Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 212 (2006) 25–51
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# Application of finite-difference methods to membrane-mediated protein interactions and to heat and magnetic field diffusion in plasmas

Gennady V. Miloshevsky [a], Valeryi A. Sizyuk [b], Michael B. Partenskii [a], Ahmed Hassanein [b], Peter C. Jordan [a,*]

[a] *Department of Chemistry, MS-015 Brandeis University, P.O. Box 549110, Waltham, MA 02454-9110, USA*
[b] *Energy Technology Division, Argonne National Laboratory, 9700 South Cass Avenue, Bldg. 308, Argonne, IL 60439, USA*

Received 7 March 2005; received in revised form 17 June 2005; accepted 21 June 2005
Available online 8 August 2005

**Abstract**

A robust finite-difference approach for solving physically distinct cross-disciplinary problems such as membrane-mediated protein–protein interactions and heat and magnetic field diffusion in plasmas is described for rectangular grids. Mathematical models representing these physical phenomena are fourth- and second-order partial differential equations with variable coefficients. The finite-difference coupled harmonic oscillators technique was developed to treat arbitrary aggregates of inclusions in membranes automatically accounting for their non-pairwise interactions. The method was applied to study the stabilization of ion channels in a cluster due to membrane-mediated interactions and to examine the effects of anisotropic membrane slope relaxation on the elastic free energy. To obtain contributions from heat and magnetic field diffusion, the splitting method for the physical processes has been used in the numerical solution of resistive magnetohydrodynamic equations. The fully implicit scheme is outlined, tested and applied to problems of the diffusive redistribution of magnetic field and heat in the plasma.
© 2005 Elsevier Inc. All rights reserved.

*MSC:* 35J40; 35K15; 65N06; 65N22; 65F50

*PACS:* 87.14.Cc; 87.15.Kg; 61.30.Dk; 52.25.Xz; 52.55.−s; 44.05.+e

*Keywords:* Finite-difference method; Elliptic and parabolic equations; Membrane-mediated interaction; Heat; Magnetic field; Diffusion

* Corresponding author. Tel.: +1 781 736 2540; fax: +1 781 736 2516.
*E-mail address:* jordan@brandeis.edu (P.C. Jordan).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.06.013

## 1. Introduction

Many scientific problems require simulating continuous physical systems, such as those involving fluids, plasma flows, lipid membranes and liquid crystals. Even though these are many particle systems, at the macroscopic level they behave as continuous entities in a wave-like or fluid fashion. Mathematical models representing continuous physical phenomena are partial differential equations (PDEs). Unfortunately most PDEs representing realistic problems (rather than idealizations) are too complex to be solved analytically. Therefore, various classes of numerical methods have been developed to computationally solve PDEs [1]. Numerical solution involves two tasks: (1) choosing a discretization scheme to transform the PDE into a discrete problem that approximates it and (2) selecting a solution method for the discretized problem. Discretization procedures use either finite difference (FD) or finite-element methods [2,3] resulting in a large, sparse system of linear algebraic equations. The number of unknowns may vary from hundreds to millions; their determination is the most time consuming step. Our focus is on robust FD techniques for solving second- and fourth-order PDEs with variable coefficients. FD algorithms have the advantage of being simply formulated for two-dimensional (2D) problems, which implies that they can be quickly adapted to realistic problems that are of both theoretical and practical interest in physics and chemistry.

Computationally, the physical processes may be treated as boundary value and initial value problems. Steady-state processes, which are time independent, can be described by elliptic PDEs [4]. Elliptic equations model the behavior of scalar quantities, such as temperature, gravitational potential, electromagnetic fields, membrane distortion fields, etc. These lead to boundary value problems. The PDEs must be satisfied at all points in the interior of the computational domain with appropriate boundary conditions specified on all boundaries within this domain. The goal is to determine the functions at all interior points. Initial value problems characterize dissipative physical processes, such as heat or magnetic field diffusion in a plasma, which are evolving toward a steady state. Such phenomena are described by parabolic PDEs [5]. For all initial value problems with constant boundary constraints, the solutions decay from an initial state to a non-varying steady state. Thus, the steady-state limit of parabolic time-dependent problems are solutions to boundary value problems. Their transient behavior is smooth and bounded and the solution does not develop local or global maxima that are outside the range of the initial data. Therefore, their solution can be 'driven' to the steady state using implicit time-stepping techniques [6]. The time step is represented as an adjustable relaxation parameter. If it is chosen as large as possible, then small scale evolution details are inessential. Put differently, averaging of small scale changes in temperature or magnetic field is performed implicitly during the large time step instead of after a huge sequence of small explicit time steps.

Our focus is on membrane-mediated protein–protein interactions [7] and problems of heat and magnetic field diffusion in plasmas [8]. These are the examples of physically distinct cross-disciplinary problems. Elastic membrane deformation due to embedded proteins leading to long-range membrane-mediated interactions can be described by a fourth-order elliptic Euler–Lagrange equation [7]. We developed a finite-difference coupled harmonic oscillators (FD-CHO) approach to treat this problem for arbitrary protein aggregates. Understanding resistive magnetohydrodynamic (MHD) plasma phenomena such as heat and magnetic field diffusion is of practical importance in dense plasma focus (DPF) and fusion plasmas. DPF devices produce the most flexible and advanced plasmas, sources of intense radiation and of charged particles for materials, medical and environmental applications. Using a splitting method for the physical processes [9–11], we decouple the effects of heat and magnetic field diffusion from the effects of plasma hydrodynamics within a moving plasma. Heat and magnetic field diffusion in plasmas is described by second-order parabolic equations. We employ a fully implicit (backward time) scheme [6] to approximate these equations. Finally, both the FD-CHO method for membrane-mediated protein–protein interactions and the fully implicit method for time-dependent heat and magnetic field diffusion in plasmas yield similar systems of linear algebraic equations with a sparse matrix. The diffusion problems require a system of linear equations to be solved at each time step.

## 2. Physical model of membrane deformation

A lipid bilayer is a bimolecular sheet embedded between aqueous phases, with its hydrophilic heads facing the water and hydrophobic chains in the interior. Models based on continuum elastic theories treat the lipid bilayer as a stratum of a smectic liquid crystal. Biological membranes contain a large number of mobile inclusions such as embedded proteins. Besides interacting directly via electrostatic or van der Waals forces, inclusions are also coupled indirectly via the elastic perturbation of the membrane. The hydrophobic length of an inclusion generally does not equal the thickness of the inclusion-free membrane; thus the bilayer thickness at the protein/bilayer boundary must adjust itself. This inclusion induced membrane deformation can be felt by a nearby inclusion, thus generating an effective membrane-mediated force. Earlier studies of membrane-mediated protein–protein interactions were based on mean-field theories [12–16]. In these, a Landau free energy [17] is expanded in an order parameter (variation of membrane thickness) and its gradient, which are both directly related to membrane fluctuations. These earlier studies were limited by the approximations needed to formulate appropriate analytical models. To describe protein clusters, an analytical expression for the total free energy was derived, based on a pairwise superposition approximation [16]. An approach using membrane elasticity theory to model membrane–inclusion interactions induced by hydrophobic mismatch was described by Huang [18]. It considers two contributions to the free energy, stretching/compression and bending (see also [19,20]), and predicts short-range membrane perturbation near a membrane inclusion, decaying with a characteristic length of ~10–20 Å. More recent work [21] treats the proteins on a hexagonal lattice associating a 2D Wigner–Seitz cell with each inclusion, assuming the membrane perturbation around each inclusion is radially symmetric. It was easier to compute the energy of an array of inclusions than to exactly calculate the interaction between an inclusion pair and the total perturbation free energy of the membrane was treated as a sum of single inclusion contributions [22]. This hexagonal approximation is reasonable for low inclusions densities, i.e. for well-separated inclusions. Recently, using a multipole expansion of a mean curvature field, it was shown that interaction involving three or more inclusions is not pairwise additive, i.e. the total energy of a protein ensemble is not a sum of two-body terms [23]. This has important implications in establishing the existence of stable protein aggregates of five or more inclusions. To treat many-body effects, a numerical method was devised for exactly solving the Euler–Lagrange equation by FD [20]. However, since finer grids required excessively long computation times, the mesh was coarse, with spacings of 3.6 Å, and inadequate to properly describe membrane distortion near inclusion boundaries. Thus, the Euler–Lagrange equation was solved assuming the membrane distortion field near inclusions was cylindrically symmetric. While adequate for regular protein aggregates and lattice arrays, it is less appropriate for the protein clusters of general geometry observed in Monte Carlo simulation [20]. Here, we present the FD-CHO technique, with which it is possible to exactly treat many-body problems with no unreasonable assumptions. It is applicable to general inclusion configurations and clusters of arbitrary size. A fine FD mesh spacing (~0.4–0.5 Å) can be used; thus membrane distortion fields near inclusions need not be cylindrically symmetric. A preliminary description was given in a recent review [7] and it has been applied to study channel stabilization [24] and anisotropic membrane slope relaxation [25] due to membrane-mediated elastic interactions.

Because the lipophilic exteriors of inclusions embedded in membranes generally differ in length from the surrounding hydrophobic region of the bilayer, the membrane deforms in accommodation (Fig. 1). We consider the simplest case, $N$ embedded rigid cylindrical inclusions, denote the membrane's displacement from its flat state as $u(x,y)$ and assume that the membrane is only slightly perturbed, in which case a quadratic approximation is valid and the free energy of the membrane is a harmonic functional of $u(x,y)$ and its derivatives [18,20]. Deformation energetics depends on the material properties of the membrane. The macroscopic coefficients needed to describe the physical problem are $B$, the elastic stretching modulus (due to

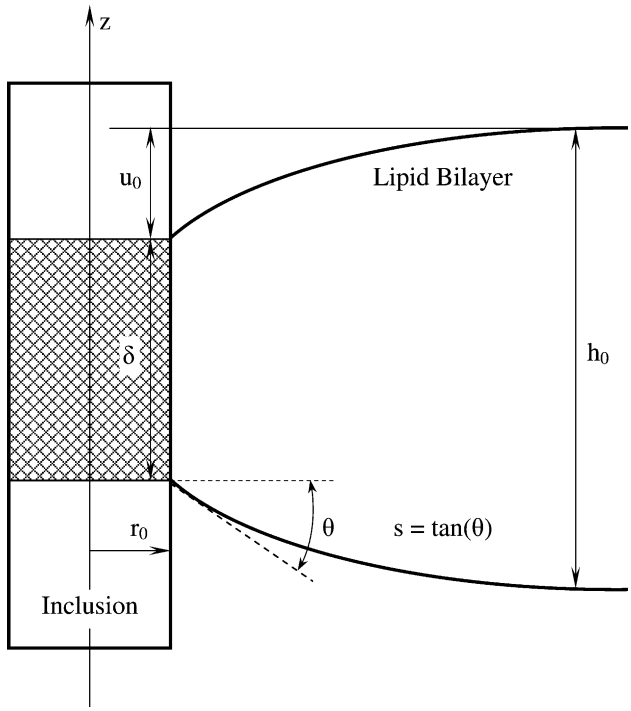


Fig. 1. Schematic illustration of a deformed bilayer matching the hydrophobic region of an inclusion. The bilayer deformation profile is shown for the boundary condition $s = s_{\min}$. $h_0$ denotes the equilibrium thickness of the unperturbed bilayer, $u_0$ the deformation depth of the monolayer, $\delta$ the hydrophobic length of inclusion, $r_0$ the radius of inclusion and $s$ the contact slope at inclusion–bilayer boundary.

membrane thickness variation), $K$, the elastic bending modulus (due to the tilt of phospholipid chains) and $\gamma$, the surface tension coefficient. In what follows we neglect surface tension, an justifiable approximation for solvent-free membranes [19,20]. In the harmonic approximation, the inclusion induced deformation free energy of the membrane is a surface integral

$$F = \int\int_S \left[ b(x,y)(\Delta u(x,y))^2 + a(x,y)u^2(x,y) \right] \mathrm{d}x\,\mathrm{d}y, \tag{1}$$

where $a(x,y) = 2B(x,y)/h_0$, $b(x,y) = K(x,y)h_0/2$, $\Delta = \nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $h_0$ is the thickness of the unperturbed membrane and, near inclusions, the membrane elastic constants, $B(x,y)$ and $K(x,y)$, may differ from their unperturbed bulk values [26]. At thermodynamic equilibrium, the free energy $F$ is minimized. Applying the variational principle and minimizing the functional $F$ with respect to $u(x,y)$ [27], yields the Euler–Lagrange equation

$$\Delta(b(x,y)\Delta u(x,y)) + a(x,y)u(x,y) = 0. \tag{2}$$

This is a fourth-order, linear, elliptic partial differential equation with the non-uniform coefficients describing the deformation surface $u(x,y)$ that minimizes $F$. Solving (2) with appropriate boundary conditions we find a 2D displacement field $u(x,y)$ and then determine the minimum free energy $F$ using (1). Eq. (2) is a fourth-order differential equation, which requires four boundary conditions. At the external boundary of a computational domain both $u(x,y)$ and $\nabla u(x,y)$ vanish, i.e. the membrane perturbation approaches zero. It is associated with areas "infinitely distant" from a given cluster of $N$ inclusions, leading to two conditions at the external membrane boundary

$$u(x,y)|_{\infty} = 0, \quad \nabla u(x,y)|_{\infty} = 0. \tag{3}$$

The other two conditions are formulated on the cylindrical boundary $\Gamma_\kappa$ of each of the $\kappa$ inclusions (Fig. 1) as

$$u(x,y)|_{\Gamma_\kappa} = u_{0\kappa}, \quad \nabla u(x,y)|_{\Gamma_\kappa} = s_\kappa, \tag{4}$$

where $u_{0\kappa} = (h_0 - \delta_\kappa)/2$ is the displacement of the membrane from its unperturbed state at the contact surface of the membrane and the $\kappa$th inclusion, $\delta_\kappa$ is the corresponding hydrophobic length and $s_\kappa$ is the corresponding contact slope. We assume strong hydrophobic coupling between the hydrophobic region of the bilayer and the hydrophobic exterior surface of the embedded inclusions [21]. The choice of the contact slope boundary condition is still a subject of controversy. The variation of gramicidin A (gA) channel lifetime as a function of bilayer thickness was used [20,28] to determine the contact slope. The so-called "null constraint" boundary condition, $s_\kappa = 0$, accounts for the effect of membrane thickness on gA single channel lifetimes [20]. However, this "null constraint" restricts the contact slope, only possible if a boundary energy term, of unspecified physical origin, is postulated. A more physically attractive development presumes that this slope adjusts itself (the "relaxed slope") to minimize the free energy of the bilayer deformation [19]. However, the "relaxed slope" boundary condition $s_\kappa = s_{\min}$ yields much lower deformation free energies than the "null constraint" condition, with results that are inconsistent with the gA lifetime measurements. Recent theoretical work [26] shows this discrepancy can be eliminated by considering inclusion-induced local membrane rigidity (perturbation of membrane elastic moduli) and further, that this can be identified with the boundary energy introduced to justify the "null constraint" [25]. The appropriate choice of the contact slope is still an open question since the bilayer/inclusion interface is inadequately characterized experimentally [29].

The solution of (2) with the boundary conditions (3) and (4) describes the equilibrium state of the membrane surface deformed in response to embedded inclusions and (1) determines the bilayer's distortion free energy. For a single inclusion, and assuming radial symmetry, analytical expressions for both the deformation profile and deformation energy can be derived [7]. However, with two or more inclusions, the surface shape and the total deformation energy cannot be found analytically. To solve (2) for arbitrary protein aggregates, numerical methods are the only alternative. However, numerical solutions require a discrete computational domain and governing equations reduced to their FD equivalents.

## 3. Physical model of heat and magnetic field diffusion in plasmas

Plasmas play a key role in controlled fusion as well as in fields such as material processing, particle accelerators and astrophysics. A standard way to study their dynamics is to treat the plasma as a magnetized fluid [30]. In such fluids both the velocity and magnetic fields are physically coupled, i.e. a perturbation of the velocity field induces a magnetic response and a perturbation of the magnetic field results in a change in the motion of the plasma. To simulate plasma behavior a system of very complicated MHD equations, comprising a combination of the Euler equations of gas dynamics and the Maxwell equations of electromagnetism and their associated boundary conditions, is formulated to describe both the plasma and the magnetic field. The unsteady system of resistive single fluid non-relativistic MHD is given by

$$\partial_t U + \nabla \cdot \mathbf{F} = Q, \tag{5}$$

where $U$ is the state vector (plasma density, momentum, total energy density and magnetic field), $\boldsymbol{F}$ is the corresponding flux vector, $Q$ is the source term describing dissipation (heat conduction, magnetic diffusion, viscosity, etc.) and $\partial_t$ is the partial differential operator with respect to time $t$. Eq. (5) describe the time dependence of physical quantities such as plasma density, velocity, temperature and magnetic field. The

ideal MHD equations ($Q = 0$) are hyperbolic, but including resistivity changes the mathematical form to mixed hyperbolic–parabolic.

We wish to extend the range of validity of existing ideal MHD models to address realistic plasma conditions. The solution of general MHD problems with arbitrary geometries and more complicated processes involving other physical phenomena such as thermal conduction, radiative transfer, magnetic field diffusion and changes of state is of great practical importance. Formation of the plasma focus in the DPF device [31,32] exemplifies problems involving coupled physical processes with multiple scales where the equations for the distinct physical processes are best solved by different numerical techniques. A powerful approach to such problems is a splitting method [9–11], which involves decoupling the full model into a separate component for each process, employing specialized numerical methods to solve each component, and coupling the resulting solutions. Thus, the MHD equations are solved as a decoupled set of hyperbolic and parabolic equations. At each time step the MHD problem (5) is split into decoupled subproblems (which may involve different meshes and solution methods) corresponding to the different physical processes (e.g. plasma flow, transport, diffusion, reactions, etc.) that occur within the computational domain or in individual regions. The various processes are treated sequentially and MHD physics updated after each separate contribution. Therefore, numerical algorithms constructed according to the principle of splitting by physical process are strictly constrained by their order of execution. The splitting error consists of a physical splitting error that would exist even if were the subproblems solved exactly (indicative of the way that subproblems are linked), and a numerical splitting error, related to approximating each subproblem.

The splitting algorithm has been used in our numerical solution of the MHD equations (5) to separate contributions from heat and magnetic field diffusion. These processes redistribute the internal energy and magnetic flux in the plasma. The resulting enthalpy conservation and magnetic field diffusion equations, obtained by splitting the MHD system (5) according to the nature of the physical processes, can be written in 2D cylindrical coordinates as

$$\frac{\partial \rho H}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(rK\frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(K\frac{\partial T}{\partial z}\right) \tag{6}$$

and

$$\frac{\partial B}{\partial t} = \frac{c^2}{4\pi\mu}\frac{\partial}{\partial r}\left(\frac{\eta}{r}\frac{\partial rB}{\partial r}\right) + \frac{c^2}{4\pi\mu}\frac{\partial}{\partial z}\left(\eta\frac{\partial B}{\partial z}\right), \tag{7}$$

where $r$ and $z$ are axisymmetric coordinates, $\rho$ the plasma density, $H$ the specific enthalpy, $K$ the local thermal conductivity of the plasma, $T$ the plasma temperature, $B = B_\varphi$ the azimuthal magnetic field, $c$ the speed of light, $\mu$ the magnetic permeability and $\eta$ the magnetic diffusivity. We assume axial symmetry about the $z$-axis, i.e. solutions are $\varphi$-independent. The current through the plasma is constrained to the $(z - r)$ plane. The magnetic field has an azimuthal component $B_\varphi$ (the self-induced magnetic field) [33]. The enthalpy equation (6) is parabolic, and describes the spatial and time variation of heat flow as well as diffusion. The relationship between the specific enthalpy and temperature is $H = c_p T$, where $c_p$ is the constant pressure heat capacity. Eq. (7) describes the resistive diffusion of the magnetic field through the plasma. Under the influence of finite resistivity the magnetic field diffuses across the plasma and field inhomogeneities are smoothed out.

For illustration consider the schematic plasma chamber of Fig. 2(a). Our aim is to describe the boundary conditions. There are four boundaries: the left ($r = 0$), right ($r = a$), bottom ($z = 0$), and top ($z = b$). At the outer chamber boundary ($r = a$), the computational domain contains an assembly of trapezium-shaped cathodes and anodes. Treating the chamber as cylindrically symmetric, we choose ABCDEFGHKLMN as the outer calculation boundary $\Gamma$. Fine details of the computational grid and our approximation of the ABC cathode boundary are illustrated in Fig. 2(b). Before solving (6) and (7), both initial and boundary conditions must be specified. Initial conditions are the initial temperature and magnetic field values

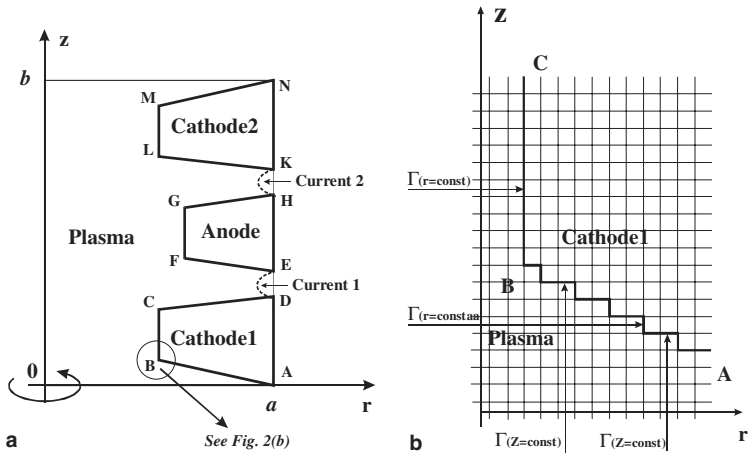


Fig. 2. The schematic diagram of the computational domain. (a) Schematic of the electrode arrangement in the plasma chamber. (b) Fine computational grid and rectangular approximation of the electrode boundaries.

assigned to interior points of the computational domain. The boundary conditions can take many forms. We use Neumann boundary conditions [6], specifying the normal temperature gradients, $\nabla T = 0$, on the computational domain boundaries. The boundary conditions for $B$ are more complex. We require the magnetic field to be sufficiently well behaved at the symmetry axis ($r = 0$) that the differential operators in (7) are non-singular. Axial symmetry implies that the azimuthal magnetic field vanishes, $B = 0$ at $r = 0$, with $\partial B^+/\partial r = \partial B^-/\partial r$ at $r \to 0$, where $B^+$ and $B^-$ are the azimuthal magnetic fields on the left and right sides of the symmetry axis. At the outer boundary $\Gamma$, the boundary conditions are motivated by the physics. We use one of the following sets of constraints: current flow, $B = 2I/rc$; Neumann boundary conditions, $\partial rB/\partial r = 0$ and $\partial B/\partial z = 0$; or conducting wall, $B = 0$.

To solve parabolic equations like (6) and (7), we use the fully implicit scheme [6] which generates a sequence of elliptic problems in the limit of large time steps. During each large time step the implicit scheme drives small scale changes of the temperature or magnetic field to equilibrium states satisfying (6) and (7) with the left-hand side set to zero. This time step is viewed as the "relaxation time" to the steady state. At a given time step the resulting equations are elliptic or Poisson's equations for $T$ and $B$. Space variables are discretized just as in steady state heat and magnetic field diffusion problems. The solution from the previous time step provides an initial guess for the new one. Thus, for the large time steps allowed in the fully implicit scheme, the parabolic equations (6) and (7) can be reduced to the solution of elliptic equations for each time slice of the spacetime. In summary, the main advantage of the fully implicit scheme is that we have accurate error control in the time step selection process allowing step sizes to automatically adjust to the problem physics while maintaining accuracy.

## 4. Finite-difference approximations

Suppose that the space continuum is replaced by a discrete spatial mesh (Fig. 3). We seek FD approximations to the $n$th-order derivative $\partial^n f(x)/\partial x^n$ on the $q$-point stencil, i.e. the function is expanded over $q$ discrete nodes with an error of order $o = q - n$, which can be specified a priori. The fewest discrete nodes $q$ needed to approximate the $n$th-order derivative is $n + 1$. More nodes provide more accurate FD approximations. For any node, the stencil defines node connectivity, i.e. which other nodes determine the

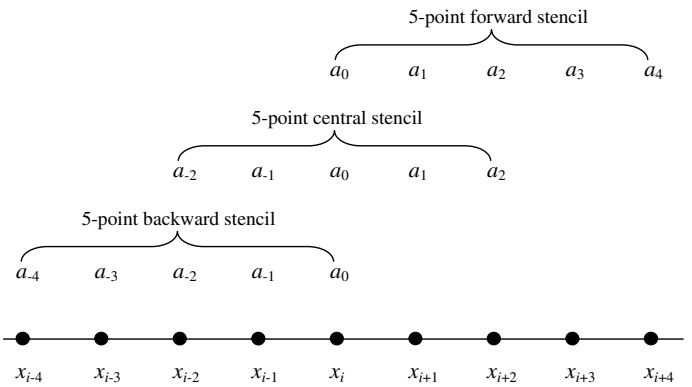


Fig. 3. Grid spacing for FD approximations of partial derivatives. The grid points are denoted by the solid circles. The derivative is taken at $x_i$. Five-point ($q = 5$) backward ($k_s = -4$ and $k_e = 0$), central ($k_s = -2$ and $k_e = 2$) and forward ($k_s = 0$ and $k_e = 4$) stencils are illustrated; $a_k$ are template coefficients.

derivative. Let $\Delta_{n,o}^{x_i}$ denote the FD approximation for the derivative $n$ with truncation error $o$. The superscript $x_i$ denotes the node $x_i$, where the derivative is evaluated (Fig. 3). We approximate the $n$th-order derivative over $q$ nodes as a linear combination

$$\frac{\partial^n f(x)}{\partial x^n} \approx \Delta_{n,o}^{x_i} f(x) = \sum_{k=k_s}^{k_e} \xi_k f_k, \tag{8}$$

where $\xi_{k_s}, \ldots, \xi_{k_e}$ are unknowns and, for a $q$-point stencil, $k$ extends from $k_s$ to $k_e$, i.e. $q = k_e - k_s + 1$. The vector $\Lambda_{n,o}^{x_i} = (\xi_{k_s}, \ldots, \xi_k, \ldots, \xi_{k_e})$ is the approximation template. Depending on the number of stencil nodes preceding or following $x_i$ the FD approximation is characterized as backward, central or forward (Fig. 3). As $q = k_e - k_s + 1$, setting $k_s = 0$ and $k_e = q - 1$ leads to a forward difference and setting $k_s = -(q-1)$ and $k_e = 0$ leads to a backward difference approximation. Setting $k_s = -\lfloor (q-1)/2 \rfloor$ and $k_e = \lfloor (q-1)/2 \rfloor$ produces a central difference approximation. Here brackets mean a truncation to integers, i.e. any remainder is dropped. The Taylor series expansion for each node $k$ about grid point $x_i$ is

$$f_k = f_i + \Delta x_k \frac{\partial f_i}{\partial x} + \frac{1}{2}\Delta x_k^2 \frac{\partial^2 f_i}{\partial x^2} + \frac{1}{6}\Delta x_k^3 \frac{\partial^3 f_i}{\partial x^3} + \cdots, \tag{9}$$

where $\Delta x_k = x_k - x_i$ is the spacing between nodes $x_k$ and $x_i$ and $f_i$ is a function evaluated at $x_i$. Substituting (9) into (8) and gathering terms yields

$$\sum_{k=k_s}^{k_e} \xi_k f_k = \sum_{k=k_s}^{k_e} \xi_k f_i + \sum_{k=k_s}^{k_e} \xi_k \Delta x_k \frac{\partial f_i}{\partial x} + \sum_{k=k_s}^{k_e} \xi_k \frac{1}{2}\Delta x_k^2 \frac{\partial^2 f_i}{\partial x^2} + \cdots \tag{10}$$

Introducing

$$A_m = \sum_{k=k_s}^{k_e} \xi_k \frac{1}{m!}\Delta x_k^m \tag{11}$$

for $m = 0, \ldots, q-1$ we find (10)

$$\sum_{k=k_s}^{k_e} \xi_k f_i + \sum_{k=k_s}^{k_e} \xi_k \Delta x_k \frac{\partial f_i}{\partial x} + \cdots + \sum_{k=k_s}^{k_e} \xi_k \frac{1}{(q-1)!}\Delta x_k^{q-1} \frac{\partial^{q-1} f_i}{\partial x^{q-1}} = A_0 f_i + A_1 \frac{\partial f_i}{\partial x} + \cdots + A_{q-1} \frac{\partial^{q-1} f_i}{\partial x^{q-1}}. \tag{12}$$

Eq. (12) is true if we set

$$A_m = \sum_{k=k_s}^{k_e} \xi_k \frac{1}{m!} \Delta x_k^m = \begin{Bmatrix} 1, & m = n \\ 0, & m \neq n \end{Bmatrix},$$

which yields $q$ linear equations in $k_e - k_s + 1$ unknowns. Multiplying (12) by $m!$ the equations for $\xi_k$ can be formulated as

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \Delta x_{k_s} & \Delta x_{k_s+1} & \Delta x_{k_s+2} & \cdots & \Delta x_{k_e} \\ \Delta x_{k_s}^2 & \Delta x_{k_s+1}^2 & \Delta x_{k_s+2}^2 & \cdots & \Delta x_{k_e}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_{k_s}^{q-1} & \Delta x_{k_s+1}^{q-1} & \Delta x_{k_s+2}^{q-1} & \ldots & \Delta x_{k_e}^{q-1} \end{bmatrix} \begin{bmatrix} \xi_{k_s} \\ \xi_{k_s+1} \\ \xi_{k_s+2} \\ \vdots \\ \xi_{k_e} \end{bmatrix} = \begin{bmatrix} 0!A_0 \\ 1!A_1 \\ 2!A_2 \\ \vdots \\ (q-1)!A_{q-1} \end{bmatrix}. \tag{13}$$

As long as the determinant of the coefficient matrix in (13) is non-zero, this system of linear equations determines the $\xi_k$ required in Eq. (8).

For multi-variant functions, mixed partial derivatives can be constructed as a tensor product of templates for functions of one variable. If $f(x,y)$ is a function of two variables, then the mixed partial derivative is obtained sequentially, first applying the $x$-derivative approximation, and using this as input for the $y$-derivative approximation. Both $x$ and $y$ partial derivatives have the same truncation error order, $o$. With $\Lambda_{n_x,o}^{x_i} = (\xi_{k_s}, \xi_{k_s+1}, \ldots, \xi_k, \ldots, \xi_{k_e})$ a template for the $x$-derivative evaluated on the $q_x$-point stencil taken at grid point $x_i$ and with $\Lambda_{n_y,o}^{y_j} = (\zeta_{l_s}, \zeta_{l_s+1}, \ldots, \zeta_l, \ldots, \zeta_{l_e})$ the corresponding $y$-derivative template ($q_y$-point stencil, grid point $y_j$) (Fig. 4), the mixed approximation template is determined by multiplying the two matrices

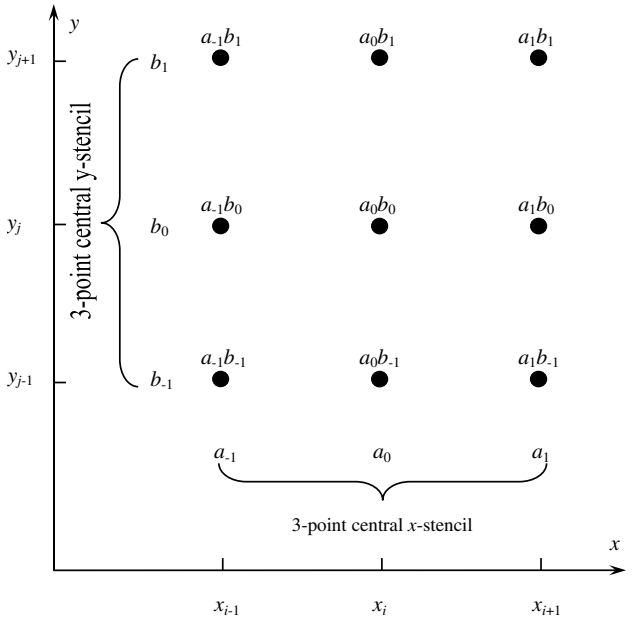


Fig. 4. Construction of a template for a function with two variables to approximate the mixed partial derivatives. The template corresponding to the central difference approximation for $\partial^2 f(x,y)/\partial x\,\partial y$ is illustrated. The 3-point stencils are shown for both $x$ and $y$ derivatives. For both directions the starting and ending indexes are $k_s = -1$ and $k_e = 1$, respectively. The resulting template for the mixed approximation is a $3 \times 3$ matrix. The coefficients of this matrix are indicated at the grid points denoted by the solid circles.

$$\begin{pmatrix} \xi_{k_s} & \xi_{k_s+1} & \cdots & \xi_{k_e} \end{pmatrix} \otimes \begin{pmatrix} \zeta_{l_s} & \zeta_{l_s+1} & \cdots & \zeta_{l_e} \end{pmatrix} = \begin{pmatrix} \xi_{k_s}\zeta_{l_s} & \xi_{k_s}\zeta_{l_s+1} & \cdots & \xi_{k_s}\zeta_{l_e} \\ \xi_{k_s+1}\zeta_{l_s} & \xi_{k_s+1}\zeta_{l_s+1} & \cdots & \xi_{k_s+1}\zeta_{l_e} \\ \cdots & \cdots & \cdots & \cdots \\ \xi_{k_e}\zeta_{l_s} & \xi_{k_e}\zeta_{l_s+1} & \cdots & \xi_{k_e}\zeta_{l_e} \end{pmatrix}. \tag{14}$$

Thus, for two variable functions the approximation coefficients are the tensor product of coefficients for the approximations to each one variable function. The upper left corner of the composite template corresponds to the most negative terms and the lower right corner corresponds to the most positive terms in FD approximations of the mixed partial derivatives (Fig. 4).

## 5. FD-CHO technique for Euler–Lagrange equation

The Euler–Lagrange equation (2) can be explicitly expressed in coordinate form as

$$\frac{\partial^2}{\partial x^2}\left(b(x,y)\frac{\partial^2 u(x,y)}{\partial x^2}\right) + \frac{\partial^2}{\partial x^2}\left(b(x,y)\frac{\partial^2 u(x,y)}{\partial y^2}\right) + \frac{\partial^2}{\partial y^2}\left(b(x,y)\frac{\partial^2 u(x,y)}{\partial x^2}\right) + \frac{\partial^2}{\partial y^2}\left(b(x,y)\frac{\partial^2 u(x,y)}{\partial y^2}\right) + a(x,y)u(x,y) = 0. \tag{15}$$

We use the following notation for the symbols in (15):

$$\frac{\partial^2 u(x,y)}{\partial x^2} = \partial_{xx}u(x,y), \tag{16}$$

$$\frac{\partial^2 u(x,y)}{\partial y^2} = \partial_{yy}u(x,y), \tag{17}$$

$$h(x,y) = b(x,y)\partial_{xx}u(x,y), \tag{18}$$

$$g(x,y) = b(x,y)\partial_{yy}u(x,y) \tag{19}$$

and write the Euler–Lagrange equation (15) as

$$\partial_{xx}h(x,y) + \partial_{xx}g(x,y) + \partial_{yy}h(x,y) + \partial_{yy}g(x,y) + a(x,y)u(x,y) = 0. \tag{20}$$

The basic FD strategy is discretization of the computational domain in each dimension. This generates a mesh with coordinates $x_i$ and $y_j$, where $i, j$ are integers. With variable spacing, distances between grid points $\Delta x_i = x_i - x_{i-1}$ and $\Delta y_j = y_j - y_{j-1}$ can be specified. The unknown $u$ and the coefficients $a$ and $b$ are approximated as $u(x_i, y_j) = u_{i,j}$, $a(x_i, y_j) = a_{i,j}$ and $b(x_i, y_j) = b_{i,j}$ and the variable coefficients $a$ and $b$ are assigned to nodes $(i,j)$ prior to calculation. Known boundary values of $u_{ij}$ are assigned to grid points of the external boundary and each inclusion boundary. We approximate second-order derivatives at $(i,j)$ using centered differences with a truncation error of order $o = 2$; thus the starting and ending indexes are $k_s = -1$ and $k_e = 1$, respectively. Higher order, more accurate FD approximations can also be formulated. However, this significantly increases memory requirements and computational demands. The linear system (13) determines the template coefficients for $\partial_{xx}u$

$$\begin{bmatrix} 1 & 1 & 1 \\ -\Delta x_i & 0 & \Delta x_{i+1} \\ (-\Delta x_i)^2 & 0 & \Delta x_{i+1}^2 \end{bmatrix} \begin{bmatrix} c_{i-1}^{x_i} \\ c_i^{x_i} \\ c_{i+1}^{x_i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}. \tag{21}$$

For convenience, we use the index $i$ instead of the stencil's index $k$. The superscript $x_i$ on the $c$ coefficients denotes the node where the derivative is evaluated and the solution to (21) is

$$
\begin{aligned}
c_{i-1}^{x_i} &= 2/\left(\Delta x_i \Delta x_{i+1} + \Delta x_i^2\right),\\
c_i^{x_i} &= -2/\left(\Delta x_i \Delta x_{i+1}\right),\\
c_{i+1}^{x_i} &= 2/\left(\Delta x_i \Delta x_{i+1} + \Delta x_{i+1}^2\right).
\end{aligned}
\tag{22}
$$

Template coefficients depend on grid spacing. From (8), $\partial_{xx}u$ is

$$
\left(\partial_{xx}u\right)_{i,j} = c_{i-1}^{x_i} u_{i-1,j} + c_i^{x_i} u_{i,j} + c_{i+1}^{x_i} u_{i+1,j} \tag{23}
$$

and, by analogy, the FD approximation to $\partial_{xx}h$ is

$$
\left(\partial_{xx}h\right)_{i,j} = c_{i-1}^{x_i} b_{i-1,j} \left(\partial_{xx}u\right)_{i-1,j} + c_i^{x_i} b_{i,j} \left(\partial_{xx}u\right)_{i,j} + c_{i+1}^{x_i} b_{i+1,j} \left(\partial_{xx}u\right)_{i+1,j} \tag{24}
$$

using (18) for $h(x,y)$ and the same template coefficients (22). FD approximations to $(\partial_{xx}u)_{i-1,j}$ and $(\partial_{xx}u)_{i+1,j}$ and the required template coefficients $c_k^{x_{i-1}}$, $i-2 \leqslant k \leqslant i$ and $c_k^{x_{i+1}}$, $i \leqslant k \leqslant i+2$ are derived from (23) and (22) by shifting the index $i$ by $\mp 1$. Substituting Eq. (23) for $(\partial_{xx}u)_{i-1,j}$, $(\partial_{xx}u)_{i,j}$ and $(\partial_{xx}u)_{i+1,j}$ into (24), yields the FD approximation to $(\partial_{xx}h)_{i,j}$

$$
\begin{aligned}
\left(\partial_{xx}h\right)_{i,j} = {} & \left[c_{i-2}^{x_{i-1}} c_{i-1}^{x_i} b_{i-1,j}\right] u_{i-2,j} + \left[c_{i-1}^{x_{i-1}} c_{i-1}^{x_i} b_{i-1,j} + c_{i-1}^{x_i} c_i^{x_i} b_{i,j}\right] u_{i-1,j}\\
& + \left[c_{i-1}^{x_i} c_i^{x_{i-1}} b_{i-1,j} + \left(c_i^{x_i}\right)^2 b_{i,j} + c_i^{x_{i+1}} c_{i+1}^{x_i} b_{i+1,j}\right] u_{i,j} + \left[c_i^{x_i} c_{i+1}^{x_i} b_{i,j} + c_{i+1}^{x_i} c_{i+1}^{x_{i+1}} b_{i+1,j}\right] u_{i+1,j}\\
& + \left[c_{i+1}^{x_i} c_{i+2}^{x_{i+1}} b_{i+1,j}\right] u_{i+2,j}.
\end{aligned}
\tag{25}
$$

With equal grid spacing ($\Delta x_{i-1} = \Delta x_i = \Delta x_{i+1} = \Delta x_{i+2} = \Delta x$) and assuming $b(x,y) = 1$, Eq. (25) becomes

$$
\left(\partial_{xx}h\right)_{i,j} = \frac{u_{i-2,j} - 4u_{i-1,j} + 6u_{i,j} - 4u_{i+1,j} + u_{i+2,j}}{\Delta x^4}.
$$

Similarly (23), $\partial_{yy}u$ is approximated as

$$
\left(\partial_{yy}u\right)_{i,j} = c_{j-1}^{y_j} u_{i,j-1} + c_j^{y_j} u_{i,j} + c_{j+1}^{y_j} u_{i,j+1}, \tag{26}
$$

with

$$
\begin{aligned}
c_{j-1}^{y_j} &= 2/\left(\Delta y_j \Delta y_{j+1} + \Delta y_j^2\right),\\
c_j^{y_j} &= -2/\left(\Delta y_j \Delta y_{j+1}\right),\\
c_{j+1}^{y_j} &= 2/\left(\Delta y_j \Delta y_{j+1} + \Delta y_{j+1}^2\right).
\end{aligned}
\tag{27}
$$

By analogy to (25), the derivative of $\partial_{yy}g$ in the $y$-direction at $y_j$ is

$$
\begin{aligned}
\left(\partial_{yy}g\right)_{i,j} = {} & \left[c_{j-2}^{y_{j-1}} c_{j-1}^{y_j} b_{i,j-1}\right] u_{i,j-2} + \left[c_{j-1}^{y_{j-1}} c_{j-1}^{y_j} b_{i,j-1} + c_{j-1}^{y_j} c_j^{y_j} b_{i,j}\right] u_{i,j-1}\\
& + \left[c_{j-1}^{y_j} c_j^{y_{j-1}} b_{i,j-1} + \left(c_j^{y_j}\right)^2 b_{i,j} + c_j^{y_{j+1}} c_{j+1}^{y_j} b_{i,j+1}\right] u_{i,j} + \left[c_j^{y_j} c_{j+1}^{y_j} b_{i,j} + c_{j+1}^{y_j} c_{j+1}^{y_{j+1}} b_{i,j+1}\right] u_{i,j+1}\\
& + \left[c_{j+1}^{y_j} c_{j+2}^{y_{j+1}} b_{i,j+1}\right] u_{i,j+2}.
\end{aligned}
\tag{28}
$$

The mixed derivative $\partial_{xx}(b\partial_{yy}u)$ in (20) can be discretized using (14). It follows from (26) that the template for $\partial_{yy}u$ taken at grid point $y_j$ is $\Lambda_{2,2}^{y_j} = (c_{j-1}^{y_j}, c_j^{y_j}, c_{j+1}^{y_j})$. Then, from (24), the template for the $x$ derivative at $x_i$ is $\Lambda_{2,2}^{x_i} = (c_{i-1}^{x_i} b_{i-1,j}, c_i^{x_i} b_{i,j}, c_{i+1}^{x_i} b_{i+1,j})$, and that for $\partial_{xx}(b\partial_{yy}u)$ is the tensor product (14)

$$\begin{pmatrix} c_{i-1}^{x_i} b_{i-1,j} & c_i^{x_i} b_{i,j} & c_{i+1}^{x_i} b_{i+1,j} \end{pmatrix} \otimes \begin{pmatrix} c_{j-1}^{y_j} & c_j^{y_j} & c_{j+1}^{y_j} \end{pmatrix} = \begin{pmatrix} c_{i-1}^{x_i} c_{j-1}^{y_j} b_{i-1,j} & c_{i-1}^{x_i} c_j^{y_j} b_{i-1,j} & c_{i-1}^{x_i} c_{j+1}^{y_j} b_{i-1,j} \\ c_i^{x_i} c_{j-1}^{y_j} b_{i,j} & c_i^{x_i} c_j^{y_j} b_{i,j} & c_i^{x_i} c_{j+1}^{y_j} b_{i,j} \\ c_{i+1}^{x_i} c_{j-1}^{y_j} b_{i+1,j} & c_{i+1}^{x_i} c_j^{y_j} b_{i+1,j} & c_{i+1}^{x_i} c_{j+1}^{y_j} b_{i+1,j} \end{pmatrix}. \tag{29}$$

The mixed derivative $\partial_{xx}(b\partial_{yy}u)$ is (8)

$$\begin{aligned} (\partial_{xx} g)_{i,j} = {} & \left[c_{i-1}^{x_i} c_{j-1}^{y_j} b_{i-1,j}\right] u_{i-1,j-1} + \left[c_{i-1}^{x_i} c_j^{y_j} b_{i-1,j}\right] u_{i-1,j} + \left[c_{i-1}^{x_i} c_{j+1}^{y_j} b_{i-1,j}\right] u_{i-1,j+1} + \left[c_i^{x_i} c_{j-1}^{y_j} b_{i,j}\right] u_{i,j-1} \\ & + \left[c_i^{x_i} c_j^{y_j} b_{i,j}\right] u_{i,j} + \left[c_i^{x_i} c_{j+1}^{y_j} b_{i,j}\right] u_{i,j+1} + \left[c_{i+1}^{x_i} c_{j-1}^{y_j} b_{i+1,j}\right] u_{i+1,j-1} + \left[c_{i+1}^{x_i} c_j^{y_j} b_{i+1,j}\right] u_{i+1,j} \\ & + \left[c_{i+1}^{x_i} c_{j+1}^{y_j} b_{i+1,j}\right] u_{i+1,j+1}. \end{aligned} \tag{30}$$

Finally, the mixed derivative $\partial_{yy}(b\partial_{xx}u)$ is determined by analogy to (29) and (30)

$$\begin{aligned} (\partial_{yy} h)_{i,j} = {} & \left[c_{i-1}^{x_i} c_{j-1}^{y_j} b_{i,j-1}\right] u_{i-1,j-1} + \left[c_i^{x_i} c_{j-1}^{y_j} b_{i,j-1}\right] u_{i,j-1} + \left[c_{i+1}^{x_i} c_{j-1}^{y_j} b_{i,j-1}\right] u_{i+1,j-1} + \left[c_{i-1}^{x_i} c_j^{y_j} b_{i,j}\right] u_{i-1,j} \\ & + \left[c_i^{x_i} c_j^{y_j} b_{i,j}\right] u_{i,j} + \left[c_{i+1}^{x_i} c_j^{y_j} b_{i,j}\right] u_{i+1,j} + \left[c_{i-1}^{x_i} c_{j+1}^{y_j} b_{i,j+1}\right] u_{i-1,j+1} + \left[c_i^{x_i} c_{j+1}^{y_j} b_{i,j+1}\right] u_{i,j+1} \\ & + \left[c_{i+1}^{x_i} c_{j+1}^{y_j} b_{i,j+1}\right] u_{i+1,j+1}. \end{aligned} \tag{31}$$

Substituting (25), (28), (30) and (31) in (20) yields the numerical approximation to the Euler–Lagrange equation

$$\begin{aligned} & c_{i,j-2} u_{i,j-2} + c_{i-1,j-1} u_{i-1,j-1} + c_{i,j-1} u_{i,j-1} + c_{i+1,j-1} u_{i+1,j-1} + c_{i-2,j} u_{i-2,j} + c_{i-1,j} u_{i-1,j} + c_{i,j} u_{i,j} \\ & + c_{i+1,j} u_{i+1,j} + c_{i+2,j} u_{i+2,j} + c_{i-1,j+1} u_{i-1,j+1} + c_{i,j+1} u_{i,j+1} + c_{i+1,j+1} u_{i+1,j+1} + c_{i,j+2} u_{i,j+2} = 0 \end{aligned} \tag{32}$$

where

$$\begin{aligned} c_{i,j-2} &= c_{j-2}^{y_{j-1}} c_{j-1}^{y_j} b_{i,j-1}, \\ c_{i-1,j-1} &= c_{i-1}^{x_i} c_{j-1}^{y_j} \left(b_{i,j-1} + b_{i-1,j}\right), \\ c_{i,j-1} &= c_{j-1}^{y_j} \left(\left(c_i^{x_i} + c_{j-1}^{y_{j-1}}\right) b_{i,j-1} + \left(c_i^{x_i} + c_j^{y_j}\right) b_{i,j}\right), \\ c_{i+1,j-1} &= c_{i+1}^{x_i} c_{j-1}^{y_j} \left(b_{i+1,j} + b_{i,j-1}\right), \\ c_{i-2,j} &= c_{i-2}^{x_{i-1}} c_{i-1}^{x_i} b_{i-1,j}, \\ c_{i-1,j} &= c_{i-1}^{x_i} \left(\left(c_{i-1}^{x_{i-1}} + c_j^{y_j}\right) b_{i-1,j} + \left(c_i^{x_i} + c_j^{y_j}\right) b_{i,j}\right), \\ c_{i,j} &= c_{i-1}^{x_i} c_i^{x_{i-1}} b_{i-1,j} + c_i^{x_{i+1}} c_{i+1}^{x_i} b_{i+1,j} + c_{j-1}^{y_j} c_j^{y_{j-1}} b_{i,j-1} + c_j^{y_{j+1}} c_{j+1}^{y_j} b_{i,j+1} + \left(c_i^{x_i} + c_j^{y_j}\right)^2 b_{i,j} + a_{i,j}, \\ c_{i+1,j} &= c_{i+1}^{x_i} \left(\left(c_{i+1}^{x_{i+1}} + c_j^{y_j}\right) b_{i+1,j} + \left(c_i^{x_i} + c_j^{y_j}\right) b_{i,j}\right), \\ c_{i+2,j} &= c_{i+1}^{x_i} c_{i+2}^{x_{i+1}} b_{i+1,j}, \\ c_{i-1,j+1} &= c_{i-1}^{x_i} c_{j+1}^{y_j} \left(b_{i-1,j} + b_{i,j+1}\right), \\ c_{i,j+1} &= c_{j+1}^{y_j} \left(\left(c_i^{x_i} + c_{j+1}^{y_{j+1}}\right) b_{i,j+1} + \left(c_i^{x_i} + c_j^{y_j}\right) b_{i,j}\right), \\ c_{i+1,j+1} &= c_{i+1}^{x_i} c_{j+1}^{y_j} \left(b_{i,j+1} + b_{i+1,j}\right), \\ c_{i,j+2} &= c_{j+1}^{y_j} c_{j+2}^{y_{j+1}} b_{i,j+1}. \end{aligned} \tag{33}$$

Here the known $c_{i,j}$ are functions of the $a$ and $b$ coefficients and a grid spacing. For derivative approximations in the Euler–Lagrange equation (20) with a second-order truncation error, a stencil incorporates 13

grid nodes (Fig. 5). Suppose that the grid points are numbered from 1 to $n_x$ in the $x$-direction and from 1 to $n_y$ in the $y$-direction. The total number of grid points is $n' = n_x \times n_y$. The grid points at the boundaries of a computational domain are assigned values from (3). The condition (4) at the cylindrical boundary of each inclusion is implemented as $u_{ij} = u_0 + sd_{ij}$ for the points $(i,j)$ nearest the circumference both inside and outside the circle. Here, $d_{ij} = r_{ij} - r_0$ is the distance between the point $(i,j)$ and the circumference of the circle and $r_{ij}$ is the distance from that point to the center of an inclusion. As the $u_{i,j}$'s on the boundary of the computational domain and on the boundary of inclusions are known (solid circles in Fig. 6), the number of the interior points, $n$ (open circles in Fig. 6) within the computational domain where variables $u_{i,j}$ have to be determined are less than $n'$. The number of interior points depends on the size of the inclusions in the cluster (Fig. 6). Interior points can be ordered as a vector

$$\left(u_{1,1},\ldots,u_{n_x^1,1},u_{1,2},\ldots,u_{n_x^2,2},\ldots,u_{1,n_y},\ldots,u_{n_x^{n_y},n_y}\right),$$

where $n = \sum_{j=1}^{n_y} n_x^j$, $n_x^j$ is the number of interior points in the $x$-direction for the $j$th row. We write the system of $n$ equations, corresponding to $n$ interior points in matrix form, with the known boundary values on the right-hand side

$$\begin{bmatrix}
\mathbf{c}_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
c_{11} & \mathbf{c}_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{11} & c_{21} & \cdots & \mathbf{c}_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & \mathbf{c}_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & \mathbf{c}_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & \mathbf{c}_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & \mathbf{c}_{1n_y} & c_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & \mathbf{c}_{2n_y} & \cdots & c_{n_x^{n_y} n_y} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{11} & c_{21} & \cdots & c_{n_x^1 1} & c_{12} & c_{22} & \cdots & c_{n_x^2 2} & \cdots & c_{1n_y} & c_{2n_y} & \cdots & \mathbf{c}_{n_x^{n_y} n_y}
\end{bmatrix}
\begin{bmatrix} u_{11} \\ u_{21} \\ \cdots \\ u_{n_x^1 1} \\ u_{12} \\ u_{21} \\ \cdots \\ u_{n_x^2 2} \\ \cdots \\ u_{1n_y} \\ u_{2n_y} \\ \cdots \\ u_{n_x^{n_y} n_y} \end{bmatrix}
=
\begin{bmatrix} d_{11} \\ d_{21} \\ \cdots \\ d_{n_x^1 1} \\ d_{12} \\ d_{21} \\ \cdots \\ d_{n_x^2 2} \\ \cdots \\ d_{1n_y} \\ d_{2n_y} \\ \cdots \\ d_{n_x^{n_y} n_y} \end{bmatrix}. \tag{34}$$

This system can be expressed compactly as $C \cdot U = D$, where $C$ is the known coefficient matrix, $U$ the solution vector and $D$ determined by the known boundary values. The linear system (34) is sparse, symmetric and positive-definite. The coefficient matrix is $n \times n$ and each matrix row contains all $n$ interior points. In our membrane deformation problem the largest grid used was $640 \times 640$ points, with a coefficient matrix of $\sim 10^{11}$ elements. It is a large sparse matrix with an extremely large number of zeros since only nodes lying within the stencil have nonzero couplings. Nonzero elements for each row are found by centering the stencil (Fig. 5) on the elements in bold in (34). Nonzero elements lie in a band along the matrix diagonal.

The key to efficiency is to store and operate with only nonzero matrix entries. Approaches to efficient solution of linear algebraic equations (34) fall into two classes. The first involves algorithms that directly solve the boundary value problem after a finite number of steps while the second involves an initial "guess" which is then improved by a finite series of iterations. Direct methods involve a form of Gaussian elimination or closely related procedures such as LU decomposition [34]. There is a large literature devoted to sparse solvers; they have been extensively developed and are very robust, reliable and efficient for a wide range of practical problems. They do not require an initial solution estimate and typically yield high accuracy solutions. For large problems, iterative methods are generally more efficient than direct methods as

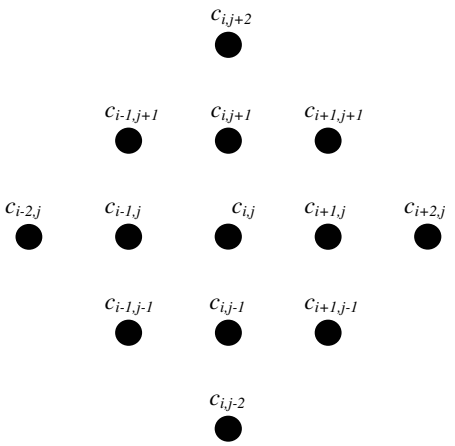


Fig. 5. The thirteen-point stencil demonstrating the finite difference scheme for the Euler–Lagrange equation.

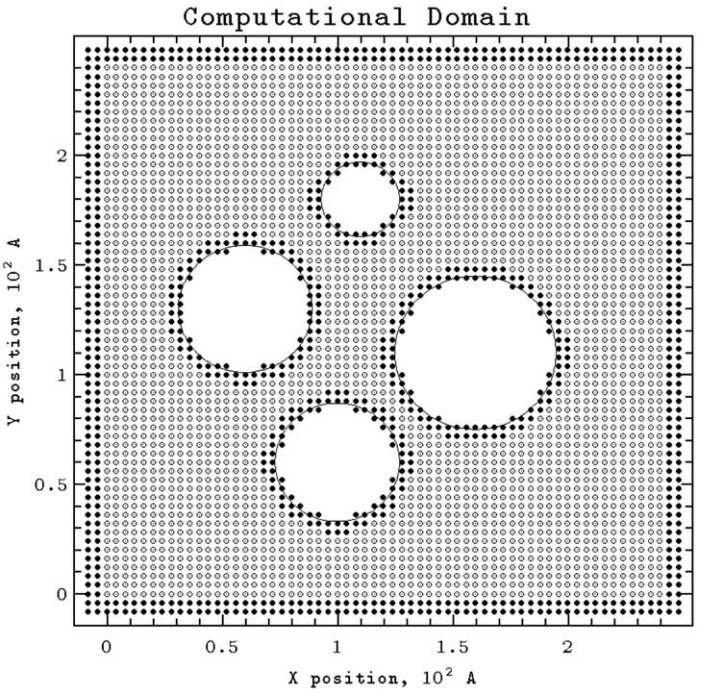


Fig. 6. Schematic illustration of the computational domain with four inclusions of different radii. For illustrative purpose the grid size is 4 Å. Interior grid points are open circles. Two layers of solid circles are the boundary points on the edge of the computational domain and on the boundary of each inclusion.

they benefit from matrix sparseness. However, these approaches often depend on special properties, such as the matrix being symmetric positive definite, and badly conditioned systems converge slowly. In solving (34), routines from the NAG Libraries [35], a product of the Numerical Algorithms Group Ltd., performing a sparse LU factorization were used. The NAG's sparse matrix routines are extensions of the original

Harwell Subroutine Library [36]. Eq. (34) was factorized using the f01brf subroutine, and this factorization solved with the f04axf subroutine. However, this direct sparse solver was inefficient for our extremely large sparse matrix because of computational and storage requirements. Therefore, we used it for evaluating the solution vector, $U$, using a rough grid with spacing of 1–2 Å. This initial solution was used iteratively with mesh spacings of 0.4–0.5 Å and successively improved until the desired solution accuracy was attained. Iteration, done with the preconditioned biconjugate gradient method [34], had the advantage that they work directly on the grid without needing extra storage.

Previous work showed that for one inclusion the total deformation free energy can be expressed in terms of a linear (Hookean) spring model where bilayer material constants are combined into a single spring constant [29,37]. The model was generalized to treat membrane-mediated interaction between a set of inclusions [24–26]; the Hookean relationship is general, reflecting the linearity of Eqs. (2)–(4) [38]. Thus the elastic energy (1) is a quadratic function of the boundary parameters

$$F = \sum_{\kappa=1}^{N} \sum_{\tau=\kappa}^{N} k_{\kappa\tau} \alpha_{\kappa} \beta_{\tau}, \tag{35}$$

where $k_{\kappa\tau}$ are unknown effective spring constants, and the indices $\kappa$ and $\tau$ enumerate the inclusions. The additional summation is performed over the repeated indexes $\alpha$, $\beta$; these symbolize the boundary parameters, $\alpha, \beta = u, s$, etc. (e.g. the parameter accounting for azimuthal variation of the slope [25]). The coefficients $k_{\kappa\tau}$ are independent of both $u$ and $s$ ($k_{\kappa\kappa}$ corresponds to elastic "self-energy" due to deformation of the membrane surrounding the $\kappa$th inclusion; $k_{\kappa\tau}$ describes coupling between inclusions $\kappa$ and $\tau$ propagated via membrane deformation).To demonstrate this CHO approach, consider seven inclusions forming a regular, centered, symmetric hexagon with "null constraint" boundary condition $s_{\kappa} = 0$ for all inclusions (Fig.7). For any inclusion geometry, the elastic free energy (35) is a quadratic

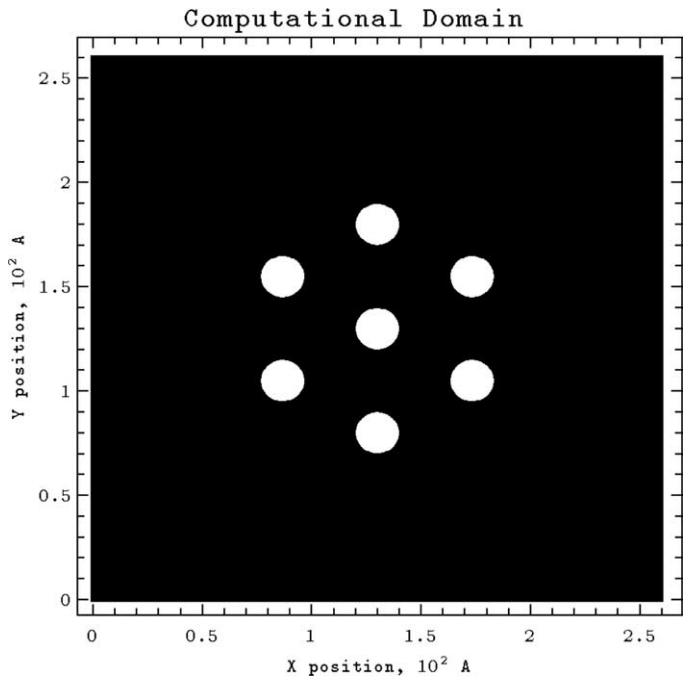


Fig. 7. Cluster of seven inclusions, with one at the center and six at the vertices of a regular hexagon. The fine mesh spacing with 0.4 Å results in a black-colored computational domain.

function of the boundary displacements $u_\kappa$. The number of independent spring constants in (35) is six, $k_{11}$, $k_{12}$, $k_{13}$, $k_{14}$, $k_{77}$ and $k_{17}$ (with 7 enumerating the central inclusion). To find effective elastic constants for a particular inclusion configuration (Fig.7), the elastic problem (2) with the boundary conditions (3) and (4) must be solved numerically for six linearly independent sets of $u_\kappa$. The six spring constants, $k_{\kappa\tau}$, are determined by substituting the six energies calculated from (1) on the left-hand side of (35). The same FD-CHO procedure is used to determine elastic constants for different cluster geometries (maintaining cluster symmetry). With these constants in hand, (35) analytically determines elastic energies.The FD-CHO approach dramatically reduces computational complexity, as it requires direct numerical solution of the boundary value problem (1)–(4) for a limited set of displacement fluctuations $u_\kappa$ at each inter-inclusion separation.

## 6. FD approximation to heat and magnetic field diffusion equations

FD methods start by discretizing space and time so that there are a specified number of points in the space domain and a specified number of time levels at which the redistributions of the temperature and magnetic field due to diffusion are calculated. We use a grid of gradually varying cell size by imposing unequal grid spacings $\Delta r_i = r_{i+1/2} - r_{i-1/2}$ and $\Delta z_j = z_{i+1/2} - z_{i-1/2}$ in the $r$ and $z$ directions, respectively. The subscripts $i+1/2$ and $j+1/2$ refer to quantities defined on the cell interfaces $r_{i+1/2}$ and $z_{j+1/2}$. Cell centers $r_i = (r_{i-1/2} + r_{i+1/2})/2$ and $z_j = (z_{i-1/2} + z_{i+1/2})/2$ are specified at positions $(i,j)$. We use standard notation for evaluating functions $T^n_{i,j}$ and $B^n_{i,j}$ defined at cell centers $(i,j)$ and time level $n$. We assume time spacings $t^n$ with intervals $\Delta t^n = t^{n+1} - t^n$. Given a grid, we use finite differencing (13) to approximate second-order spatial derivatives in (6) and (7) at our grid points. We approximate the spatial derivatives at each point $(i,j)$ using centered differences with truncation error of order $o = 2$.

The spatial derivative of temperature in the $r$-direction at $r_i$ can be approximated as (8)

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(rK\frac{\partial T}{\partial r}\right)\right]_{i,j} = c^{r_i}_{i-1,j}T^{n+1}_{i-1,j} + c^{r_i}_{i,j}T^{n+1}_{i,j} + c^{r_i}_{i+1,j}T^{n+1}_{i+1,j}, \tag{36}$$

where

$$c^{r_i}_{i-1,j} = \frac{2K_{i-1/2,j}r_{i-1/2}}{r_i\Delta r_i(\Delta r_{i-1} + \Delta r_i)}, \quad c^{r_i}_{i+1,j} = \frac{2K_{i+1/2,j}r_{i+1/2}}{r_i\Delta r_i(\Delta r_{i+1} + \Delta r_i)},$$
$$c^{r_i}_{i,j} = -\frac{2K_{i+1/2,j}r_{i+1/2}(\Delta r_{i-1} + \Delta r_i) + 2K_{i-1/2,j}r_{i-1/2}(\Delta r_{i+1} + \Delta r_i)}{r_i\Delta r_i(\Delta r_{i-1} + \Delta r_i)(\Delta r_{i+1} + \Delta r_i)}.$$

The terms $K_{i-1/2,j}$ and $K_{i+1/2,j}$ on the cell interfaces are determined from values at the grid points, by linear interpolation between adjacent grid points

$$K_{i+1/2,j} = \frac{\Delta r_i K_{i+1,j} + \Delta r_{i+1}K_{i,j}}{\Delta r_{i+1} + \Delta r_i}.$$

The FD approximation (36) is formulated with the unknown temperatures, $T^{n+1}$, at the new time level, $t^{n+1}$, in order to use large time steps constrained by the physics, not the numerics. If, instead, we use $T^n$ (the explicit scheme), the maximum allowable time step is severely limited to $\Delta t < \min(\Delta r^2, \Delta z^2)/(2K)$, the diffusion time across a cell. The number of time steps required for evolution across characteristic spatial scales are prohibitively large. The spatial temperature derivative in the $z$-direction at $z_j$ is (8)

$$\left[\frac{\partial}{\partial z}\left(K\frac{\partial T}{\partial z}\right)\right]_{i,j} = c^{z_j}_{i,j-1}T^{n+1}_{i,j-1} + c^{z_j}_{i,j}T^{n+1}_{i,j} + c^{z_j}_{i,j+1}T^{n+1}_{i,j+1}, \tag{37}$$

where

$$c^{z_j}_{i,j-1} = \frac{2K_{i,j-1/2}}{\Delta z_j(\Delta z_{j-1}+\Delta z_j)}, \quad c^{z_j}_{i,j+1} = \frac{2K_{i,j+1/2}}{\Delta z_j(\Delta z_{j+1}+\Delta z_j)},$$
$$c^{z_j}_{i,j} = -\frac{2K_{i,j+1/2}(\Delta z_{j-1}+\Delta z_j)+2K_{i,j-1/2}(\Delta z_{j+1}+\Delta z_j)}{\Delta z_j(\Delta z_{j-1}+\Delta z_j)(\Delta z_{j+1}+\Delta z_j)}.$$

Given an implicit FD discretization of the diffusion term in (6) we still must approximate the time derivative. Using a forward time step the FD approximation to the time derivative in (6) is

$$\left[\frac{\partial \rho c_p T}{\partial t}\right]_{i,j} = \frac{\rho^n_{i,j}(c_p)^n_{i,j}T^{n+1}_{i,j} - \rho^n_{i,j}(c_p)^n_{i,j}T^n_{i,j}}{\Delta t^n}. \tag{38}$$

Plasma properties such as density and heat capacity are assumed constant during each time step. Substituting (36)–(38) into (6) and rearranging gives us the discretized FD form for heat diffusion

$$c_{i-1,j}T^{n+1}_{i-1,j} + c_{i,j-1}T^{n+1}_{i,j-1} + c_{i,j}T^{n+1}_{i,j} + c_{i+1,j}T^{n+1}_{i+1,j} + c_{i,j+1}T^{n+1}_{i,j+1} = d_{i,j}, \tag{39}$$

where the resulting coefficients

$$c_{i-1,j} = c^{r_i}_{i-1,j}, \quad c_{i,j-1} = c^{z_j}_{i,j-1}, \quad c_{i+1,j} = c^{r_i}_{i+1,j}, \quad c_{i,j+1} = c^{z_j}_{i,j+1}, \quad c_{i,j} = c^{r_i}_{i,j} + c^{z_j}_{i,j} - \frac{\rho^n_{i,j}(c_p)^n_{i,j}}{\Delta t^n},$$
$$d_{i,j} = -\frac{\rho^n_{i,j}(c_p)^n_{i,j}T^n_{i,j}}{\Delta t^n} \tag{40}$$

are expressed in terms of grid spacing, thermal conductivity, heat capacity, density and temperature of the plasma at the previous time level. Similar to (6), we generalize this fully implicit scheme to describe the diffusion equation for the magnetic field (7). Its FD approximation is

$$c_{i-1,j}B^{n+1}_{i-1,j} + c_{i,j-1}B^{n+1}_{i,j-1} + c_{i,j}B^{n+1}_{i,j} + c_{i+1,j}B^{n+1}_{i+1,j} + c_{i,j+1}B^{n+1}_{i,j+1} = d_{i,j}, \tag{41}$$

where the known coefficients $c_{i,j}$ and $d_{i,j}$ are functions of grid spacing, magnetic diffusivity and magnetic field at the previous time level. The spatial derivative of the magnetic field in the $r$-direction in Eq. (7) differs from that in Eq. (6 ). It becomes singular at $r = 0$. The template coefficients for the $r$-derivative in (7) has the form

$$c^{r_i}_{i-1,j} = \frac{2\eta_{i-1/2,j}r_{i-1}}{r_{i-1/2}\Delta r_i(\Delta r_{i-1}+\Delta r_i)}, \quad c^{r_i}_{i+1,j} = \frac{2\eta_{i+1/2,j}r_{i+1}}{r_{i+1/2}\Delta r_i(\Delta r_{i+1}+\Delta r_i)},$$
$$c^{r_i}_{i,j} = -\left[\frac{2\eta_{i+1/2,j}r_i}{r_{i+1/2}\Delta r_i(\Delta r_{i+1}+\Delta r_i)} + \frac{2\eta_{i-1/2,j}r_i}{r_{i-1/2}\Delta r_i(\Delta r_{i-1}+\Delta r_i)}\right].$$

For the first cell (at $r = 0$) the coefficients are

$$c^{r_i}_{0,j} = 0, \quad c^{r_i}_{2,j} = \frac{2\eta_{3/2,j}r_2}{r_{3/2}\Delta r_1(\Delta r_2+\Delta r_1)},$$
$$c^{r_i}_{1,j} = -\left[\frac{2\eta_{3/2,j}r_1}{r_{3/2}\Delta r_1(\Delta r_2+\Delta r_1)} + \frac{4\eta_{1/2,j}}{\Delta r_1^2}\right].$$

The template coefficients for the $z$-derivative in (7) have the same form as those in (37). For the given initial and boundary conditions (see Section 3) at each time step the systems of algebraic equations (39) and (41) are solved for all nodes $(i,j)$ to find the $T^{n+1}_{i,j}$ and $B^{n+1}_{i,j}$ at the next time step. This contrasts with explicit time discretization where temperature and magnetic field at the next time step are found without solving an

algebraic system. The advantage of the implicit method is that it is unconditionally stable, requiring no stability condition on the time step. However, for accuracy, the equations must be advanced by time steps one or two orders of magnitude less than the physical time scale. A characteristic feature of the fully implicit method is that details of small-scale evolution of temperature and magnetic field from their initial conditions are smeared out on the large implicit time steps. However, the correct steady-state solution is obtained. For an enormous time step $\Delta t \to \infty$, $d_{i,j} \to 0$ in (40). The implicit method actually solves the steady state 2D heat and magnetic field problems at each time step.

Implicit schemes require solving a set of simultaneous linear equations for $T_{i,j}^{n+1}$ and $B_{i,j}^{n+1}$ at each time step. Eqs. (39) and (41) are expressed in matrix form (Section 5) as $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ is the pentadiagonal coefficient matrix with two outer diagonals widely separated from three inner diagonals, $\boldsymbol{x}$ corresponds to the array of temperature or magnetic field values at time step $n+1$ and $\boldsymbol{b}$ are known values at time $n$. Values of temperature or magnetic field from all boundaries in the computational domain determine the right-hand side vector (i.e. $\boldsymbol{b}$) and direct sparse solvers, as described in Section 5, solve the linear algebraic system. Matrix coefficients and the vector $b$ (boundary conditions) are updated after each time step.

## 7. Results

### 7.1. Membrane-mediated interactions in protein aggregates

The PAMEMD[1] (Protein Aggregation Mediated by Elastic Membrane Deformations) code based on the FD-CHO algorithm was developed in Fortran-95 programming language. It determines numerical solutions of the Euler–Lagrange equation (15) in 2D Cartesian coordinates for arbitrarily configured inclusion clusters and the effective spring constants needed in (35) to analytically determine elastic deformation free energies. We first tested the method for one cylindrical inclusion embedded in a glycerol monooleate (GMO) membrane, where the problem can be solved analytically [7]. Exact membrane deformation profiles for slopes $s = 0$ and $s = -0.45$ ($s_{\min}$ for GMO) are compared in Fig. 8(a) with numerical results. A grid spacing of 0.5 Å (near our limiting computational capability) accurately reproduces analytical results (Fig. 8(a)). However, the deformation free energy is very sensitive to the choice of grid spacing as illustrated in Fig. 8(b), comparing profiles as a function of the contact slope. Numerical calculations, carried out at three grid spacings (0.2, 0.5 and 1 Å) equal in both $x$ and $y$ directions, show that regardless of mesh size, numerical and analytical results agree especially well (to within 2%) for the physically most interesting slope ranges (−0.5 to 0). The error is greater for large negative slopes, reaching ~20% for $s = -1$ and a 0.5 Å mesh. For such unphysical slopes the distortion surface is very steep; it drops abruptly near the inclusion, in which case the deformation surface displays pronounced non-monotonic behavior with a deep well near the inclusion and a pronounced peak ~20–30 Å away from the inclusion [7]. In that region (~20–30 Å) membrane thickness exceeds the unperturbed value $h_0$. Clearly, for large negative slopes a fine grid spacing is needed to reproduce complex distortion surface behavior. Two main error sources contribute to the energy integral. First, the distortion surface is calculated at a limited number of points; $u(x, y)$ defined on a mesh is approximate and its precision grid size-dependent. The bending energy (1), determined by the second derivatives of $u$, is especially sensitive. Second, we approximate inclusion boundaries by zigzag strips of grid points (Fig. 6). In this region, surface distortion is maximal and its variation greatest; thus changing boundary shape may introduce further error. Both errors decrease with grid refinement. The numerical scheme (32,33) with truncation error $o = 2$ is theoretically expected to be globally second-order accurate.

[1] The PAMEMD code and program manual are freely available to download from http://people.brandeis.edu/~gennady/pamemd.html.

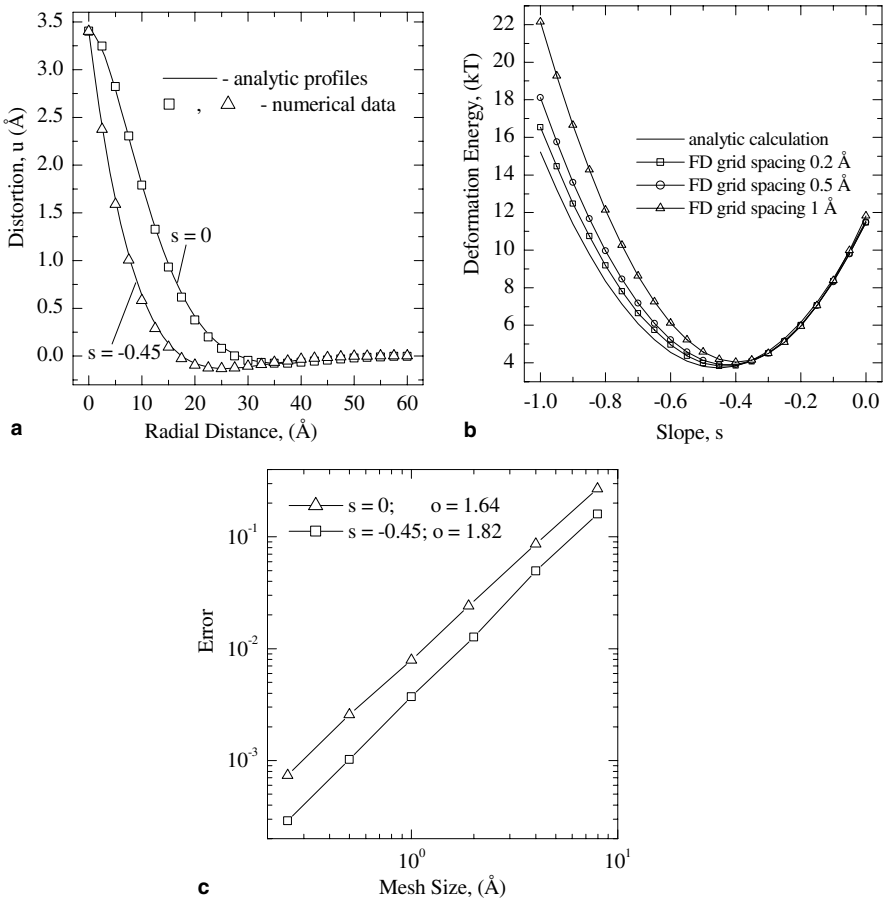


Fig. 8. Comparison between (a) the analytic and numerical membrane deformation profiles, (b) deformation free energy profiles and (c) the evolution of the global error as a function of grid spacing. In (c) the global convergence order $o$ is included in the figure. The parameters of a GMO membrane are: $B = 5 \times 10^{-8}$dyn Å $^{-2}$, $K = 10^{-6}$ dyn, $h_0 = 28.5$ Å. The hydrophobic length of an inclusion is $\delta = 21.7$ Å, and its radius is $r_0 = 10$ Å. The displacement of the membrane from its unperturbed state at the membrane-inclusion contact surface is $u_0 = 3.4$ Å.

However, the aforementioned factors (boundary conditions and grid spacing) affect the global order. To specifically demonstrate the convergence properties of our numerical solution, we define the error at each grid point $(i,j)$ as the norm of the difference between numerical and analytic solutions

$$\Delta\varepsilon_{ij} = |u^h_{ij} - u^a_{ij}|,$$

where $u^a_{ij}$ is the analytic solution and $u^h_{ij}$ is the numerical result for a prescribed isotropic grid spacing $h$. The global error is then found by averaging norms

$$\varepsilon = \frac{1}{N}\sum_{i,j}\Delta\varepsilon_{ij};$$

$N$ is the number of grid points at which the $\Delta\varepsilon_{ij}$ are evaluated. To estimate the order of the numerical scheme, we solved the problem of one cylindrical inclusion analytically [7] and on grids of increasing resolution. The dependence of the global error on grid resolution is illustrated in Fig. 8(c) for slopes $s = 0$ and

$s = -0.45$, where the convergence order $o$ is the slope of the curve of $\log(\varepsilon)$ versus $\log(h)$. Steeper slopes indicate faster convergence. As seen from Fig. 8(c), $o$ is approximately 1.64 for $s = 0$ and 1.82 for $s = -0.45$, respectively. Global convergence is affected by the boundary condition (slope of $s$) on the inclusion boundary. It is known that the order of accuracy of the boundary conditions can be one order lower than the local truncation error of an FD scheme without reducing overall global accuracy. However, the global accuracy order can be, in general, one degree lower than the local discretization order at one stencil due to error propagation from other stencils. This limiting factor probably accounts for the global convergence order being less than 2.

We use the FD-CHO method to study: (i) stabilization of the gA ion channel [39] due to membrane-mediated interactions between gA channels in a cluster [24] and (ii) to examine the effects of anisotropic membrane slope relaxation on the channel interaction energy [25]. Here, we briefly highlight our major results. In [24], we considered representative clusters, explicitly accounting for possible fluctuations of the hydrophobic length of a selected channel in the cluster, emphasizing how neighboring channels influence its stability. Clustering, which affects length fluctuations of the selected gA channel, can increase gA lifetimes by orders of magnitude, an effect that is more pronounced for a channel with many near neighbors. In clusters, gA channels are better adjusted to the collectively deformed membrane than the isolated channel was to the original membrane; by thinning the membrane immediately surrounding a selected channel, its neighbors decrease the elastic force tending to separate gA monomers (the channel is a dimer), thus stabilizing the channel. Recent experimental observation of significant stabilization of so-called ''double-barreled'' and ''tandem'' gA channels (up to 100-fold increases in channel lifetimes) [40,41] can be reasonably ascribed to membrane-mediated elastic interactions. A similar experimental study [42] confirms that formation of tandem channels is strongly favored in thicker and stiffer membranes. In a tandem channel, the number of inter-channel hydrogen bonds is doubled. Our theoretical analysis [24] interprets this as reflecting the weaker elastic influence of the collectively deformed membrane on the tandem channel as compared to that on two single channels. In [25], we analyzed the complex boundary conditions that permit anisotropic relaxation of the contact slope along inclusion contours. We found that anisotropic angular variation of the contact slope crucially affects the interaction energy, leading to a short-range attraction between two inclusions, while conventional isotropic boundary conditions result in their strong repulsion. In a multi-inclusion cluster, this attraction is further enhanced due to non-pairwise interactions, a result valid regardless of whether the membrane is treated as uniform or non-uniform [26]. In addition, the non-uniform approach [26], assuming local perturbation of membrane elastic moduli in the vicinity of the inclusion and a contact slope determined by energy minimization ($s = s_{\min}$), yields results qualitatively identical to that from the (uniform) conventional model based on the ''null constraint'' for the contact slope ($s = 0$).

We showed that the FD-CHO algorithm is a practicable way to treat membrane-mediated interactions between inclusions in aggregates. Our approach can be extended in numerous ways, including consideration of non-cylindrical inclusions, incorporation of specific molecular degrees of freedom such as acyl chain tilt and stretching, accounting for spontaneous monolayer curvature and treating systems assembled from subunits. The treatment of non-uniformity [25] was only preliminary as we used a trial function approach to describe the angular dependence of the contact slope for interacting inclusions and assumed the influence of inclusions on elastic moduli is additive. Further development is needed to consistently describe membrane relaxation in contact with inclusions and for self-consistent treatment of the combined influence of inclusions on membrane elastic constants. In particular, the description of the spatial variation of inter-inclusion elastic moduli should satisfy a free energy minimization principle.

### 7.2. Heat and magnetic field diffusion in plasmas

Validity of the fully implicit scheme for the heat diffusion equation (39) has been established by solving one-dimensional test cases. Numerical results are compared with those from analytical solution [43].

*Case 1.* This one-dimensional problem treats the temperature front moving along the $z$-axis in a semi-infinite slab. The heat capacity and density of slab is taken to be unity and the thermal conductivity is a power function of temperature $K = K_0 T^{\alpha}$. The heat conduction equation (6) reduces to

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z} K_0 T^{\alpha} \frac{\partial T}{\partial z} \quad \text{for } z > 0, \quad t > 0. \tag{42}$$

The slab temperature is initially $T = 0$, and at $t > 0$ a semi-infinite slab is exposed to the initial and boundary conditions

$$T(0,t) = \left[\frac{\alpha D}{K_0}(z_1 + Dt)\right]^{1/\alpha}, \quad t > 0,$$

$$T(z,0) = \begin{cases} \left[\frac{\alpha D}{K_0}(z_1 - z)\right]^{1/\alpha}, & 0 < z \leqslant z_1, \\ 0, & z > z_1, \end{cases}$$

with the parameter set $\alpha = 2$, $K_0 = 0.5$, $z_1 = 0$, $D = 5$. Samarskii and Popov [43] solved this problem analytically where the resulting temperature front propagates through a cold medium with the speed $D$

$$T(z,t) = \begin{cases} \left[\frac{\alpha D}{K_0}(Dt + z_1 - z)\right]^{1/\alpha}, & 0 < z \leqslant (z_1 + Dt), \\ 0, & z > (z_1 + Dt). \end{cases}$$

Test calculations were performed utilizing uniform and essentially non-uniform grids. Fig. 9(a) illustrates the temperature profiles along the slab at time $t = 0.1$ for a variety of time step sizes on a uniform grid with $\Delta z = 10^{-2}$. Here, numerical results (open symbols) are compared with the analytical solution (solid curve) of Samarskii and Popov [43]. The temperature front is sharp and the speed of the thermal wave is constant due to the nonlinear thermal conductivity. The results of the fully implicit scheme agree well with the exact solution for a wide range of time steps. For time steps $\Delta t \leqslant 10^{-4}$, the implicit numerical method and the analytical results are in full agreement. The problem (42) was also solved with the Forward Time Centered Space (FTCS) scheme [44,45]. Euler's FTCS method is the simplest way to explicitly solve initial value problems. Explicit time steps were limited to $\Delta t \leqslant 10^{-6}$; larger time steps led to substantial spurious oscillations. Thus, there was a two orders of magnitude difference between implicit

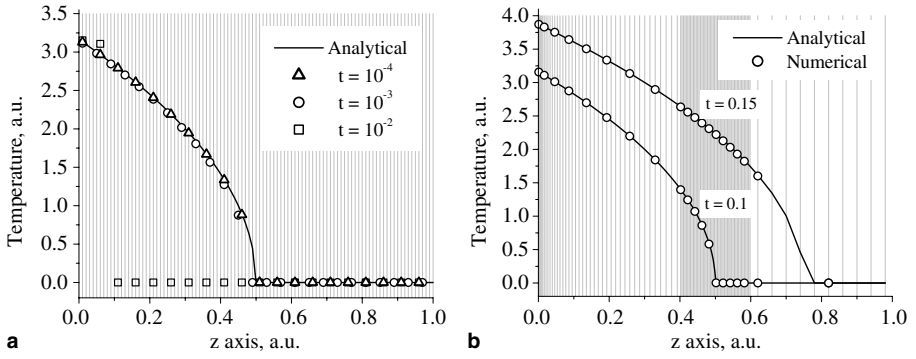


Fig. 9. Comparison of numerical and analytical solution of the one-dimensional problem of the temperature front propagating along the $z$-axis in a semi-infinite slab. (a) Comparison of analytical (solid curve) and numerical (open symbols) temperature profiles at time $t = 0.1$ on a uniform grid with $\Delta z = 10^{-2}$ for a variety of time step sizes. (b) The comparison of the numerical solution (open symbols) to the analytical solution (solid curves) at $t = 0.1$ and $t = 0.15$ on a non-uniform grid. The time step is $\Delta t = 10^{-4}$. In both figures, the grid is shown with light gray vertical lines.

and explicit time steps. The fully implicit algorithm (39) is also numerically stable on essentially non-uniform grids. Propagation of the temperature front along the slab with a non-uniform grid is illustrated in Fig. 9(b). Here results of the implicit method at $t = 0.1$ and $t = 0.15$ are compared with analytical results [43] and calculations performed with time step $\Delta t = 10^{-4}$. The implicit method agrees beautifully with the analytical solution.

A common question in numerical solutions is estimating the order of the global error term. Here, our focus is on how the global error depends on grid spacing for various fixed times. In order to investigate the method's spatial order with increasing refinement of the grid, we consider a 2D cylindrical test case. A cylinder 0.5 cm in diameter and 1 cm in height is heated uniformly on one end as shown in Fig. 10(a). The normal derivative of temperature is zero on all other boundary surfaces. The temperature front moves along the cylinder and for each $z$ the temperature is constant in the $r$-direction. The resulting analytic solution is the same as that for a semi-infinite slab. In this example, an estimate of $o$, the order of the method, is computed as described in the previous section. The global error was calculated at the times $10^{-2}$, $3\times10^{-2}$ and $5\times10^{-2}$ s for grid spacings of 0.005, 0.01 and 0.02 cm. The evolution of the global error as a function of the mesh size is plotted in Fig. 10(b) where the convergence order is about 1.48.

*Case 2*. Here, we consider the evolution of a "stopped" thermal wave in one-dimension. Such a temperature wave is a solution of (42) with the following initial and boundary conditions:

$$T(0,t) = \left[\frac{\alpha z_1}{2K_0(\alpha+2)(C-t)}\right]^{1/\alpha}, \quad 0 < t < C.$$

$$T(z,0) = \begin{cases} \left[\frac{\alpha(z_1-z)^2}{2K_0(\alpha+2)C}\right]^{1/\alpha}, & 0 < z \leqslant z_1, \\ 0, & z > z_1. \end{cases}$$

This problem also has an analytical solution [43] and the temperature front evolves as

$$T(z,t) = \begin{cases} \left[\frac{\alpha(z_1-z)^2}{2K_0(\alpha+2)(C-t)}\right]^{1/\alpha}, & 0 < z \leqslant z_1, \\ 0, & z > z_1. \end{cases}$$

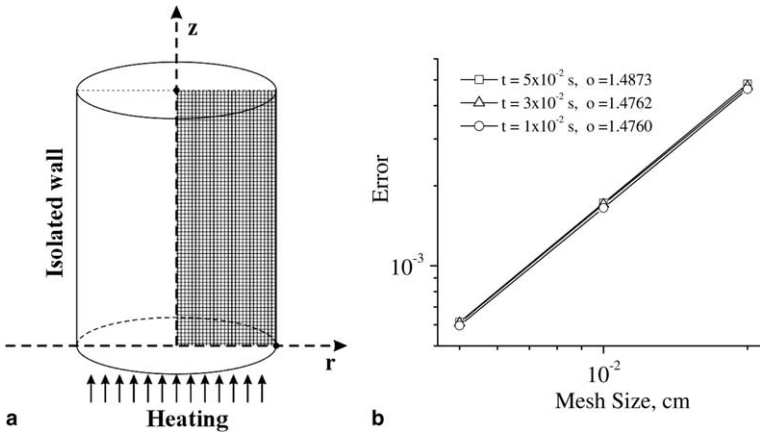


Fig. 10. The two-dimensional problem of: (a) the temperature front propagating along the heat-isolated cylinder and (b) the dependence of the global error on mesh size at three fixed times. The global convergence order is about 1.48.

Fig. 11 shows temperature profiles along the slab at times $t = 0.1$, $t = 0.105$ and $t = 0.112$. At $t > 0$, the left wall temperature is increased. The temperature remains zero in the region $z > z_1$ in spite of instantaneous heat release on the left boundary. Test calculations were performed with a uniform spatial grid of $\Delta z = 10^{-2}$ and time step $\Delta t = 10^{-4}$ using the parameters set $\alpha = 2$, $K_0 = 0.5$, $z_1 = 0.5$, $C = 0.1125$. Fig. 11 shows excellent agreement between numerical and analytical results [43].

These simple test cases demonstrate the stability and accuracy of the fully implicit method (39) in a cold medium. However, we want to validate the fully implicit scheme for practical applications to real plasma devices. Therefore, we modeled the physical processes of heat and magnetic field diffusion in the DPF device [46,47]. It is noteworthy that the DPF pinch discharge has practical implications, as a light source for extreme ultraviolet (EUV) lithography, a promising new technology for producing microchips. EUV lithography based on a xenon pinch plasma requires high radiation intensities at shorter wavelengths, ~13.5 nm, thus enabling high resolution printing of smaller circuit features. The power requirement for EUV lithography (~115 W) obliges source developers to have a better understanding of plasma behavior in real plasma devices, especially near the electrodes [48]. Realistically describing plasma behavior entails developing theoretical approaches that account for heat and magnetic field diffusion [32,49–51]. Heat and magnetic field redistribution was studied in two ways. First, we explicitly treated heat and magnetic field diffusion within the whole MHD system (5) solved by the Total Variation Diminishing method in Lax–Friedrich formulation [52]. Second, these physical processes were decoupled from (5) using the splitting algorithm and treated independently on the basis of the fully implicit scheme. The two modeling techniques agree well. We compare the explicit and implicit approaches for a discharge plasma device, showing that the implicit scheme gives results comparable to the explicit approach while permitting time steps 100 times those of the explicit method. Fig. 12 plots temperature and magnetic field isolines around the device electrode at $t = 200$ ns after the discharge started. An electric current flowing through an external circuit generates a magnetic field. The thermal sources of energy in the plasma are Joule heating and efficient plasma compression by the magnetic field. The temperature distribution is highly non-uniform with the region near the electrodes a hot spot of ~15 eV (see Fig. 12(a)). Higher temperatures near electrodes reflect current spreading in the inter-electrode space. Heat diffusion to the plasma periphery lowers the temperature in this hot region. A region of magnetic field diffusion near the electrode, shown in Fig. 12(b), is also where the current density is greatest. The magnetic field evolves due to resistive diffusion and convection. A diffuse volume of magnetized plasma forms near the electrodes (Fig. 12(b)). The magnitude of the magnetic

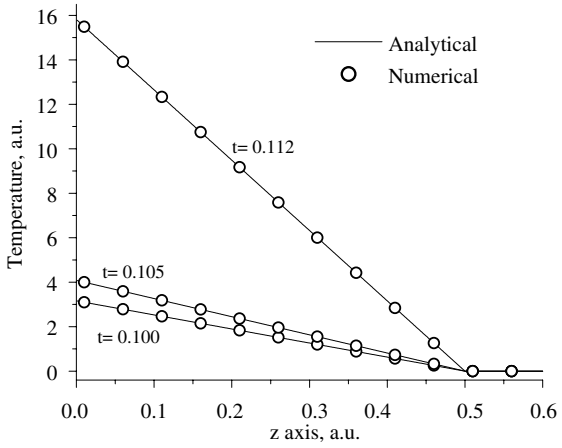


Fig. 11. Comparison of "stopped" thermal wave using the fully implicit scheme (open circles) and analytical predictions (solid curves) [43] for different times (0.1, 0.105 and 0.112) on a uniform grid with $\Delta z = 10^{-2}$. The time step is $\Delta t = 10^{-4}$.

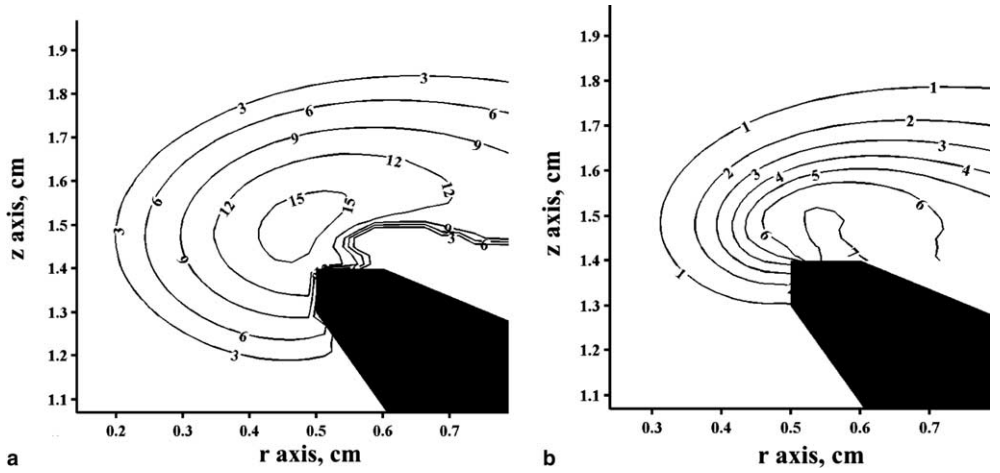


Fig. 12. Isolines of (a) temperature and (b) magnetic field around electrode at time 200 ns. In (a) the contour labels refer to temperature in eV. In (b) they refer to the magnetic field in kG.

field drops 7-fold at the edge of the plasma region. Radial profiles of temperature and magnetic field determined by the explicit and implicit approaches at the monitor point $z = 1.5$ cm are compared in Fig. 13. The calculations were performed with $\Delta t = 5$ ps and $\Delta t = 5 \times 10^{-2}$ ps for the implicit and explicit methods, respectively. While providing accuracy similar to the explicit method, the independent implicit scheme has a number of advantages compared with methods directly incorporated into the MHD system (5). The most crucial feature is the use of large time steps. In calculations of heat transfer through rarefied plasmas, the explicit method is highly unstable. The size of time step is greatly restricted due to heat transfer in areas of low plasma density, typically the area behind the magnetic "snowplow." This can be physically understood as reflecting the collisional mechanism of thermal conductivity. If energy exchange between computational cells occurs in one time step, then the explicit scheme performs well. However, the decrease of the plasma density and time step severely limits energy exchange. The explicit scheme cannot account for collisional heat transfer beyond neighbor cells so that heat transfer is limited by numerics, not by the physics. A similar situation arises in magnetic field diffusion near a zero point at the plasma compression on the

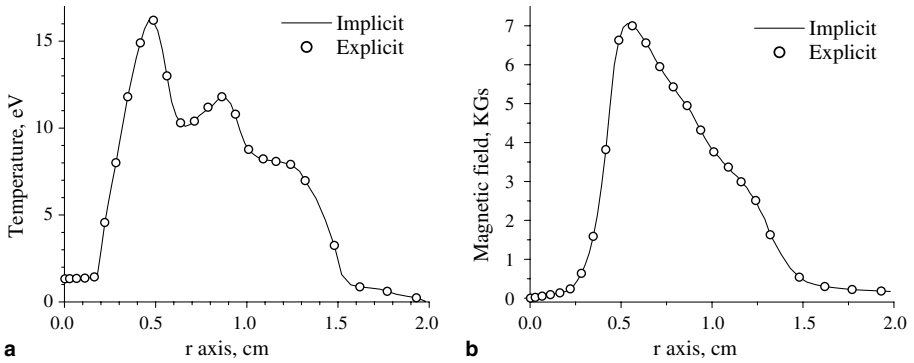


Fig. 13. Comparison of radial profiles of: (a) temperature and (b) magnetic field determined by explicit and implicit schemes in a discharge plasma device at monitor point $z = 1.5$ cm and at $t = 200$ ns after the start of the discharge. The time step is $\Delta t = 5$ ps and $\Delta t = 5 \times 10^{-2}$ ps for implicit and explicit methods, respectively.

radial axis [53] where an additional procedure is needed to damp non-physical oscillations arising in the explicit method.

## 8. Conclusions

Our objective was to develop a simple and consistent approach to solving boundary value and initial value problems encountered in the fields of physics and biophysics. We explored an FD approach for solving cross-disciplinary problems such as membrane-mediated protein–protein interactions and heat and magnetic field diffusion in plasmas. Both cases exhibit inclusions (proteins or electrodes) within the computational domain. Boundary conditions are formulated on the external boundary of the computational domain and on each inclusion boundary. Although the physics of the problems and the corresponding mathematical models are mutually incompatible, FD discretization of the equations describing the phenomena lead to similar systems of linear algebraic equations with a sparse matrix. The FD approach is simple, consistent, stable and convergent.

We showed that many-body elastic interactions between proteins in membranes can be efficiently treated using the FD-CHO approach. This allows treatment of arbitrary finite-size aggregates of inclusions automatically accounting for non-pairwise interactions. The elastic problem (1)–(4) must be solved numerically for just a few sets of boundary parameters to determine the associated effective elastic constants. With these spring constants defined, the elastic free energy of all possible fluctuations in interfacial inclusion displacements and inter-inclusion separations in the regular cluster can be calculated analytically.

Results of two test cases in a cold medium and simulations of heat and magnetic field diffusion in a plasma are presented to confirm the accuracy and stability of the fully implicit scheme. The fully implicit scheme yields results comparable to the explicit method while permitting time steps 100 times larger. Using the implicit scheme, we simulated heat and magnetic field diffusion in discharge-produced plasma devices [54]. The fully implicit scheme was fully stable, even in combination with other processes in the plasma such as radiation transfer, thermomagnetic source, laser beam interactions.

## Acknowledgments

G.V.M., M.B.P. and P.C.J. were supported by a grant from the National Institutes of Health, GM-28643. V.A.S. and A.H. were supported by a grant from Intel Corporation.

## References

[1] K.W. Morton, D.F. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, 1994.
[2] J.W. Thomas, Numerical Partial Differential Equations: Finite Difference Methods, Springer-Verlag, 1995.
[3] D. Braess, Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics, Cambridge University Press, 1997.
[4] G. Birkhoff, R.E. Lynch, Numerical Solutions of Elliptic Problems, SIAM Publications, Philadelphia, 1984.
[5] R.D. Richtmyer, K.W. Morton, Difference Method for Initial-Value Problems, Interscience, New York, 1967.
[6] P.J. Roach, Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, NM, 1976.
[7] P. Jordan, G. Miloshevsky, M. Partenskii, Energetics and gating of narrow ionic channels: The influence of channel architecture and lipid-channel interactions, in interfacial catalysis, in: A.G. Volkov (Ed.), Interfacial Catalysis, vol. 95, Marcel Dekker, Inc, New York, 2003, pp. 493–534 (Chapter 3).
[8] M.A. Liberman, J.S. DeGroot, A. Toor, R.B. Spielman, Physics of High-Density Z-Pinch Plasmas, Springer, New York, 1999.
[9] V.M. Kovenya, S.G. Cherny, V.I. Pinchukov, Numerical modeling of stationary separation flows, Proceedings of the IV International Conference on Boundared and International or Layers, Boole Press, 1986, pp. 117–129.

[10] V.M. Kovenya, S.G. Cherny, A.S. Lebedev, Numerical methods for solving the gas dynamics and Navier–Stokes equation on the basic of the splitting up-method, in: G.V. Davis, C. Fletcher (Eds.), Computational Fluid Dynamics, North-Holland, 1988.
[11] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[12] S. Marcelja, Lipid-mediated protein interaction in membranes, Biochim. Biophys. Acta 455 (1976) 1–7.
[13] H. Schröder, Aggregation of proteins in membranes. An example of fluctuation-induced interactions in liquid crystals, J. Chem. Phys. 67 (1977) 1617–1619.
[14] J.C. Owicki, M.W. Springgate, H.M. McConnell, Theoretical study of protein–lipid interactions in bilayer membranes, Proc. Natl. Acad. Sci. USA 75 (1978) 1616–1619.
[15] J.C. Owicki, H.M. McConnell, Theory of protein–lipid and protein–protein interactions in bilayer membranes, Proc. Natl. Acad. Sci. USA 76 (1979) 4750–4754.
[16] L.T. Pearson, J. Edelman, S.I. Chan, Statistical mechanics of lipid membranes. Protein correlation functions and lipid ordering, Biophys. J. 45 (1984) 863–871.
[17] L.D. Landau, E.M. Lifshitz, Statistical Physics, Addison-Wesley, Reading, MA, 1969.
[18] H.W. Huang, Deformation free energy of bilayer membrane and its effect on gramicidin channel lifetime, Biophys. J. 50 (1986) 1061–1070.
[19] P. Helfrich, E. Jakobsson, Calculation of deformation energies and conformations in lipid membranes containing gramicidin channels, Biophys. J. 57 (1990) 1075–1084.
[20] T.A. Harroun, W.T. Heller, T.M. Weiss, L. Yang, H.W. Huang, Theoretical analysis of hydrophobic matching and membrane-mediated interactions in lipid bilayers containing gramicidin, Biophys. J. 76 (1999) 3176–3185.
[21] H. Aranda-Espinoza, A. Berman, N. Dan, P. Pincus, S.A. Safran, Interaction between inclusions embedded in membranes, Biophys. J. 71 (1996) 648–656.
[22] S. May, A. Ben-Shaul, Molecular theory of lipid–protein interaction and the L-HII transition, Biophys. J. 76 (1999) 751–767.
[23] K.S. Kim, J. Neu, G. Oster, Curvature-mediated interactions between membrane proteins, Biophys. J. 75 (1998) 2274–2291.
[24] M.B. Partenskii, G.V. Miloshevsky, P.C. Jordan, Stabilization of ion channels due to membrane-mediated elastic interaction, J. Chem. Phys. 118 (2003) 10306–10311.
[25] M.B. Partenskii, G.V. Miloshevsky, P.C. Jordan, Membrane inclusions as coupled harmonic oscillators: Effects due to anisotropic membrane slope relaxation, J. Chem. Phys. 120 (2004) 7183–7193.
[26] M.B. Partenskii, P.C. Jordan, Membrane deformation and the elastic energy of insertion: Perturbation of membrane elastic constants due to peptide insertion, J. Chem. Phys. 117 (2002) 10768–10776.
[27] S. Miklin, Variational Methods in Mathematical Physics, Macmillan, 1964.
[28] T.A. Harroun, W.T. Heller, T.M. Weiss, L. Yang, H.W. Huang, Experimental evidence for hydrophobic matching and membrane-mediated interactions in lipid bilayers containing gramicidin, Biophys. J. 76 (1999) 937–945.
[29] C. Nielsen, M. Goulian, O.S. Andersen, Energetics of inclusion-induced bilayer deformations, Biophys. J. 74 (1998) 1966–1983.
[30] N.A. Krall, A.W. Trivelpiece, Principles of Plasma Physics, McGraw-Hill, New York, 1973.
[31] D.E. Potter, Numerical studies of the plasma focus, Phys. Fluids 14 (1971) 1911–1924.
[32] S. Maxon, J. Addleman, Two-dimensional magnetohydrodynamic calculations of the plasma focus, Phys. Fluids 21 (1978) 1856–1865.
[33] H. Knoepfel, Pulsed High Magnetic Fields, North-Holland Publisher, Amsterdam-London, 1970.
[34] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge University Press, 1989.
[35] The NAG Fortran Library Manual, NAG Central Office, Mayfield House, 256 Banbury Road, Oxford. Available from: <http://www.nag.co.uk>.
[36] Harwell Subroutine Library, UKAEA Harwell, 1988. Available from: <http://www.cse.clrc.ac.uk/nag/hsl/contents.shtml>.
[37] J.A. Lundbaek, O.S. Andersen, Spring constants for channel-induced lipid bilayer deformations. Estimates using gramicidin channels, Biophys. J. 76 (1999) 889–895.
[38] Y.S. Neustadt, M.B. Partenskii. Available from: <arXiv:physics/0212038>, 2002.
[39] L.E. Townsley, W.A. Tucker, S. Sham, J.F. Hinton, Structures of gramicidins A, B, and C incorporated into sodium dodecyl sulfate micelles, Biochemistry 40 (2001) 11676–11686.
[40] R.L. Goforth, K.C. Aung, V.G. Denise, L.P. Lyndon, R.E. Koeppe II, O.S. Andersen, Hydrophobic coupling of lipid bilayer energetics to channel function, J. Gen. Physiol. 121 (2003) 477–493.
[41] T.I. Rokitskaya, E.A. Kotova, Y.N. Antonenko, Tandem gramicidin channels cross-linked by streptavidin, J. Gen. Physiol. 121 (2003) 463–476.
[42] L. Al-Momani, P. Reiß, U. Koert, A lipid dependence in the formation of twin ion channels, Biochem. Biophys. Res. Commun. 328 (2005) 342–347.
[43] A.A. Samarskii, Yu.P. Popov, Difference Methods for Solving Problems of Gas Dynamics, Nauka, Moscow, 1980 (in Russian).
[44] J.S. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Hemisphere, New York, 1997.
[45] F.J. Vesely, Computational Physics. An Introduction, second revised ed., Kluwer Academic/Plenum Publishers, New York-London, 2001.

[46] A. Hassanein, V. Sizyuk, V. Tolkach, V. Morozov, B.J. Rice, HEIGHTS initial simulation of discharge-produced plasma hydrodynamics and radiation transport for EUV lithography, in: Proceedings of SPIE. Emerging Lithographic Technologies VII, vol. 5037 part 2, 2003, pp. 714–727.
[47] A. Hassanein, V. Sizyuk, V. Tolkach, V. Morozov, B.J. Rice, HEIGHTS initial simulation of discharge produced plasma hydrodynamics and radiation transport for extreme ultraviolet lithography, J. Microlithogr. Microfabric. Microsyst. 3 (2004) 130–138.
[48] T. Krucken, K. Bergmann, L. Juschkin, R. Lebert, Fundamentals and limits for the EUV emission of pinch plasma sources for EUV lithography, J. Phys. D: Appl. Phys. 37 (2004) 3213–3224.
[49] K.T. Lee, S.H. Kim, D. Kim, T.N. Lee, Numerical study of Z-pinch carbon plasma, Phys. Plasmas 3 (1996) 1340–1347.
[50] K.G. Whitney, Momentum and heat conduction in highly ionizable plasmas, Phys. Plasmas 6 (1999) 816–830.
[51] A.S. Kingsep, V.E. Karpov, A.I. Lobanov, Y. Maron, A.A. Starobinets, V.I. Fisher, Numerical modeling of the dynamics of a slow Z-pinch, Plasma Phys. Rep. 28 (2002) 286–295.
[52] G. Toth, D. Odstrcil, Comparison of some flux corrected transport and total variation diminishing numerical schemes for hydrodynamic and magnetohydrodynamic problems, J. Comput. Phys. 128 (1996) 82–100.
[53] 2D MHD Calculations of the Plasma Focus, Lawrence Livermor Laboratory, Preprint 79067, 1977.
[54] A. Hassanein, V. Sizyuk, V. Tolkach, V. Morozov, T. Sizyuk, B.J. Rice, V. Bakshi, Simulation and optimization of DPP hydrodynamics and radiation transport for EUV lithography devices, in: Proceedings of SPIE. Emerging Lithographic Technologies VIII, vol. 5374 part 1, 2004, pp. 413–422.