{r}) = \sum_i \int \mathrm{d}\boldsymbol{r}' \rho_i(\boldsymbol{r}') w_i^{(\gamma)}(|\boldsymbol{r} - \boldsymbol{r}'|), \tag{5}$$

where the $w^{(\gamma)}$ are a set (4 scalar and 2 vector) of weight functions based on the so-called fundamental measures (radius, surface area, volume, etc) of the fluid particles. Specifically, the weight functions are

$$\begin{aligned} w_i^{(3)}(r) &= \theta(r - R_i) \\ w_i^{(2)}(r) &= 4\pi R_i w^{(1)}(r) = 4\pi R_i^2 w_i^{(0)}(r) = \delta(r - R_i) \\ \boldsymbol{w}_i^{(V2)}(r) &= 4\pi R_i \boldsymbol{w}_i^{(V1)}(r) = (\boldsymbol{r}/r)\delta(r - R_i) \end{aligned} \tag{6}$$

where the bold type indicates a vector and R_i is the radius of species i. The hard sphere free energy density Φ is given by a sum of terms, $\Phi = \Phi_s + \Phi_v$ with

$$\Phi_s = -\bar{\rho}_0 \ln(1 - \bar{\rho}_3) + \frac{\bar{\rho}_1 \bar{\rho}_2}{1 - \bar{\rho}_3}, \tag{7}$$

$$\Phi_v = -\frac{\bar{\rho}_{V1} \bar{\rho}_{V2}}{1 - \bar{\rho}_3} + \frac{1}{24\pi(1 - \bar{\rho}_3)^2} \left(\bar{\rho}_2 - \frac{\bar{\rho}_{V2} \bar{\rho}_{V2}}{\bar{\rho}_2} \right)^3. \tag{8}$$

This functional correctly treats a crossover from a 3D to a 0D fluid [26]. Such a crossover occurs on crystallization when fluid particles are confined to a point in a 3D volume. We note that, in the ion channel problem discussed in section 4.1, such a crossover does occur.

The key equations to be solved for the case of atomic fluids are the Euler–Lagrange equations defined by the functional minimization of the grand potential

$$\begin{aligned} \frac{\delta \Omega}{\delta \rho_i(\boldsymbol{r})} = 0 = \ln \rho_i(\boldsymbol{r}) + V_i(\boldsymbol{r}) - \mu_i + \int \mathrm{d}\boldsymbol{r}' \sum_\gamma \frac{\partial \Phi}{\partial \bar{\rho}_\gamma} \frac{\delta \bar{\rho}_\gamma}{\delta \rho_i(\boldsymbol{r})} \\ + \sum_j \int \mathrm{d}\boldsymbol{r}' \, \rho_j(\boldsymbol{r}') u_{ij}(|\boldsymbol{r} - \boldsymbol{r}'|) \end{aligned} \tag{9}$$

where $\delta \bar{\rho}_\gamma / \delta \rho_i(\boldsymbol{r}) = w_i^{(\gamma)}(|\boldsymbol{r} - \boldsymbol{r}'|)$. Equation (9) will be discussed in more detail from a numerical point of view in section 3.

2.2. Polymeric fluids

In order to study inhomogeneous polymer fluids, we have also developed a polymer-DFT code [27]. This code is based on the molecular DFT of Chandler *et al* [6] and follows closely the approach of Donley *et al* [28]. Here we simply summarize the system of equations that must be solved for freely jointed chain polymers. The strategy is to solve the DFT for a system composed of a noninteracting gas of ideal chains in an *unknown field*, $U(\boldsymbol{r})$. The unknown field is the field required for the ideal system to ultimately have the same density distribution, $\rho(\boldsymbol{r})$, as the system of interest in the *known* external field, $V(\boldsymbol{r})$. For chains with multiple types of sites, the inhomogeneous monomer density profile is $\rho(\boldsymbol{r}) = \sum_\alpha \rho_\alpha(\boldsymbol{r})$, where $\rho_\alpha(\boldsymbol{r})$ is the density of site type α at $\boldsymbol{r}$. The unknown field is defined more precisely as a medium

induced potential on site type α, $U_\alpha(\boldsymbol{r})$. A functional minimization of the grand potential free energy, Ω, with respect to $\rho_\alpha(\boldsymbol{r})$ and $\mu - U_\alpha(\boldsymbol{r})$ leads to

$$U_\alpha(\boldsymbol{r}) = V_\alpha(\boldsymbol{r}) - \sum_\beta \int \mathrm{d}\boldsymbol{r}'\, c_{\alpha\beta}(\boldsymbol{r}-\boldsymbol{r}')(\rho_\beta(\boldsymbol{r}') - \rho_{b\beta}), \tag{10}$$

where $\rho_{b\beta}$ is the bulk density for site type β and $c_{\alpha\beta}$ is the direct correlation function (DCF) between the α and β site types in the bulk polymer fluid.

Donley *et al* focused on hard chain polymers at hard walls and computed DCFs from the polymer reference interaction site model (PRISM) liquid state theory [28]. We are interested in treating systems with attractions as well. PRISM theory is known to be more accurate in its treatment of hard chain systems [29a]. Therefore, we again use a perturbation scheme to include attractions. Specifically, we apply the random phase approximation

$$c_{\alpha\beta}(r) = \begin{cases} c^h_{\alpha\beta}(r) & \text{if } r \leqslant \sigma_{\alpha\beta} \\ -\beta u_{\alpha\beta}(r) & \text{otherwise} \end{cases} \tag{11}$$

where the hard chain DCF, c^h, is obtained from PRISM calculations and $\beta = 1/kT$.

Equation (10) has two unknown functions, the density distribution of interest, $\rho_\alpha(\boldsymbol{r})$, and the medium induced external field, $U_\alpha(\boldsymbol{r})$. Clearly the relationship between the two must be defined. For the ideal polymer gas, this relationship is

$$\rho_\alpha(\boldsymbol{r}) = \frac{V}{N}\rho_{bulk}\left\langle \delta(\boldsymbol{r}-\boldsymbol{r}_\alpha)\exp\left[-\beta\sum_\beta U_\beta(\boldsymbol{r}_\beta)\right]\right\rangle_S \tag{12}$$

where the brackets denote a configurational integral over all possible configurations, $\{\boldsymbol{r}\} = \{\boldsymbol{r_1}, \boldsymbol{r_2}, \ldots, \boldsymbol{r}_\mathrm{N}\}$, based on the configuration probability $S\{\boldsymbol{r}\}$. For a single random walk chain, the configurational probability can be written as

$$S(\boldsymbol{r_1}, \ldots, \boldsymbol{r}_N) = \prod_{s=2}^{N_s}[\omega_{12}(\boldsymbol{r}_s - \boldsymbol{r}_{s-1})], \tag{13}$$

where $\omega_{12}(\boldsymbol{r}-\boldsymbol{r}')$ is the probability of a bond of a certain length. In this work we will consider freely jointed chains with bond length σ, for which we have

$$\omega_{12}(\boldsymbol{r}) = \frac{1}{4\pi\sigma^2}\delta(|\boldsymbol{r}| - \sigma). \tag{14}$$

Given the definition for $S\{\boldsymbol{r}\}$ in equation (13), the density distribution for sites of type α on a linear chain becomes

$$\rho_\alpha(\boldsymbol{r}) = \frac{\rho_{b,\alpha}}{N_\alpha}\sum_{s=1}^{N_\alpha}\frac{G_s(\boldsymbol{r})G^{inv}_s(\boldsymbol{r})}{\exp[-\beta U_{\beta(s)}(\boldsymbol{r})]} \tag{15}$$

where the sum over s is a sum over all the segments of type α in a chain, $\beta(s)$ denotes the bead type associated with segment s and the G and G^{inv} are propagator functions that describe the configurational probability based on chain connectivity. The propagator functions in equation (15) obey the recursion relations

$$G_s(\boldsymbol{r}) = \exp[-\beta U_{\alpha(s)}(\boldsymbol{r})]\int \mathrm{d}\boldsymbol{r}'\,\omega_{12}(\boldsymbol{r}-\boldsymbol{r}')G_{s-1}(\boldsymbol{r}') \tag{16}$$

$$G^{inv}_s(\boldsymbol{r}) = \exp[-\beta U_{\alpha(s)}(\boldsymbol{r})]\int \mathrm{d}\boldsymbol{r}'\,\omega_{12}(\boldsymbol{r}-\boldsymbol{r}')G^i_{s+1}(\boldsymbol{r}') \tag{17}$$

for the $2\ldots N_s$ segments in the case of equation (16) and the $N_s - 1\ldots 1$ segments in the case of equation (17). They satisfy the 'initial' conditions $G_1(\boldsymbol{r}) = \exp[-\beta U_{\alpha(1)}(\boldsymbol{r})]$ and $G^{inv}_{N_s}(\boldsymbol{r}) = \exp[-\beta U_{\alpha(N_s)}(\boldsymbol{r})]$. Note that equation (15) reduces to the expected result, $\rho_\alpha(\boldsymbol{r}) = \rho_{b,\alpha}\exp[-\beta U_\alpha(\boldsymbol{r})]$, in the case of an atomic liquid. In our calculations, equations (10) and (15)–(17) make up the system of equations to be solved.

3. Numerical approaches

In this section, we summarize several numerical approaches we have taken to solve the systems of equations described above. In all cases we use a fully coupled Newton iteration to locate equilibrium solutions. The unknowns in the problem are represented by $\boldsymbol{x}$ (e.g. $\rho_i(\boldsymbol{r})$) and $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ is the system of equations to be solved. While we eventually discretized the unknown functions into a vector of unknown coefficients, and the functional equations into a set of algebraic equations, the equations will be presented in functional form. Newton's method is posed as the solution to

$$\boldsymbol{J}\boldsymbol{d} = -\boldsymbol{f} \tag{18}$$

where $\boldsymbol{d} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ is the difference between successive iterations of the unknowns and the Jacobian matrix is given by

$$J_{ij}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta f_i(\boldsymbol{r})}{\delta x_j(\boldsymbol{r}')}. \tag{19}$$

Here subscripts i and j represent different functional equations and unknowns, such as when solving for multiple components. The linear problem posed by equation (18) is solved for approximately using GMRES iterative solvers with a variety of preconditioners. We will discuss the details of the derivation of the residual equations, the algorithms for solving them, observations about preconditioning and strategies for parallelization below. We note that, while we are seeking solutions to the nonlinear equations $\boldsymbol{f}(\boldsymbol{x}) = 0$, multiple solutions may exist. Away from a phase transition, only one of the solutions will be thermodynamically stable; other solutions will be either unstable or metastable. Multiple thermodynamically stable solutions exist at phase transitions. Since the phase diagrams for inhomogeneous fluids are particularly rich, and investigating those phase diagrams is a central application of DFTs, we have implemented automated continuation and phase transition tracking algorithms. These are described further in section 3.4.3.

3.1. Real space numerical approaches

One approach we have implemented is based on forming integration stencils for the nonlocal densities (atomic fluids) or propagator functions (polymeric fluids) defined above. The computational strategy is to form the Jacobian matrix based on the real space integration stencils [8]. The precise nature of the Jacobian matrix depends on how one chooses to formulate the systems of equations. Here we briefly present implicit and explicit nonlocal functional approaches for both atomic and polymer fluid models based on this real space approach.

3.1.1. Hard sphere fluids with implicit nonlocal densities. The residual equation for hard sphere fluids was given in equation (9). From a numerical point of view the critical question is how to formulate the Jacobian. Taking the functional derivative in equation (19) yields

$$J_{ij}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta_{ij}(\boldsymbol{r} - \boldsymbol{r}')}{\rho(\boldsymbol{r}')} + \int \mathrm{d}\boldsymbol{r}'' \sum_{\gamma} \sum_{\epsilon} \frac{\partial^2 \Phi}{\partial \bar{\rho}_\gamma \partial \bar{\rho}_\epsilon} w_i^{(\gamma)}(|\boldsymbol{r} - \boldsymbol{r}''|) w_j^{(\epsilon)}(|\boldsymbol{r}' - \boldsymbol{r}''|) \tag{20}$$

where $\delta_{ij}(\boldsymbol{r} - \boldsymbol{r}') = 1$ only when $i = j$ and $\boldsymbol{r} = \boldsymbol{r}'$; otherwise $\delta_{ij}(\boldsymbol{r} - \boldsymbol{r}') = 0$. The physical meaning of the product of weight functions in the above equation is clear. This product will be nonzero only if two weight function stencils (centred at $\boldsymbol{r}$ and $\boldsymbol{r}'$) are overlapping. Each entry in the Jacobian contains an integral of the intersecting volumes of all possible products

of weight functions. Performing these integrals is time consuming because it is an operation that is $O(N_s^2)$, where N_s is the number of points in a given integration stencil.

In addition to computational complexity, the memory required to store the Jacobian is another important factor in the feasibility of large three-dimensional calculations using these accurate DFTs. In this case, there are nonzero Jacobian entries for every node within two stencil lengths (this is the particle radius, R_i, in the Rosenfeld functional) of the row position, $\boldsymbol{r}$. Thus the required memory is proportional to the number of nonzeros in the matrix, M, where

$$M = \frac{4\pi}{3}\left(\frac{2R}{h}\right)^3 {N_c}^2 N, \tag{21}$$

N_c is the number of components in the system and N is the number of mesh points. We note that the role of the Jacobian is really to direct the Newton updates in the direction of the solution. Therefore, it is often possible to approximate the Jacobian by coarsening the integrals (e.g. compute the residual integrals on a mesh of $h = 0.1\sigma$ while the Jacobians are computed on a mesh of $h = 0.2\sigma$). This coarsening can be set independently in different regions, and can often reduce the size of the matrix, and consequently the solve time, by a factor of 10 or more [8]. Of course the effect of Jacobian coarsening on numerical stability is highly problem dependent.

3.1.2. Hard sphere fluids with explicit nonlocal densities. Another algorithm with lower computational complexity can also be formulated in real space by treating the nonlocal densities, $\bar{\rho}_\gamma(\boldsymbol{r})$, as an explicit part of the matrix problem. In this case, the matrix problem involves two residual equations that must be solved simultaneously. The first, $\boldsymbol{f}^{(1)}$, is given by equation (9) while the second, $\boldsymbol{f}^{(2)}$, is equation (5). In this formulation, there is a $\boldsymbol{f}^{(1)}$ vector for each species and a $\boldsymbol{f}^{(2)}$ vector for each weight function. The Rosenfeld functionals have 6 weight functions (4 scalar, 2 vector) in one dimension and 10 weight functions (4 scalar, 6 vector) in three dimensions. Thus a three-dimensional three-component system where the species have different sizes has 13 unknowns per node. In this case, the matrix problem is

$$\begin{bmatrix} \frac{\delta \boldsymbol{f}_i^{(1)}}{\delta \rho_j} & \frac{\delta \boldsymbol{f}_i^{(1)}}{\delta \bar{\rho}_\epsilon} \\ \frac{\delta \boldsymbol{f}_\gamma^{(2)}}{\delta \rho_j} & \frac{\delta \boldsymbol{f}_\gamma^{(2)}}{\delta \bar{\rho}_\epsilon} \end{bmatrix} \begin{bmatrix} \Delta\rho_i \\ \Delta\bar{\rho}_\gamma \end{bmatrix} = -\begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \end{bmatrix}, \tag{22}$$

and the specific Jacobian entries are

$$J_{ij}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta \boldsymbol{f}_i^{(1)}}{\delta \rho_j} = \frac{\delta_{ij}(\boldsymbol{r} - \boldsymbol{r}')}{\rho(\boldsymbol{r}')} \tag{23}$$

$$J_{i\epsilon}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta \boldsymbol{f}_i^{(1)}}{\delta \bar{\rho}_\epsilon} = \sum_\gamma \sum_\epsilon \frac{\partial^2 \Phi}{\partial \bar{\rho}_\gamma(\boldsymbol{r}) \partial \bar{\rho}_\epsilon(\mathbf{r}')} w_{i\gamma}(\boldsymbol{r} - \boldsymbol{r}') \tag{24}$$

$$J_{\gamma j}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta \boldsymbol{f}_\gamma^{(2)}}{\delta \rho_j} = w_{j\gamma}(\boldsymbol{r} - \boldsymbol{r}') \tag{25}$$

$$J_{\gamma\epsilon}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\delta \boldsymbol{f}_\gamma^{(2)}}{\delta \bar{\rho}_\epsilon} = \delta_{\gamma\epsilon}(\boldsymbol{r} - \boldsymbol{r}'). \tag{26}$$

In this case, the computational complexity involved with forming the Jacobian is much reduced from the previous section because there are no integrals to perform. More specifically, loading one row of the matrix is an $O(N_s)$ operation. The result is that the overall computational scaling is now dominated by the GMRES solution rather than the matrix fill.

The memory required for this explicit nonlocal density approach may be greater or less than the memory needed for the implicit approach of the previous section. In this case, there

are nonzero Jacobian entries for every node within one stencil length (this is the particle radius, R_i, in the Rosenfeld functional) of the row position, $\boldsymbol{r}$. Thus the number of nonzeros in the matrix is now approximately

$$M_{ex} = \frac{4\pi}{3}\left(\frac{R}{h}\right)^3 2N_w N_c N, \tag{27}$$

and the ratio between this method and that of the previous section is $N_w/(4N_c)$ Since the two different formulations lead to different Jacobian matrices for Newton's method, comparisons of the convergence behaviour and compute times are more complicated and problem dependent. Jacobian coarsening strategies can be applied to this system of equations as well; however, these strategies are generally less successful than for the implicit nonlocal density approach.

In figure 1 (curves labelled RS1 and RS2) we show computational scaling results for both the implicit and explicit nonlocal density approaches for a 3D calculation of a hard sphere fluid of density $\rho\sigma^3 = 0.8785$ solved in a periodic (cubic) porous matrix where the lattice spacing was 3.2σ, and both the fixed particles of the matrix and the fluid particles had size σ. The figure shows the time per Newton iteration and demonstrates that the implicit method scales as $\mathrm{O}(N^{2.2})$ while the explicit method scales as $\mathrm{O}(N^{1.6})$.

3.1.3. Hard chain polymer fluids with explicit propagator functions. We now turn briefly to the computational problem encountered by the polymer systems described above. Again we restrict our initial discussion to hard chain polymers. We have implemented an explicit propagator function algorithm similar to the explicit nonlocal density approach described above. In this method, the unknowns we solve for are $\{\rho_i(\boldsymbol{r})\}$, $\{U_i(\boldsymbol{r})\}$, $\{G_s(\boldsymbol{r})\}$ and $\{G_s^{inv}(\boldsymbol{r})\}$. So for a 10-bead polymer with two segment types, there will be 24 unknowns per mesh point. The four residual equations we solve were previously enumerated. Specifically, $\boldsymbol{f}_\alpha^{(1)}(\boldsymbol{r})$, $\boldsymbol{f}_\alpha^{(2)}(\boldsymbol{r})$, $\boldsymbol{f}_s^{(3)}(\boldsymbol{r})$ and $\boldsymbol{f}_s^{(4)}(\boldsymbol{r})$ were given by equations (10), (15), (16), and (17), respectively. The associated matrix problem is

$$\begin{bmatrix} \frac{\delta \boldsymbol{f}_\alpha^{(1)}}{\delta\rho_\beta} & \boldsymbol{I} & 0 & 0 \\ \boldsymbol{I} & \frac{\delta \boldsymbol{f}_\alpha^{(2)}}{\delta U_\alpha} & \frac{\delta \boldsymbol{f}_\alpha^{(2)}}{\delta G_s} & \frac{\delta \boldsymbol{f}_\alpha^{(2)}}{\delta G_s^{inv}} \\ 0 & \frac{\delta \boldsymbol{f}_s^{(3)}}{\delta U_\alpha} & \frac{\delta \boldsymbol{f}_s^{(3)}}{\delta G_s} & 0 \\ s0 & \frac{\delta \boldsymbol{f}_s^{(4)}}{\delta U_\alpha} & 0 & \frac{\delta \boldsymbol{f}_s^{(4)}}{\delta G_s^{inv}} \end{bmatrix} \begin{bmatrix} \Delta\rho_\alpha \\ \Delta U_\alpha \\ \Delta G_s \\ \Delta G_s^{inv} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \\ \boldsymbol{f}^{(3)} \\ \boldsymbol{f}^{(4)} \end{bmatrix}, \tag{28}$$

where the $\boldsymbol{I}$ indicates blocks of the matrix that are just the identity matrix. The remaining entries are straightforward to enumerate, and we simply note that, as in the previous section, there are no integrals in the Jacobian. If one attempts to formulate this system of equations using implicit propagator functions, N-dimensional integrals will appear in the Jacobian where N is the chain length. These integrals are due to the recursive nature of the propagator equations and will result in an intractable algorithm.

The approach presented here for polymer fluids can clearly be applied in the limit of a single monomer as well. From a numerical perspective such an approach can be very fast because the complex functionals of the previous sections are completely removed from the problem. Instead, the DCF (taken either from a bulk molecular simulation or an integral equation theory with some closure) captures the effects of the hard core. This approach has, in fact, been recently demonstrated by Zhou [30]. One caveat here is that integral equation theories (for both atomic and polymeric fluids) are often not thermodynamically self-consistent [29a], so that using DCFs calculated from these theories will necessarily introduce some inaccuracies into the DFT.

3.2. A matrix-free approach with FFT convolutions

Another approach we have investigated with respect to solving equation (9) relies on fast Fourier transforms, FFTs, to compute convolutions [9]. We have furthermore implemented this scheme as a *matrix-free* GMRES approach. In this case, the entire Jacobian is never computed and stored. Rather, we seek only to compute the operation of the Jacobian on a field, $\boldsymbol{x}$, which will come from the iterative scheme of interest. Specifically, we seek to compute $y(\boldsymbol{r})$ where

$$y_i(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{r}' \sum_j J_{ij}(\boldsymbol{r}, \boldsymbol{r}')x_j(\boldsymbol{r}'). \tag{29}$$

Consider the Jacobian defined earlier in equation (20). The hard sphere contribution to the operational Jacobian will be

$$y_i(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{r}' \sum_j \int \mathrm{d}\boldsymbol{r}'' \sum_\gamma \sum_\epsilon \frac{\partial^2 \Phi}{\partial \bar{\rho}_\gamma \bar{\rho}_\epsilon} w_i^{(\gamma)}(|\boldsymbol{r} - \boldsymbol{r}''|) w_j^{(\epsilon)}(|\boldsymbol{r}' - \boldsymbol{r}''|)x_j(\boldsymbol{r}'). \tag{30}$$

Efficient computation of equation (30) relies on noticing that it can be evaluated via two successive sets of convolutions. To this end, we define two intermediate quantities. The first is a convolution of the field x with the weight function w:

$$t^{(\epsilon)}(\boldsymbol{r}'') = \int \mathrm{d}\boldsymbol{r} \sum_j w_j^{(\epsilon)}(\boldsymbol{r}'' - \boldsymbol{r}')x_j(\boldsymbol{r}'). \tag{31}$$

The second multiplies t with the second derivatives of Φ:

$$z^{(\gamma)}(\boldsymbol{r}'') = \sum_\epsilon \left.\frac{\partial^2 \Phi}{\partial \bar{\rho}_\gamma \partial \bar{\rho}_\epsilon}\right|_{r''} t^{(\epsilon)}(\boldsymbol{r}''). \tag{32}$$

Finally, the contribution to $y(\boldsymbol{r})$ from the hard sphere part of the free energy functional is

$$y(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{r}'' \sum_\gamma w_i^{(\gamma)}(\boldsymbol{r}'' - \boldsymbol{r}) z^{(\gamma)}(\boldsymbol{r}''). \tag{33}$$

Equation (30) can now be computed with two successive convolutions. The first set of convolutions is done for each weight function (equation (31)) and the second set of convolutions is done for each component (equation (33)). Both of these convolutions can be computed with FFTs, and so the total cost of the computation scales as the sum of the number of weight functions and the number of components.

Since the matrix is never stored in this scheme the memory requirements are much smaller than the matrix-based algorithms of the previous sections. The total memory needed will simply be that necessary to store the residuals and perform the FFTs as described above. It will thus scale as N, the number of mesh points in the problem of interest. Figure 1 compares the scaling of this matrix-free Fourier space algorithm (curve labelled FS) to the algorithms of the previous sections. Clearly, for a single processor computation, this approach has a significant advantage and allows computation of problems with $O(10^6)$ unknowns in a modest amount of computational time.

The tradeoffs of this approach are that one no longer has access to matrix-based preconditioning methods. Rather (as we discuss further below), physics based preconditioning will be required, and it is likely that these preconditioning schemes will require tuning when numerically stiff problems are encountered. A second tradeoff is that the FFT method for doing convolutions has to be carefully adapted for non-periodic systems. Finally, we note that a similar algorithm can clearly be developed to solve the polymer-DFT system of equations, and implementation of those equations is currently underway.

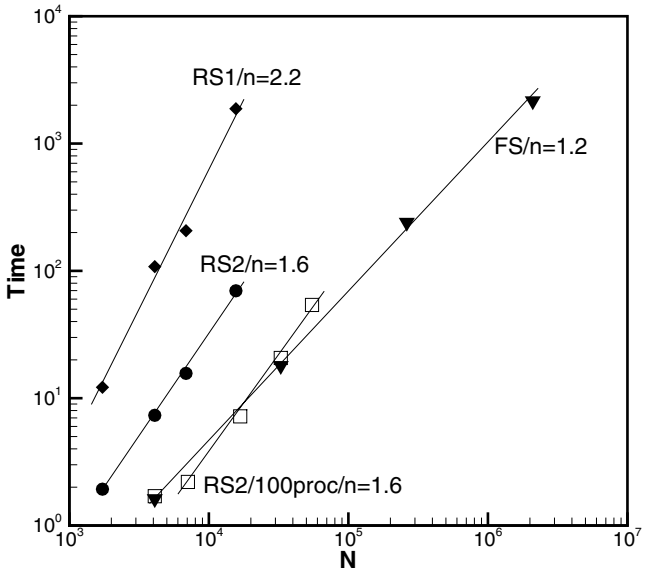

Figure 1. The time per Newton iteration as a function of the number of unknowns for the real space and Fourier space algorithms described above. The various curves show scaling results for the algorithms in sections 3.1.1 (RS1), 3.1.2 (RS2) and 3.2 (FS). In addition a parallel timing for the RS2 algorithm is shown by open symbols. This parallel timing curve was based on 100 processors of the CPlant commodity cluster at Sandia National Labs. Note that the single processor timings were done on faster processors (by about a factor of 4) than the processors in the CPlant cluster. The scaling exponent, n, is given for each curve where the time was taken to scale as $O(N^n)$. Note that the number of unknowns, N, is increased by decreasing the mesh spacing for a constant domain size. All of the algorithms exhibit linear scaling when the domain size is increased at constant mesh spacing.

3.3. Numerics of perturbation potentials

Finally, we briefly consider the perturbation potentials. The two types of perturbations we consider are based on Lennard-Jones and Coulomb interactions.

3.3.1. Lennard-Jones perturbations. We typically apply the perturbation approximation of Weeks *et al* [32] when we treat Lennard-Jones fluids. In that approximation, the repulsive part of the potential is replaced. More specifically, for $r < r_{min}$ we set $u_{ij} = u_{ij}(r_{min})$, where r_{min} is the position where the potential minimum is found. We also typically cut and shift the potential with cutoff distances ranging from $r_c = 2.5\sigma$ to $r_c = 10\sigma$. The perturbation contribution to the Jacobian in this case is simply

$$J_{ij}(\boldsymbol{r}, \boldsymbol{r}') = u_{ij}(|\boldsymbol{r} - \boldsymbol{r}'|), \tag{34}$$

and the matrix sparsity now depends on the range of the interaction potential. For small r_c (e.g. 2.5σ), the cost of the Jacobian is tolerable although large with a real space approach. Specifically the memory required to store the Jacobian will be

$$M_P = \frac{4\pi}{3}\left(\frac{r_c}{h}\right)^3 {N_c}^2 N. \tag{35}$$

Thus $M_P/M = (r_c/\sigma)^3$. Once again in the matrix-free approach of section 3.2, the Jacobian is not stored and so essentially there are no additional memory requirements for attractions.

3.3.2. Coulomb perturbations. In the case of Coulomb fluids, the memory issue could be particularly severe since these interactions are of infinite extent. However, this problem can be finessed by introducing the electrostatic potential, ϕ, as an additional field in the problem. Specifically, the Coulomb contribution to the Euler–Lagrange equation (equation (9)) is

$$z_i e\left(V^c + \sum_j \int \mathrm{d}\boldsymbol{r}' \frac{\rho_j(\boldsymbol{r}')z_j e}{|\boldsymbol{r}-\boldsymbol{r}'|}\right) = z_j e\phi(\boldsymbol{r}) \tag{36}$$

where V^c is the Coulombic part of the external field. Then, in addition to solving equation (9), we also solve Poisson's equation:

$$\nabla^2\phi(\boldsymbol{r}) + \sum_i z_i e\rho_i(\boldsymbol{r}) = 0. \tag{37}$$

This differential equation is very short range compared even to the hard sphere calculations and results in a very sparse matrix. So, the memory requirements of solving Poisson's equation are negligible.

Finally, we note that the strict mean field approximation we have applied here is quite crude. While it will capture gross features of the system (e.g. charge neutrality and the exponential decay of double layers for Coulomb systems), it will not capture more subtle features, such as attractions between charged surfaces in multivalent electrolytes [33] or liquid–vapour equilibria in molten salts [34]. In studying those cases other groups have suggested several corrections to the basic theory above. These corrections add additional terms composed of integrals over either DCFs [33] or total correlation functions [34] to the Euler–Lagrange equation. It is clear that for biology applications (e.g. to model selectivity in ion channels), an accurate description of ions in an electrolyte will be required.

3.4. More strategies for fast DFT calculations

In the previous sections, we have detailed various algorithmic approaches we have developed to solve the DFT equations accurately, rapidly, and with good convergence. There are several additional strategies for improved performance that we have developed to approach the goal of being able to solve and analyse large two- and three-dimensional problems. We summarize some of these strategies here.

3.4.1. Parallelization. The first approach is to apply parallel computations to perform 3D calculations. We have implemented a spatial decomposition of the mesh in the real space algorithms across many (up to 1000) processors [8]. In these calculations, each processor owns different rows of the residual vector and Jacobian matrix corresponding to the mesh points (and corresponding unknowns) in its domain. One of the most complicated parts of parallelization is the identification and communication of ghost unknowns. These are unknowns within an integration stencil length of those unknowns owned by a processor, which must be communicated before each matrix assembly and matrix–vector multiply.

In the case of Fourier space algorithms, the majority of the computational cost comes in performing the convolutions with the FFTs. Therefore, parallel speed-up will be possible with parallelization of the FFT operation. It is known there is initially a decrease in efficiency of the FFT operation with increasing processors for small numbers of processors, but we expect that even a modest sized cluster (e.g. 10 processors) will be sufficient to see substantial performance increase on parallelization.

3.4.2. Preconditioning. The GMRES iterative solutions of the linear subproblem of Newton's method can be greatly accelerated by preconditioning of the linear system. An in-depth discussion of preconditioners is beyond the scope of this paper; however, we briefly mention the various strategies we employ. We note that the difference between good and poor preconditioning strategies can account for an order of magnitude or more in the time to solution, and in some cases it makes the difference between convergence and no convergence of the method. In a more general sense, optimal preconditioning for these kinds of DFTs is an open question that we will be pursuing in the future.

The real space algorithms outlined above have been solved with the parallel iterative solver package Aztec. This package contains a variety of matrix-based preconditioners including Jacobi and domain-decomposition methods such as ilut, with variable levels of overlap and fill-in. For the systems of equations described above the type of preconditioner needed depends on the physics of the problem of interest. Generally, hard sphere fluids can be solved easily with no preconditioning. Lennard-Jones systems often require modest (Jacobi or ilut with no overlap) preconditioning while polymer systems often require ilut preconditioning with two levels of overlap between processors and considerable amounts of fill-in.

Clearly, none of these matrix-based preconditioners are available to the matrix-free approach outlined in section 3.2. In this case, we have implemented a simple physics-based preconditioner that works well for hard sphere and Lennard-Jones fluids [9]. Preconditioning has not yet been a limiting factor, but we anticipate that this could be an issue in the application of the matrix-free approach.

3.4.3. Continuation and phase transition tracking algorithms. While the time to compute an individual DFT solution is critically important, often one seeks not only to solve a single case, but rather to solve the equations many times in order to explore parameter space and build phase diagrams. To study the phase behaviour of confined fluids, we need to be able to track whole solution branches (e.g. adsorption isotherms) as a function of the system parameters (e.g. bulk density, ρ). This task is complicated by the fact that the system exhibits multiplicity and hysteresis. Therefore, we have implemented algorithms for tracking solution branches and phase transitions, using methods and approaches from the bifurcation analysis community. We briefly describe these algorithms here.

The pseudo arclength continuation algorithm of Keller [35] was developed for tracking solution branches with hysteresis. We have used the implementation in the LOCA software library [36]. This algorithm alleviates the singularity seen at spinodal points by augmenting the system of equations with an arclength equation and freeing ρ as an unknown. The augmented system is then described by

$$\boldsymbol{f}(\boldsymbol{x}(s), \rho(s)) = 0 \tag{38}$$

$$g(\boldsymbol{x}(s), \rho(s), s) = 0, \tag{39}$$

where $\boldsymbol{f}$ is the same residual presented in previous sections and g is an arclength equation. Here, both $\boldsymbol{x}$ and ρ are parametrized as functions of s, which can be defined as a distance down the solution branch. It is convenient to use a linearized form of a distance formula. Given a converged solution $\boldsymbol{x}_{i-1}$ at parameter ρ_{i-1} and step size Δs, the next step must satisfy

$$g(\boldsymbol{x}(s), \rho(s), s) = (\boldsymbol{x}_i - \boldsymbol{x}_{i-1}) \cdot \frac{\partial \boldsymbol{x}_{i-1}}{\partial s} + (\rho_i - \rho_{i-1}) \frac{\partial \rho_{i-1}}{\partial s} - \Delta s = 0. \tag{40}$$

This algorithm seeks steady state solutions at steps of arclength Δs rather than of the parameter. The parameter step $\rho_i - \rho_{i-1}$ is free to decrease in regions of steep response and even turn negative when tracing out hysteresis.

One iteration of the Newton method for the augmented system can be expressed in matrix form as

$$\begin{bmatrix} \boldsymbol{J} & \frac{\partial \boldsymbol{f}}{\partial \rho} \\ \left(\frac{\partial \boldsymbol{x}}{\partial s}\right)^T & \frac{\partial \rho}{\partial s} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \rho \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f} \\ g \end{bmatrix}, \tag{41}$$

where $\Delta \boldsymbol{x}$ and $\Delta \rho$ are updates to $\boldsymbol{x}_i$ and ρ_i. Convergence is determined by the norm of the updates being below a set tolerance. Then a new step size Δs is chosen and the next solution at a new parameter value is computed. We have found that, even with the arclength continuation algorithm, sophisticated step size control is needed to robustly follow solution branches in regions with numerous phase transitions [37].

To find a single phase transition, the arclength continuation algorithm is a powerful technique. A transition where two solution branches cross can be detected when the free energy Ω is plotted versus the parameter. However, it would be very time consuming to manually construct a phase diagram this way. We have therefore developed a phase transition tracking algorithm to compute phase diagrams efficiently and automatically [37]. Specifically, our algorithm uses Newton's method to converge to a single parameter (e.g. the bulk density) value and two solution vectors which have equal free energies. The phase coexistence curves can then be tracked by stepping in a second parameter (e.g. temperature). We therefore repeatedly solve the following set of $2N + 1$ equations at a series of fixed temperatures:

$$\boldsymbol{f}(\boldsymbol{x}_1, \rho_b) = \boldsymbol{0} \tag{42}$$

$$\boldsymbol{f}(\boldsymbol{x}_2, \rho_b) = \boldsymbol{0} \tag{43}$$

$$G = \Omega(\boldsymbol{x}_1, \rho_b) - \Omega(\boldsymbol{x}_2, \rho_b) = 0. \tag{44}$$

Here the $2N + 1$ unknowns are ρ_b, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The final equation $G = 0$ requires that the two profiles have equal free energies, Ω.

An iteration of a full Newton method for this system has the form

$$\begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{0} & \frac{\partial \boldsymbol{f}_1}{\partial \rho} \\ \boldsymbol{0} & \boldsymbol{J}_2 & \frac{\partial \boldsymbol{f}_2}{\partial \rho} \\ \frac{\partial \Omega_1}{\partial \boldsymbol{x}_1} & -\frac{\partial \Omega_2}{\partial \boldsymbol{x}_2} & \frac{\partial G}{\partial \rho} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_1 \\ \Delta \boldsymbol{x}_2 \\ \Delta \rho \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ G \end{bmatrix}. \tag{45}$$

Here the subscript k on the variables $\boldsymbol{f}$, $\boldsymbol{J}$ and Ω represent evaluation with solution vector $\boldsymbol{x}_k$ ($k = 1$ or 2). The algorithm is started using initial guesses near a phase transition as detected from a single arclength continuation run. At each subsequent temperature, the solutions at the previous temperature are used as initial guesses to the Newton iteration. A triple-point tracking algorithm could similarly be constructed as a system of $3N + 2$ unknowns.

The matrix solutions for both the arclength continuation and phase transition tracking are reformulated using bordering algorithms, as detailed elsewhere [36, 37]. With this approach the augmented matrices in equations (41) and (45) never need to be constructed or solved. Instead, the updates are calculated using repeated solutions of the $N \times N$ Jacobian matrix. It is therefore very simple to modify a code that uses Newton's method to also use the analysis algorithms.

4. Applications

4.1. Solvation of and transport through gramicidin A

In this section we briefly discuss recent calculations we have performed on a simple pore forming membrane spanning polypeptide gramicidin A [38]. This system is a well studied model system that is similar in many respects to more complex ion channels [39]. It is important

to note that experiments can be done to measure the current through single isolated channels as a function of the applied voltage (or, more generally, electrochemical potential gradient) across the membrane. These patch clamp experiments are done routinely, and they demonstrate that the open channel current measured through a given channel is constant. Thus it is reasonable to model this process as a steady state diffusion process. As was noted in the introduction, steady state diffusion models of ion channels at the PB electrolyte level of approximation have been pursued extensively in the biophysics community. This approach is known as the Poisson–Nernst–Plank (PNP) theory [23].

We have begun to investigate the application of DFTs to ion channels in order to increase the rigour in the underlying fluid model. As a first approximation, we have modelled the transport through the pores as a steady state diffusion process similar to the route pursued in PNP theory. More specifically, we allow the electrochemical potential μ to vary spatially, $\mu(\boldsymbol{r})$, and its distribution is solved by coupling steady state flux equations,

$$\nabla \cdot J_i = 0 \tag{46}$$

to the Euler–Lagrange equations discussed previously (equation (9)). To date, we have taken a very simple species-independent expression for the diffusive flux:

$$J_i = -D_i \rho_i(\boldsymbol{r}) \nabla \mu_i(\boldsymbol{r}). \tag{47}$$

We have previously shown that this simple transport-DFT agrees well with grand canonical molecular dynamics simulations of diffusion through a simple one-dimensional membrane [40].

While there is not sufficient space here to present a complete summary of our gramicidin calculations, we show slices of two three-dimensional density distributions for a case where the fluid is at equilibrium in figure 2. The slices are taken down the centre of the channel. The gramicidin pore was formed by taking the atomic coordinates of its atoms from the 1MAG structure in the protein data bank. The partial charge and size of each atom was based on the Charm-22 force field. This polypeptide is net neutral, but the charge distribution is such that there is a net negative charge at the ends of the pore and a net positive charge in the centre of the pore. These calculations were done on 128 processors of the CPlant commodity cluster at Sandia Labs. Each three-dimensional solution of a three-component electrolyte that included Rosenfeld volume exclusions and Coulomb interactions took approximately 2 h with the real space algorithms (with a mesh spacing $\sigma/4$) described in section 3.1.2.

Figure 2 compares the density profiles from a PB treatment of the electrolyte with a finite sized electrolyte. The solvent accessible pore volume is smaller with the finite sized particle model, and the difference is significant because the pore is small enough that the fluid becomes a one-dimensional (beads on a string) fluid in this confined space. When liquid-like solvent densities (e.g. $\rho\sigma^3 = 0.7$) are considered, the fluid exhibits significant packing in the pore and can begin to resemble a zero-dimensional fluid.

One way to assess the mechanisms for transport in ion channels is to assume that the transport proceeds via an energy barrier hopping mechanism. The fact that the fluid in gramicidin A is a one-dimensional fluid means that the reaction coordinate for this transition state approach is particularly simple. We have computed the potential of mean force experienced by an ion as it moves through the channel (down the z axis) as

$$W_i(z) = \int [\mu(z; A) - \mu_r - kT \ln(\rho_i(z, A)/\rho_{i,bulk})] \, \mathrm{d}A \tag{48}$$

where μ_r is a convenient reference value for the electrochemical potential and the integral is taken over areas perpendicular to the z axis inclusive of all the accessible pore volume.

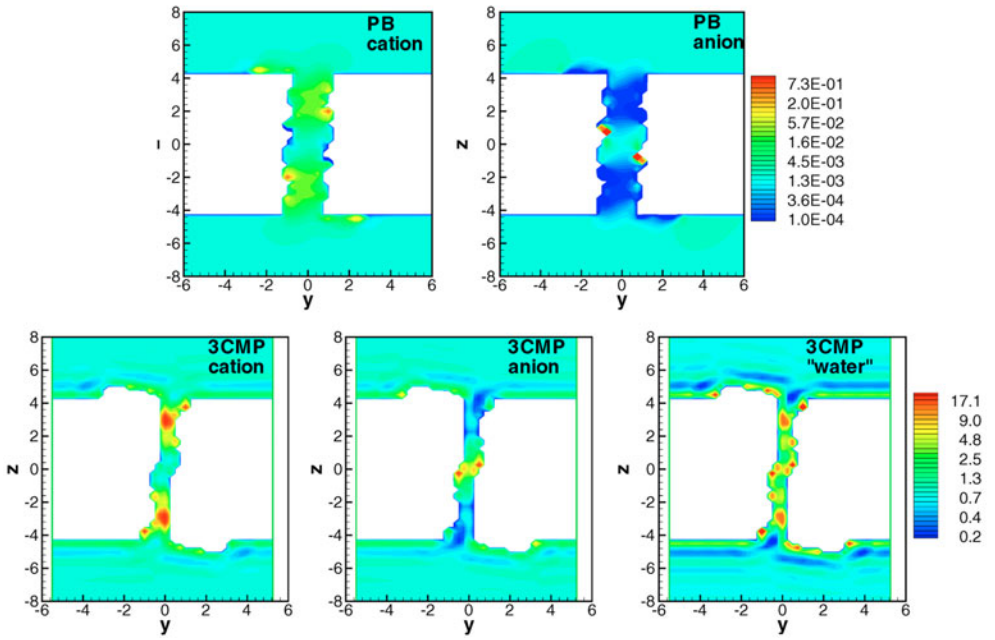

Figure 2. The density ($\rho\sigma^3$) distribution in one slice through the gramicidin A pore. The y and z distances are given in σ units where we took $\sigma = 0.3$ nm. The top two figures show the cation and anion density distributions using a PB electrolyte. The bottom three figures show the cation, anion and neutral solvent in the three-component (3CMP) model. All the ion figures are based on the density scale on the top. The solvent for the 3CMP model uses the density scale shown on the bottom.

In figure 3 we show the potential of mean force for a monovalent cation as it traverses the gramicidin A channel. In these calculations there is a drop in applied electric potential across the membrane ranging from 0 to 150 mV. This plot shows two deep free energy wells at either end of the channel that can be attributed to the charge distribution on the protein. Recall that gramicidin has a net negative charge (and hence a low free energy for a mobile cation) near the ends of the pore. The fine details (e.g. oscillations on a period of σ) can be attributed to volume exclusion based packing effects. The inhomogeneous boundary conditions are responsible for the linear bias on which both Coulomb and packing effect are superimposed.

The calculations we have performed to date on gramicidin were done with a very crude restricted primitive model electrolyte where all species have the same particle size. Furthermore, the calculations shown above did not include either second-order DCF or perturbation corrections that other groups have discussed [33, 34]. It seems apparent that there is very broad scope for the application of DFT to problems in biology, but they may hinge on accurate treatment of electrolytes.

4.2. Capillary condensation in disordered porous media

DFT models can effectively simulate adsorption of confined fluids. We present here recent calculations on how disorder affects capillary condensation in a porous material [41]. Adsorption in disordered porous materials has received attention lately in lattice based DFT studies on large three-dimensional systems [3]. These lattice approaches are quite coarse

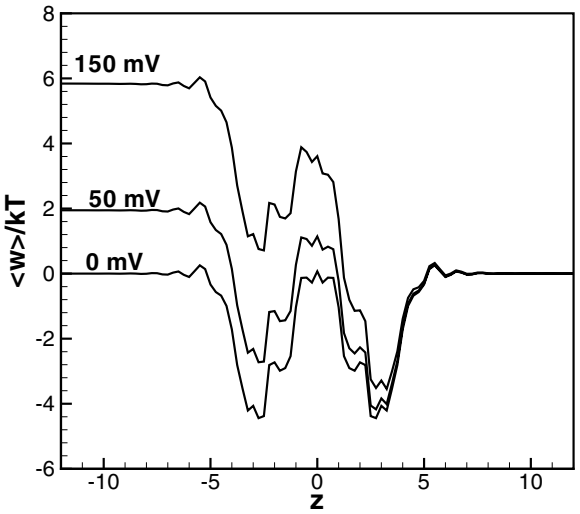

Figure 3. The average potential of mean force for a cation in a 1 M 3CMP electrolyte moving through gramicidin A. The three curves show the PMF for different values of an applied electrostatic potential across the membrane.

grained, and so we wanted to study systems with the more detailed functionals presented here in the limit of just a small degree of disorder in the porous material. To that end, we have performed extensive calculations on two basic model systems. In both cases, the porous material is composed of parallel cylindrical fibres (thus enabling two-dimensional solutions), but those fibres are randomly placed in a computational domain with periodic boundary conditions. One system is composed of 4 cylinders and the other has 16 cylinders. The interactions between the fibres and the fluid are found by integrating a 12–6 Lennard-Jones potential over the geometry of the fibre for each mesh point in the fluid. The fluid is also assumed to interact via a 12–6 Lennard-Jones potential. Fluid–fibre interactions were stronger than fluid–fluid interactions by a factor of 5. We show here a few results from our analysis which were enabled by the continuation and phase transition tracking algorithms [37].

Adsorption isotherms were computed with arclength continuation for this porous material, and two examples are shown in figure 4. Each isotherm is computed automatically with little or no user intervention. Notice that there is a complex path from the gas-like solutions to partially filled states to the complete capillary condensed state at high densities. There are many metastable (and unstable) solution branches in the vicinity of the capillary condensation transition. As the degree of disorder increases (more fibres), the complexity of the adsorption isotherm also increases. In both cases, capillary condensation occurs through a progression of stable partially filled states. In the 16 cylinder case shown here, there are two stable partially filled states in addition to the fully filled liquid phase (at high densities) and the near-surface monolayer coverage state (at low densities). The tie lines in figure 4 show the three transitions that together make up capillary condensation for this system. The 4 cylinder case has one stable partially filled state. In figure 5, we show three coexisting solutions at $kT/\epsilon = 0.7$. The middle figure shows the partially filled state where there is liquid bridge forming between cylinders 1 and 4 through the periodic boundary.

Finally, a phase diagram in temperature–adsorption space for the 4 cylinder case was generated with the phase transition tracking algorithms. It is shown in figure 6. The partially filled intermediate phase is found in the V-shaped region around $\Gamma/L = 34$ and $kT/\epsilon = 0.8$

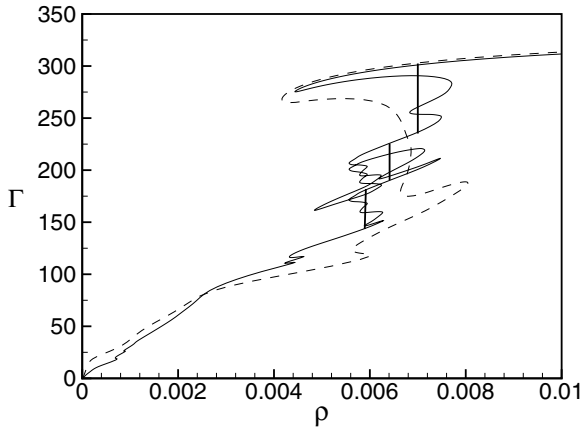

Figure 4. Adsorption Isotherms for 4 (broken curve) and 16 (full curve) random cylinders at $kT/\epsilon = 0.8$. In both cases, wall–fluid interactions are strongly attractive with $\epsilon_{wf}/\epsilon = 5$.

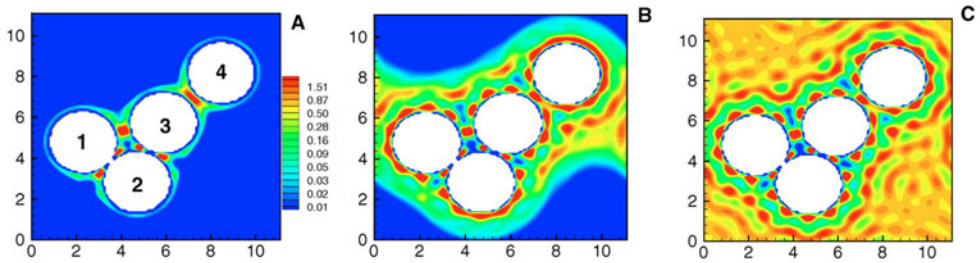

Figure 5. Density profiles for a Lennard-Jones fluid in a disordered porous matrix composed of cylindrical fibrous particles.

in figure 6. For an infinitely disordered system, these intermediate phase transitions will be layered on top of one another and the adsorption isotherm could appear to be continuous.

4.3. Confined diblock copolymers

Finally, the polymer-DFT algorithms give us the capability of studying self-assembly for multi-site fluids. In the future we anticipate applying these computational tools to lipid bilayers and to proteins in those bilayers. Previous studies have used DFT-like integral equation theories to study membrane protein interactions [21] and self-consistent field theory to study lipid phase behaviour [42]. The polymer-DFT presented here includes physics that is lacking in these studies. However, we have to date focused on a simpler well studied polymer system. Specifically, we have chosen to investigate confined block copolymers [43]. This system has been well studied in the past with self-consistent field theory [44]. The results of our DFT calculations are qualitatively comparable with previous calculations, but they include the effects of compressibility. Compressibility effects are particularly important near the surfaces where particle packing occurs. In the calculations presented below we have an 8-8 AB diblock copolymer where the A beads are attractive to the surfaces and the B beads are neutral (with interaction potential parameter $\epsilon_{wB} = 0$). Since we have a symmetric diblock, the system forms lamellae in the bulk.

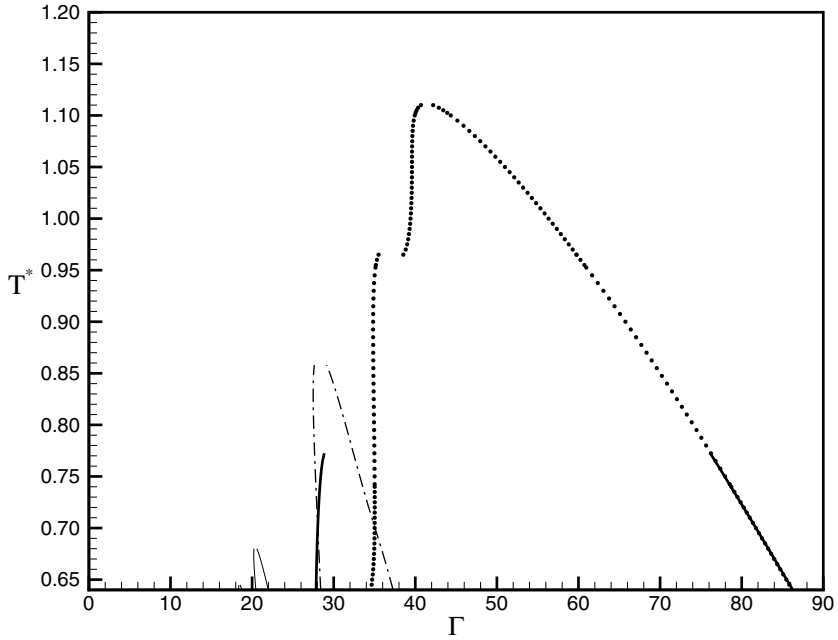

Figure 6. Phase diagram for a slightly disorderd porous matrix composed of four randomly placed cylindrical fibres. The excess adsorption, Γ is given per unit length of the fibres. It is calculated via $\Gamma = \int \mathrm{d}x \int \mathrm{d}y\, (\rho(x, y) - \rho_b)$.

If the diblock copolymer is confined between parallel planar surfaces, the fluid can in fact form lamellae that are either parallel to or perpendicular to the surfaces. Generally, the parallel phase is found whenever the surface separation can accommodate integral numbers of lamellae into the film. Mixed phases (perpendicular on one surface, parallel on the other) can also be found. Examples of these three possibilities are shown in the density plots in figure 7.

While a variety of morphologies are possible, the equilibrium structure is the one that minimizes the free energy. A comparison of the free energies of the various cases is shown in figure 8. While the mixed phase is never globally stable according to this plot, the parallel and perpendicular phases are both stable but at different surface separations. As expected, the parallel phase becomes stable when the surface separation is such that the lamellae can form with spacings close to the bulk spacing. Clearly, figure 8 shows that multiple solutions can exist to the polymer-DFT equations, and that phase diagrams are likely to be very rich. Once again the arclength continuation and phase transition tracking algorithms are important tools for exploring these phase diagrams.

5. Summary

In this paper we have summarized the activities in our group in developing advanced algorithms for two- and three-dimensional DFTs near complex surfaces. Fast algorithms are important because they will allow application of the DFT approach to complex interfaces in biology and nanotechnology. We have demonstrated that it is currently possible to study solvation of small proteins with our real-space matrix-based Newton algorithms and accelerated with efficient use of O(100) parallel processors. Given the clear advantages of the matrix-free

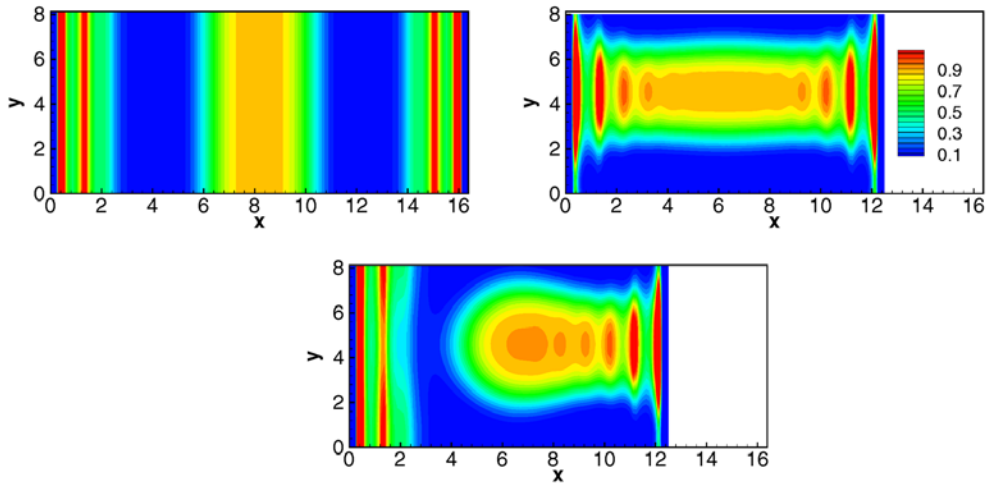

Figure 7. Density contour plot for three possible morphologies found in a confined symmetric diblock copolymer.

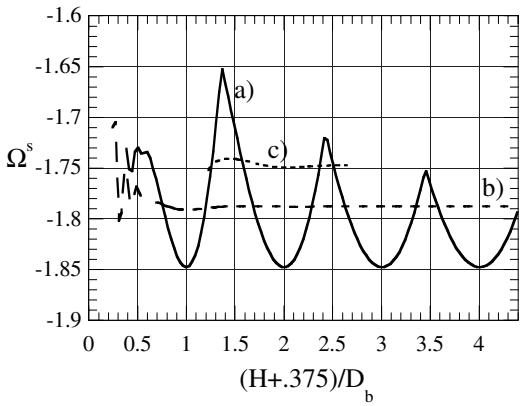

Figure 8. Excess surface free energy for parallel (a), perpendicular (b) and mixed (c) phases as a function of the wall separation H, normalized by the lamellar spacing in the bulk D_b.

Fourier space method outlined above, we expect to be able to treat protein solvation on single-processor workstations with that approach. We have also demonstrated that some of the algorithmic strategies developed to treat atomistic fluids can be immediately applied to polymer fluids. The polymer-DFT approach described here is much simpler than the weighted density approximation approaches to polymers described elsewhere [45]. However, our liquid state theory based approach does capture complex mesostructures as well as the interplay of those mesostructures with nanoscopic effects at surfaces.

Acknowledgments

LJDF would like to thank Frank van Swol for many years of enjoyable collaboration and Bob Eisenberg for an introduction to ion channels.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

This work has been supported by the DOE Office of Science under the MICS program and the Sandia LDRD program.

References

[1] Russel W B, Saville D A and Schowalter W R 1989 *Colloidal Dispersions* (Cambridge: Cambridge University Press)
[2] Evans R 1990 *J. Phys. Condens. Matter* **2** 8989
Rojas F, Kornhauser I, Felipe C and Cordero S 2001 *J. Mol. Catal.* A **167** 141
[3] Kierlik E, Monson P A, Rosinberg M L, Sarkisov L and Tarjus G 2001 *Phys. Rev. Lett.* **87** 55701
Sarkisov L and Monson P A 2001 *Phys. Rev.* E **65** 11202
[4] van Swol F and Henderson J R 1986 *J. Chem. Soc. Faraday Trans.* II **2** 1685
van Giessen A E, Bulkman D J and Widom B 1997 *J. Colloid Interface Sci.* **192** 257
van Swol F and Henderson J R 1989 *Phys. Rev.* A **40** 2567
van Swol F and Henderson J R 1991 *Phys. Rev.* A **43** 2932
Frink L J D and Salinger AG 1999 *J. Chem. Phys.* **110** 5969
[5] For a review see
Evans R 1992 Density functionals in nonuniform fluids *Fundamentals of Inhomogeneous Fluids* ed D Henderson (New York: Dekker)
[6] Chandler D, McCoy J D and Singer S J 1986 *J. Chem. Phys.* **85** 5971
Chandler D, McCoy J D and Singer S J 1986 *J. Chem. Phys.* **85** 5977
McCoy J D, Singer S J and Chandler D 1987 *J. Chem. Phys.* **87** 4853
[7] See, e.g.,
Woodward C W 1991 *J. Chem. Phys.* **94** 3183
Kierlik E and Rosinberg M L 1994 *J. Chem. Phys.* **100** 1716
Hooper J B, McCoy J D and Curro J G 2000 *J. Chem. Phys.* **112** 3090
Hooper J B, Pileggi T, Morgan J D, McCoy J G, Curro J D and Weinhold 2000 *J. Chem. Phys.* **112** 3094
[8] Frink L J D and Salinger A G 2000 *J. Comput. Phys.* **159** 407
Frink L J D and Salinger A G 2000 *J. Comput. Phys.* **159** 425
[9] Sears M P and Frink L J D 2002 A new efficient method for density functional theory calculations of inhomogeneous fluids *J. Comput. Phys.* submitted
[10] A few examples are
Israelachvili J N and Tabor D 1972 *Proc. R. Soc.* A **331** 19
Horn R G and Israelachvili J N 1980 *Chem. Phys. Lett.* **71** 192
Leckband D E, Israelachvili J N, Schmitt F-J and Knoll W 1992 *Science* **255** 1419
[11] Binning G, Quate C F and Gerber C 1986 *Phys. Rev. Lett.* **56** 930
Binning G, Gerber C, Stoll E, Albrecht T R and Quate C F 1987 *Surf. Sci.* **189** 1
[12] Pool R 1988 *Science* **241** 1042
[13] Parsegian V A, Rand R P and Rau D C 1995 *Methods Enzymol.* **259** 43
[14] Samuel J, Brinker C J, Frink L J D and van Swol F 1998 *Langmuir* **14** 2602
[15] Frink L J D and van Swol F 2000 *Colloids Surf.* A **162** 25
[16] Frink L J D and van Swol F 1996 *J. Chem. Phys.* **105** 2884
Frink L J D and van Swol F 1997 *J. Chem. Phys.* **106** 3782
Frink L J D and van Swol F 1998 *J. Chem. Phys.* **108** 5588
[17] Xia Y N and Whitesides G M 1998 *Ann. Rev. Mater. Sci.* **28** 153
[18] Brinker C J, Lu Y F, Sellinger A and Fan H Y 1999 *Adv. Mater.* **11** 579
Fan H Y, van Swol F, L u Y F and Brinker C J 2001 *J. Non-Cryst. Solids* **285** 71
[19] Last J A, Waggoner T A and Sasaki D Y 2001 *Biophys. J.* **81** 2737
[20] Ashcroft F M 2000 *Ion Channels and Disease* (San Diego, CA: Academic)
[21] Roux B 1999 *Biophys. J.* **77** 139
Lague P, Zuckermann M J and Roux B 2000 *Biophys. J.* **79** 2867
Lague P, Zuckermann M J and Roux B 2000 *Biophys. J.* **81** 276
[22] Im W, Beglov D and Roux B 1988 *Comput. Phys. Commun.* **111** 59
Roux B and Simonson T 1999 *Biophys. Chem.* **78** 1
Roux B and Simonson T 1997 *J. Phys. Chem.* B **101** 1190
[23] Hollarbach U, Chen D P, Busath D D and Eisenberg B 2000 *Langmuir* **16** 5509
Hollarbach U, Chen D-P and Eisenberg R S 2002 *J. Sci. Comput.* **16** 373

Nonner W, Chen D-P and Eisenberg B 1998 *Biophys. J.* **74** 2327
[24] Lowen H, Madden P A and Hanson J-P 1992 *Phys. Rev. Lett.* **68** 1081
Frink L J D and van Swol F 1994 *J. Chem. Phys.* **100** 9106
Im W, Seefeld S and Roux B 2000 *Biophys. J.* **79** 788
[25] Henderson D (ed) 1992 *Fundamentals of Inhomogeneous Fluids* (New York: Dekker)
[26] Rosenfeld Y 1989 *Phys. Rev. Lett.* **63** 980
Rosenfeld Y, Schmidt M, Lowen H and Tarazona P 1996 *J. Phys.: Condens. Matter* **8** L577
[27] Frischknecht A L, Weinhold J D, Salinger A G, Curro J G, Frink L J D and McCoy J D 2002 Density functional theory for inhomogeneous polymer systems: I numerical methods *J. Chem. Phys.* **117** at press
[28] Donley J P, Rajasekaran J J, McCoy J D and Curro J G 1995 *J. Chem. Phys.* **103** 5061
[29] For recent reviews see
Schweizer K S and Curro J G 1994 *Adv. Polym. Sci.* **116** 319
Schweizer K S and Curro J G 1997 *Adv. Chem. Phys.* **98** 1
[30] Zhou S 2000 *J. Chem. Phys.* **113** 8717
[31] Hansen J-P and McDonald I R 1986 *Theory of Simple Liquids* (New York: Academic)
[32] Weeks J D, Chandler D and Anderson H C 1971 *J. Chem. Phys.* **54** 5237
[33] Tang Z X, Teran L M Y, Davis H T, Scriven L E and White H S 1990 *Mol. Phys.* **71** 369
Tang Z X, Scriven L E and Davis H T 1994 *J. Chem. Phys.* **100** 4527
[34] Groh B, Evans R and Dietrich S 1998 *Phys. Rev.* E **57** 6944
[35] Keller H B 1977 *Applications of Bifurcation Theory* ed P H Rabinowitz (New York: Academic) pp 359–84
[36] Salinger A G, Bou-Rabee N M, Burroughs E A, Lehoucq R B, Pawlowski R P, Romero L A and Wilkes E D 2002 *Sandia National Laboratories Technical Report* SAND2002-0396 *(Albuquerque)*
[37] Salinger A G and Frink L J D 2002 Rapid analysis of phase behaviour with density functional theory: I. Novel numerical methods *J. Chem. Phys.* submitted
[38] Frink L J D and Mitchell M C 2002 *Biophys. J.* **82** 340A in preparation
[39] Chadwick D and Cardew G (ed) 1999 *Gramicidin and Related Ion Channel-forming Peptides* (New York: Wiley)
[40] Frink L J D, Thompson A and Salinger A G 2000 *J. Chem. Phys.* **112** 7564
[41] Frink L J D and Salinger A G 2002 Rapid analysis of phase behaviour with density functional theory: II. Capillary condensation in disordered porous media *J. Chem. Phys.* submitted
[42] Li X and Schick M 2000 *Biophys. J.* **78** 34
Li X and Schick M 2000 *J. Chem. Phys.* **112** 6063
[43] Frischknecht A L, Curro J G and Frink L J D 2002 Density functional theory for inhomogeneous polymer systems: II. Application to block copolymer thin films *J. Chem. Phys.* **117** at press
[44] For a recent review, see
Fasolka M J and Mayes A M 2001 *Annu. Rev. Mater. Res.* **31** 323
[45] Yethiraj A 1998 *J. Chem. Phys.* **109** 3269 and references therein