JOURNAL OF COMPUTATIONAL PHYSICS **143,** 346–380 (1998)
ARTICLE NO. CP985967

An Adaptive, Cartesian, Front-Tracking Method for the Motion, Deformation and Adhesion of Circulating Cells

G. Agresar,* J. J. Linderman,† G. Tryggvason,‡ and K. G. Powell§

*Departments of Microbiology and Immunology and Biomedical Engineering; †Department of Chemical Engineering; ‡Department of Mechanical Engineering and Applied Mechanics; and §Department of Aerospace Engineering, The University of Michigan, Ann Arbor, Michigan 48109

E-mail: *agresar@engin.umich.edu, †linderma@engin.umich.edu, ‡gretar@engin.umich.edu, and §powell@engin.umich.edu

Received March 21, 1997; revised March 17, 1998

Cells in circulatory systems adhere through a competition between molecular interactions and colloidal repulsion, while the cells arbitrarily deform in the presence of external fluid forces. The complex coupling of the forces involved, the disparate length scales at which they act, and uncertainties in the mechanics of cell deformation have complicated the study of cell adhesion. To address these difficulties, a multi-fluid, front-tracking method with staggered, adaptively refined meshes has been developed. As a tool to study cell mechanics, the program allows the incorporation and testing of different mechanical models of the cell without significant changes in the setup. As a tool to study cell adhesion, the method models the coupling of the relevant forces resolving the disparate length scales involved. The method was validated by simulating various test cases, and the results were found to agree well with analytical and other numerical solutions. The capabilities of the method are demonstrated with the simulation of a common cell-mechanics experiment (a micropipet assay) and a common physiological situation for cell adhesion (the adhesion of two cells under shear flow).

Key Words: cell mechanics; cell detachment; incompressible Navier–Stokes; micropipet; micropipette; drops; adaptive refinement; immersed boundary method.

1. INTRODUCTION

The transient and specific adhesion of cells is crucial to numerous physiological processes including cell-mediated immunity, embryogenesis, wound healing, and the spread of cancer. Understanding what determines the specificity, strength, and occurrence of cell adhesion could lead to the therapeutic treatment of many disorders, as well as the acceleration of advances in biotechnological processes. Already, there is much pharmacological effort

0021-9991/98 $25.00

aimed at controlling cell adhesion at a molecular level [1]. In addition, the development of manufactured devices for use in the human body—such as synthetic vascular grafts to replace diseased or damaged segments of arteries, cardiac valves, fully artificial hearts, and other artificial organs—has provided both the need and the opportunity to determine the forces present in cell adhesion under flow [2].

Previous research has demonstrated that cell adhesion is governed by the coupling of several physical and biochemical events that occur at drastically different length scales. On the scale of the cell diameter ($\sim$10 μm), the cells deform while immersed in a stream of fluid. At the gap near the area of contact ($\sim$10 nm), molecules diffuse and react while colloidal forces act to repel the cells. The complex coupling of these events and the disparate length scales at which they occur have complicated both experimental and mathematical studies of cell adhesion. Experimental work ranges from laminar flow chamber and micromanipulation assays aimed at determining the occurrence of adhesion and at measuring the forces required to disrupt it, to molecular biology techniques used to characterize the surface molecules involved in the binding. Assays which relate molecular and colloidal properties to the forces required to disrupt adhesion are difficult and scarce. A mathematical relationship between these properties is desirable. Most mathematical models of cell adhesion are based on two equilibrium models. In one, adhesion is considered to be a competition between the specific binding of surface molecules and the non-specific repulsion between the cells [3]; in the other, a relationship between the mechanical properties of a portion of the cell membrane and adhesive bonds was developed [4, 5]. These models have been combined and extended to account for some kinetic phenomena [6–8] and for the influence of external flow [9, 10]. Unfortunately, the mathematical work so far treats subsets of the relevant events in cell adhesion, ignoring others that are equally important. For example, one model [9] includes all binding dynamics and hydrodynamic flow but neglects cell deformation, while another model [6] includes both binding dynamics and mechanical deformation but considers only a portion of the cell membrane.

Further difficulties arise from the fact that the mechanical behavior of most cells is not well defined and that it varies with cell type and cell status (e.g., whether the cell is activated or not). Cell deformation affects studies of adhesion by helping determine the size and geometry of the area of contact, by altering the flow field surrounding the cells, and by absorbing some of the force intended to disrupt adhesion. Although cellular deformation has been studied extensively (e.g., [11–13]), the constitutive relationship between stress and deformation for most cells is still elusive. The common procedure for testing hypotheses for such a relationship requires that the relationship be established *a priori* and that a new analytical setup be developed for each test.

A computational model could accommodate alternative models for the cell mechanics, as well as provide the coupling of the relevant events in cell adhesion. This approach was used by Fogelson [14] to simulate platelet aggregation under flow. He modeled the platelets as two-dimensional fluid bodies enclosed by an infinitesimally thin elastic membrane and surrounded by a fluid of identical properties. However, he made no provisions for varying the properties of the fluid inside the cell with respect to the surrounding fluid, limiting the alternatives for mechanical models. Also, while Fogelson's model includes the activation of platelets that precedes their aggregation, the adhesive interactions are treated with a simple model which ignores the different length scales.

This paper describes the implementation of a method that incorporates the most relevant events involved in the adhesion of cells under flow. Extending Fogelson's idea, the biological

cells are modeled as composites of incompressible, Newtonian fluid bodies enclosed by membranes. Alternative models for the overall deformation of the cell can be constructed by varying the properties of the internal fluids, by varying the constitutive equation and material parameters for the mechanics of the membranes, and by varying the number of internal bodies (which represent the nucleus or other organelles inside the cell).

A multi-fluid program has been developed to describe the behavior of such cells and the encompassing fluid in both two-dimensional and axisymmetric geometries. The incompressible Navier–Stokes equations are discretized using a finite-volume formulation of a semi-implicit pressure correction method. The interfaces are tracked explicitly and discretized independently of the background mesh using the Eulerian–Lagrangian method developed by Univerdi and Tryggvason [15]. In this method, all membrane processes are computed on the interfaces and the resultant force is distributed to the background mesh. This separate treatment, as well as the lower dimensionality of the membranes, simplifies the alteration of their mechanical properties and the inclusion of additional processes. To account for the disparate length scales of the problem, the fluid equations are solved on unstructured Cartesian grids that refine adaptively near the interfaces [16]. The unstructured meshes allow the adaptation to be performed easily and efficiently, while the Cartesian elements allow a simpler discretization of the equations, avoiding the many complications introduced by the transformations usually needed on non-orthogonal grids.

The combination of front-tracking and adaptive refinement has been used by other researchers to study problems from different fields. For example, a front-tracking method in which a composite grid is formed by the overlay of the interface mesh on the background Cartesian mesh has been combined with adaptive refinement to study unsteady inviscid flows with a collocated variable arrangement [16] and, independently, incompressible viscous flows with a staggered variable arrangement [17]. A variation of the level-set approach [18] has been combined with a similar Cartesian adaptive-refinement method using a collocated variable arrangement to study the motion of drops in an incompressible Newtonian fluid [19, 20]. Level-set methods have also been used with a different type of Cartesian-adaptive mesh to study incompressible two-phase flows [21]. Our work is the first report of adaptively refined, staggered meshes used in conjunction with an explicit interface tracking in which the interfaces are not part of the background grid. Although the motivation for our work is the adhesion of deformable cells, the method applies to many other physical systems such as the interactions of drops and bubbles and the swimming of microorganisms.

In the following section, the governing equations that describe the behavior of incompressible, Newtonian fluids with surface forces are presented. Next, the implementation of the numerical method is described. The grid generation, the front tracking, and the discretization of the equations on a staggered mesh are described in Sections 3.1–3.4, while the boundary conditions are discussed in Section 3.5. The discussion and derivations in these sections are carried out in two dimensions, and the extension of the algorithm to axisymmetric geometries is presented in Section 3.6. Finally, Section 4 presents the results of several simulations: the deformation a 2-D drop under shear and the oscillations of a drop are used to validate the program, while the simulation of a micropipet assay (a common technique used to study cell mechanics) and two cells adhering under shear are used to demonstrate the capabilities of the method.

2. GOVERNING EQUATIONS

The fluids modeled in this work (e.g., the cytosol and extracellular fluids such as plasma, media or buffers) are incompressible and will be assumed to remain isothermal. The governing equations that describe the behavior of such fluids are the laws of conservation of mass,

$$\oint_S \vec{v} \cdot \vec{n}\, dS = 0, \tag{1}$$

and momentum,

$$\frac{\partial}{\partial t} \iiint_V \rho \vec{v}\, dV + \oint_S \rho \vec{v}(\vec{v} \cdot \vec{n})\, dS = -\oint_S p\overline{\overline{I}} \cdot \vec{n}\, dS - \oint_S \overline{\overline{\tau}} \cdot \vec{n}\, dS + \iiint_V \vec{f}_V\, dV, \tag{2}$$

where t represents time, while $\vec{v}$, p, and ρ are the velocity vector, the pressure, and the density of the fluid, respectively. $\overline{\overline{I}}$ is the identity matrix, V is the volume occupied by the fluid, S is the surface area formed by the boundary of this volume, and $\vec{n}$ is the outward-pointing unit vector, normal to S. $\vec{f}_V$ is the sum of the forces acting on the fluid; although it has the form of a body force, it also includes the effect of the surface forces due to the front tracking method used (as will be discussed in Section 3.3). $\overline{\overline{\tau}}$ is the stress tensor, which for a Newtonian fluid in two dimensions is

$$\overline{\overline{\tau}} = \begin{bmatrix} -2\mu \dfrac{\partial u}{\partial x} & -\mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) \\ -\mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) & -2\mu \dfrac{\partial v}{\partial y} \end{bmatrix}, \tag{3}$$

where μ is the dynamic viscosity of the fluid, and u and v are the horizontal and vertical components of the velocity, respectively.

3. NUMERICAL METHOD

3.1. *Grid Generation*

The grid-generation code developed by Bayyuk *et al.* [16] is used to generate the grids for solving the discretized fluid equations. The code uses a quadtree-based algorithm to generate Cartesian meshes with adaptive refinement and was originally developed for compressible, inviscid flows. A quadtree structure begins with a root cell which is said to be at refinement level 1. This root cell is refined by creating four children of equal size, which are said to be at refinement level 2. Each child can in turn be refined, increasing the refinement level by one, until the desired grid is generated. All cells have a pointer to their parents and one to each of their four children, if they exist. From this tree structure most geometric information as well as the connectivity of the cells can be determined. The state variables are stored at the leaf cells (those at the bottom of the tree), and all flow-solver computations are done using these cells. For more details on the data structure see De Zeeuw [22] or Bayyuk [23].

The creation of the grid can be summarized in four steps: (i) a uniform Cartesian mesh is generated, (ii) points are distributed along each interface, (iii) the intersections between

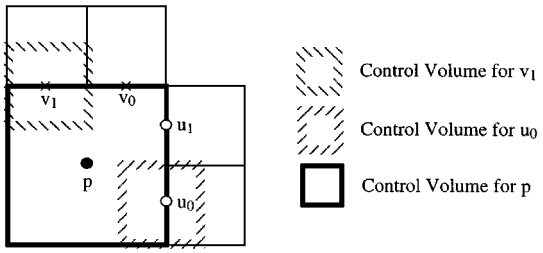

FIG. 1. Staggered variable arrangement near refinement interfaces with the corresponding control volumes.

the interface(s) and the uniform Cartesian mesh are computed, and (iv) the Cartesian mesh is refined appropriately. At each time step, each interface can deform and the background mesh is adapted accordingly. Effectively, the interfaces move through a stationary grid, and this provides a very efficient method to generate grids for arbitrarily deforming bodies. For more details on this procedure see Bayyuk *et al.* [16] or Bayyuk [23].

For the problems studied in this work, cells are refined using two criteria: intersection by the interface, and proximity either of adjacent interfaces or of an interface and wall. The first provides a method to control the sharpness of the interface, while the second provides the resolution needed to study interfacial phenomena efficiently. Additional refinement criteria based on the gradient of the state variables could be used to achieve accuracy in more complicated flows (e.g., [16, 24]). In the work of Bayyuk and Powell, the interfaces appear in the background grid, forming irregularly cut cells. In this study, the complications introduced by these cut cells are avoided by keeping the interfaces separate from the background mesh and distributing the quantities carried by the interface to points in the Cartesian mesh. The implementation of the distribution of these quantities is the subject of Section 3.3.

3.2. *Implementation of the Staggered Variable Arrangement*

A diagram of a staggered, refined mesh is shown in Fig. 1. The pressure, density, and viscosity are located at the cell centroids; the horizontal component of the velocity is placed at the center of the vertical cell faces; and the vertical component of the velocity is located at the center of the horizontal cell faces.

The refinement has been restricted so that the refinement levels of neighboring cells do not differ by more than one. This restriction is effectively a smoothing of the grid and not only simplifies the implementation of the staggered mesh, but also prevents inaccuracies introduced by large size differences between adjacent cells [22]. With this restriction, each face of a cell can have a maximum of two nodes. In order to accommodate the staggered location of the unknowns into the quadtree data structures used in the grid generation code (which are cell-based), each cell is assigned the nodes at its east and north faces, along with the node at its centroid. Hence, each cell contains five pointers to variables, and whenever a face has only one node (when the refinement level of the face neighbor is greater than or equal to that of the cell), one of the pointers is set to NULL.

3.3. *Front Tracking*

The method used to track the interfaces was originally developed by Peskin and McQueen [25] and was extended to fluids with different densities and viscosities by Univerdi and Tryggvason [15]. In this method, a moving, unstructured mesh is used to track the interface,

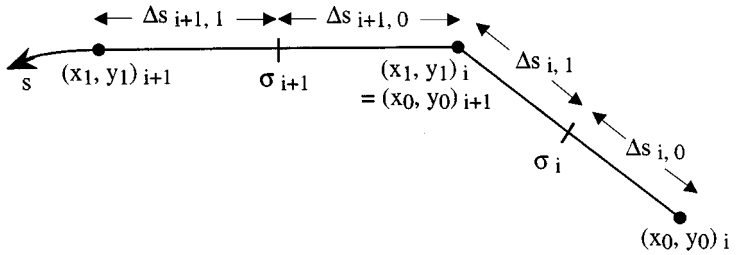

FIG. 2. Sketch of the discretization of the interface. The subscript i represents the segment number, s is the coordinate along the interface, and the subscripts 0 and 1 are the endpoints of each segment numbered in a counterclockwise manner.

and a stationary Cartesian mesh is used to solve the transport equations. The two meshes are kept separate, and information transfer is required from one mesh to another. This transfer is as follows: (i) a resultant surface force is computed on the interface and must be distributed to the Cartesian mesh; (ii) the velocity of the fluid is computed on the background mesh and must be interpolated to the interface; and (iii) the location of the jumps in density and viscosity is determined by the interface and must be communicated to the background mesh.

The discretization of a two-dimensional interface is sketched in Fig. 2. The coordinates at the endpoints of each segment and the property variables at the centroids are stored. All membrane processes and properties are computed using this mesh and the effects of the membrane on the fluid are transmitted through a resultant surface force, $\vec{f}_S$. For example, at the interface between two fluids [26],

$$\vec{f}_S = \frac{\partial(\sigma \vec{e}_t)}{\partial s}\, ds. \tag{4}$$

The discretization of this force on the interface mesh of Fig. 2 results in the following expression for each segment,

$$\vec{f}_{S_i} = (\sigma \vec{e}_t)_{i,1} - (\sigma \vec{e}_t)_{i,0}, \tag{5}$$

where $(\sigma \vec{e}_t)$ is the product of the surface force coefficient and a unit tangent vector at one endpoint of the segment. This product is computed by a linear interpolation of similar terms evaluated at the centroid of the two segments that share the endpoint (the surface tension coefficient is stored at that location, whereas the unit tangent vector is evaluated by finding the vector connecting the endpoints in a counterclockwise manner and normalizing it by the length of the segment).

The resultant force per unit surface element must be distributed to the Cartesian grid and incorporated into Eq. (2) through the force term, $\vec{f}_V$. This is done by transforming the surface force to a volume force using an area-weighted extrapolation. Since the mesh is staggered, the component of the force acting in the horizontal direction is distributed to the u-nodes (the nodes containing the horizontal component of the velocity), while the y component is distributed to the v-nodes. The procedure is as follows: (i) the quadtree is searched to find the cell that encloses the midpoint of an element in the interface; (ii) by searching the neighbors of the cell just found, the four nodes closest to the midpoint of the element are found; (iii) using a bilinear transformation, the quadrilateral formed by

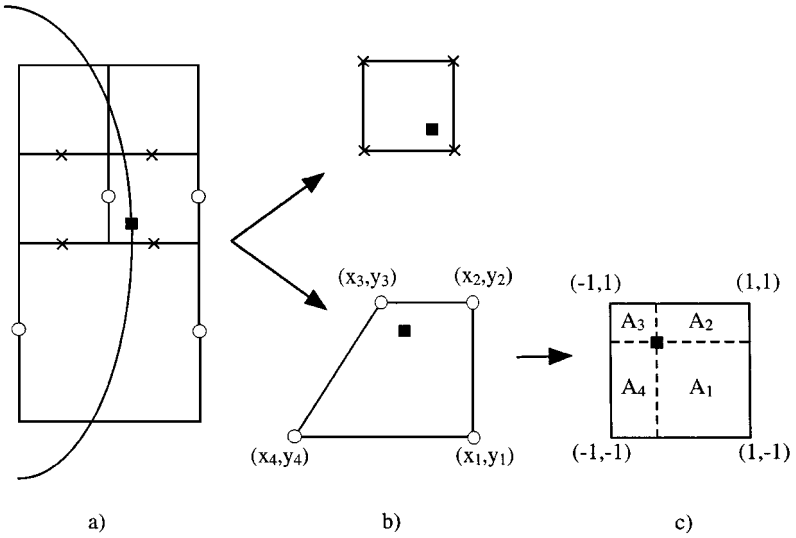

FIG. 3. Distribution of the surface force: (a) Interface overlaid on the Cartesian mesh. The open circles indicate the u-nodes, to which the x-component of the force is interpolated; the crosses mark the v-nodes, to which the y-component of the force is interpolated; and the filled square represents the midpoint of an element of the interface. (b) Quadrilaterals formed by the v-nodes (top) and u-nodes (bottom) which are used to compute the fractional areas. (c) Mapping of the quadrilateral formed by the u-nodes around the interface point to a square with vertices $[(1, 1), (-1, 1), (-1, -1), (1, -1)]$.

these nodes is mapped to a unitary square where fractional areas are computed[1]; (iv) these fractional areas are used to distribute a component of the surface force to the four nodes found in step (ii). This procedure is sketched in Fig. 3. The expression for the force per unit fluid element assigned to each u-node is

$$(\vec{f}_V \cdot \vec{i})_k = \frac{1}{V_k}(\vec{f}_S \cdot \vec{i})_i A_k, \tag{6}$$

where $\vec{i}$ is a unit vector in the horizontal direction, k indicates each of the four u-nodes forming a quadrilateral around the midpoint of the ith element on the interface, V_k is the size of the control volume around the kth u-node, and A_k is the fractional area associated with the kth u-node ($\sum_{k=1}^{4} A_k = 1$), as shown in Fig. 3c. A similar expression is used for the y-component of the force. This distribution is performed for all interface elements, and the contributions of different elements to the same node in the Cartesian mesh are added.

The local fluid velocity is interpolated to the endpoints of each interface segment using an area-weighted interpolation which incorporates most of the features developed for the distribution of the surface force described above. The expressions for the components of the velocity at endpoint 0 of the ith element are

$$u_{i,0} = \sum_{k=1}^{4} u_k A_k \tag{7}$$

[1] The actual fractional areas would require mapping back to the real space, but after some testing it was found that the areas in the transformed space are a very good approximation to the real ones for the quadrilaterals encountered in the meshes of this work. To avoid further complications in the setup and for efficiency, the fractional areas in the transformed space were used.

$$v_{i,0} = \sum_{j=1}^{4} v_j A_j, \tag{8}$$

where k refers to each of the four u-nodes forming a quadrilateral around the endpoint, and j refers to the corresponding v-nodes (as sketched in Fig. 3b). The appropriate nodes of the Cartesian mesh and the fractional areas are found using the same method described for the distribution of the surface force.

The density and viscosity in the computational cells which are not intersected by any front are assigned according to the fluid regime in which the cell lies. In intersected cells, these variables are computed using an area-weighted average,

$$\rho_k = \frac{1}{N_k + 1} \sum_{i=1}^{N_k+1} \left\{ \frac{1}{A_i} (\rho_L A_L + \rho_R A_R)_i \right\} \tag{9}$$

$$\mu_k = \frac{1}{N_k + 1} \sum_{i=1}^{N_k+1} \left\{ \frac{1}{A_i} (\mu_L A_L + \mu_R A_R)_i \right\}, \tag{10}$$

where N_k is the number of cells neighboring cell k, $i = N_k + 1$ refers to cell k, $A_i = (A_L + A_R)$ is the total area of cell i, and the subscripts L and R represent the values at each side of the interface on the ith cell (if the ith cell is not intersected, either A_L or A_R is zero). The cell neighbors were included in the averaging as a form of smoothing to avoid sharp jumps in the density and viscosity, which can produce spurious spikes in the solution. This method of reconstructing the density and viscosity takes advantage of the geometric information provided by the grid-generation code (the fluid or fluids that occupy a cell, and the geometry of the intersection) [23]. An alternative method for this reconstruction is described in Refs. [15, 26].

The current implementation of the grid-generation code prohibits intersected boundary cells and cells intersected by more than two interfaces (this latter restriction will be lifted in future work). This is enforced by defining a contact threshold to be a distance 1.5 times the side of a computational cell. The velocities of interfacial points that, at their new location, would lie within a contact threshold of other interfaces or solid boundaries are adjusted using the following rules: (i) if a point will get within a contact threshold of a solid boundary, the component of its velocity normal to the boundary is set to zero; (ii) if two points (on separate interfaces) will get within a contact threshold of each other, the velocity of both points is set to the average of the two. The first of these rules effectively imposes an inviscid boundary condition for the solid wall one contact threshold away from the boundary. The second rule assumes that portions of the interfaces "collide" when the distance between them is equal to the contact threshold, that they "stick" together for one time step after the collision, and that the portions colliding have equal mass. In most problems, these assumptions are justified since the velocity adjustments needed are typically very small and the errors introduced decrease with the size of the mesh, the size of the segments, and the size of the time step. In cases in which many velocity adjustments are performed, these adjustments might significantly alter the volume of the bodies. One way to correct this problem is to communicate the velocity adjustments on the interface to the immersing fluids. This can be done by including the force needed to displace the fluid between the unadjusted and adjusted locations of each segment into the total surface force, $\vec{f}_S$ (Eqs. (4)–(6)).

The number of interface points is computed automatically so that the size of each segment on the interface is at most the length of a face of the smallest computational cell on the Cartesian grid. This guarantees that there is at least one interface point per cell. In the cases tried, the authors have found that the difference in the solution between one and two interface points per cell is small, and using more than two points does not affect the solution significantly.

Since the points on the interface move independently of one another, special measures must be taken to prevent two points from becoming unevenly spaced. This is accomplished simply by redistributing the points along the interface, while maintaining the total number of segments constant. The quantities carried by the interface are adjusted as needed. This redistribution is performed after several iterations (the exact number of iterations is problem dependent).

3.4. *Discretization of the Incompressible Navier–Stokes Equations*

3.4.1. *Finite-volume discretization.* The following finite-volume discretization of Eqs. (1) and (2) is used,

$$\sum_{\text{faces}} \vec{v} \cdot \vec{n} \Delta S = 0 \tag{11}$$

$$\begin{aligned}\frac{(\rho \vec{v} V)^{n+1} - (\rho \vec{v} V)^n}{\Delta t} = &- \sum_{\text{faces}} (\bar{\bar{F}}_{\text{conv}} \cdot \vec{n} \Delta S)^n - \sum_{\text{faces}} (\bar{\bar{F}}_{\text{press}} \cdot \vec{n} \Delta S)^{n+1} \\ &+ \theta \sum_{\text{faces}} (\bar{\bar{F}}_{\text{visc}} \cdot \vec{n} \Delta S)^{n+1} + (1-\theta) \sum_{\text{faces}} (\bar{\bar{F}}_{\text{visc}} \cdot \vec{n} \Delta S)^n + \vec{f} V,\end{aligned} \tag{12}$$

where V is the size of the control volume; ΔS is the size of a face; $\vec{n}$ is the outward normal to the face; θ is a constant ($0 \le \theta \le 1$) used to control the degree of implicitness used for the viscous terms; and the total flux has been divided into three components—convective fluxes ($\bar{\bar{F}}_{\text{conv}}$), pressure fluxes ($\bar{\bar{F}}_{\text{press}}$), and viscous fluxes ($\bar{\bar{F}}_{\text{visc}}$)—to facilitate the discussion in the following section. These fluxes are defined as

$$\begin{aligned}\bar{\bar{F}}_{\text{conv}} &= \begin{bmatrix} \rho u^2 & \rho u v \\ \rho u v & \rho v^2 \end{bmatrix} \\ \bar{\bar{F}}_{\text{press}} &= \begin{bmatrix} p & 0 \\ 0 & p \end{bmatrix} \\ \bar{\bar{F}}_{\text{visc}} &= \begin{bmatrix} 2\mu \dfrac{\partial u}{\partial x} & \mu \left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \right) \\ \mu \left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \right) & 2\mu \dfrac{\partial v}{\partial y} \end{bmatrix},\end{aligned} \tag{13}$$

where u and v are the x- and y-components of the velocity vector, respectively.

Since the biological problems of interest occur at low Reynolds numbers, the convective terms do not enforce a stringent stability constraint on the time step, and, by choosing $\theta \ge \frac{1}{2}$ in Eq. (12), the viscous terms are unconditionally stable. In the cases tried so far, the

restriction on the time step has been due to the membrane forces. Therefore, the condition

$$\Delta t < \text{ minimum over all intersected cells} \left(c \sqrt{\frac{\Delta x \rho^{n+1}}{f_V}} \right)$$

is imposed on the time step, where f_V is the magnitude of the force per unit volume defined in Eq. (6), Δx is the length of the computational cell, ρ at the newest time level is defined by Eq. (9), and c is a safety factor. This is a CFL-like condition based on the acceleration of the fluid due to the surface forces. This condition is designed to prevent any particle of fluid and any interface point from moving more than one cell length at each time step.

In the staggered mesh, the control volume used for conservation of mass is centered at the cell centroid, the one for the conservation of momentum in the x-direction is centered at the east face, and the one for conservation of momentum in the y-direction is centered at the north face (Fig. 1). The size of the control volume for p is always the same as the computational cell in which p is located, whereas the size of the control volume around u or v is the same as the smallest cell that shares the face in which the variable is located. In what follows, the notation $\sum_{\text{faces}} \vec{n}' \Delta S'$ will be used for control volumes around the velocity variables, and $\sum_{\text{faces}} \vec{n} \Delta S$ will be for control volumes around the pressure.

3.4.2. *Pressure correction algorithm.* A method similar to the pressure-implicit with splitting of operators (PISO) algorithm developed by Issa [27] is used to couple Eqs. (11) and (12). This method is also similar in nature to the second-order projection method developed by Bell *et al.* [28]. It allows the simulation of the low Reynolds number flows often seen in biological situations without an excessive penalty in the size of the time step. To derive this method in a finite volume formulation, two auxiliary velocities are defined based on Eqs. (12)–(13) as

$$\frac{\rho^{n+1}\vec{v}^0 V^{n+1} - (\rho \vec{v} V)^n}{\Delta t} = -\boldsymbol{\mathcal{F}}^n_{\text{conv}} - \boldsymbol{\mathcal{F}}^n_{\text{press}} + \theta \boldsymbol{\mathcal{F}}^0_{\text{visc}} + (1-\theta)\boldsymbol{\mathcal{F}}^n_{\text{visc}} + \vec{f} V \quad (14)$$

and

$$\frac{\rho^{n+1}\vec{v}^{k+1} V^{n+1} - (\rho \vec{v} V)^n}{\Delta t} = -\boldsymbol{\mathcal{F}}^n_{\text{conv}} - \boldsymbol{\mathcal{F}}^{k+1}_{\text{press}} + \theta\left(a_c \vec{v}^{k+1} + F^k_{\text{visc}}\right) + (1-\theta)\boldsymbol{\mathcal{F}}^n_{\text{visc}} + \vec{f} V, \quad (15)$$

where

$$\boldsymbol{\mathcal{F}}_{\text{conv}} = \sum_{\text{faces}'} (\overline{\overline{F}}_{\text{conv}} \cdot \vec{n}' \Delta S') \quad (16)$$

$$\boldsymbol{\mathcal{F}}_{\text{press}} = \sum_{\text{faces}'} (\overline{\overline{F}}_{\text{press}} \cdot \vec{n}' \Delta S') \quad (17)$$

$$\boldsymbol{\mathcal{F}}_{\text{visc}} = \sum_{\text{faces}'} (\overline{\overline{F}}_{\text{visc}} \cdot \vec{n}' \Delta S') = a_c \vec{v} + F_{\text{visc}} \quad (18)$$

and $a_c \vec{v}$ and F_{visc} are the diagonal and non-diagonal terms of $\boldsymbol{\mathcal{F}}_{\text{visc}}$, respectively. The superscript k is used since the auxiliary velocity must be corrected more than once in order to obtain a velocity at $n+1$ which is very close to divergence-free.

Two types of correction equations are obtained. The first is obtained by subtracting Eq. (14) from (15) at $k = 0$, the result of which is

$$\left(\frac{\rho^{n+1}V^{n+1}}{\Delta t} - \theta a_c\right)(\vec{v}^1 - \vec{v}^0) = -\left(\boldsymbol{\mathcal{F}}^1_{\text{press}} - \boldsymbol{\mathcal{F}}^n_{\text{press}}\right). \tag{19}$$

The second is derived by subtracting (15) at k from the same equation at $k + 1$. This gives

$$\left(\frac{\rho^{n+1}V^{n+1}}{\Delta t} - \theta a_c\right)(\vec{v}^{k+1} - \vec{v}^k) = -\left(\boldsymbol{\mathcal{F}}^{k+1}_{\text{press}} - \boldsymbol{\mathcal{F}}^k_{\text{press}}\right) + \theta\left(F^k_{\text{visc}} - F^{k-1}_{\text{visc}}\right). \tag{20}$$

Pressure equations are obtained by taking the discrete divergence of the correction equations and enforcing continuity for all $\vec{v}^k$, $k \geq 1$. The two pressure equations are

$$\sum_{\text{faces}}\left\{\frac{1}{C_0}\left(\boldsymbol{\mathcal{F}}^1_{\text{press}} - \boldsymbol{\mathcal{F}}^n_{\text{press}}\right)\cdot\vec{n}\Delta\vec{S}\right\} = \sum_{\text{faces}}\{\vec{v}^0\cdot\vec{n}\Delta\vec{S}\} \tag{21}$$

$$\sum_{\text{faces}}\left\{\frac{1}{C_0}\left(\boldsymbol{\mathcal{F}}^{k+1}_{\text{press}} - \boldsymbol{\mathcal{F}}^k_{\text{press}}\right)\cdot\vec{n}\Delta\vec{S}\right\} = \sum_{\text{faces}}\left\{\frac{1}{C_0}\theta\left(F^k_{\text{visc}} - F^{k-1}_{\text{visc}}\right)\cdot\vec{n}\Delta S\right\}, \tag{22}$$

where

$$C_0 = \left(\frac{\rho^{n+1}V^{n+1}}{\Delta t} - \theta a_c\right).$$

The algorithm consists of solving Eqs. (14), (21), and (19) sequentially to obtain $\vec{v}^1$, and iterating Eqs. (22) and (20) until either the L_2 norm of the velocities and the pressure converge to a certain criteria ($\sim 10^{-6}$ for the velocities and $\sim 10^{-4}$ for the pressure) or until the maximum number of iterations is reached (10 or fewer iterations produced good results in the cases tried). For more details on this method see Issa [27]. Note that setting $\theta = 0$ produces an explicit pressure correction method.

The matrices generated by this algorithm are solved using the package, SPARSKIT, developed by Saad [29]. The modules for incomplete LU factorization and conjugate gradients are used in this work.

3.4.3. *Computational molecule and interpolations.* Far from refinement interfaces, the computational molecules shown in Fig. 4 are used to formulate Eqs. (14), (19), and (20) around the velocity control volumes. Using the notation in this figure, the flux terms for the x-component of the velocity (u) around the control volume shown in Fig. 4a can be written as

$$\begin{aligned}\boldsymbol{\mathcal{F}}_{\text{conv}} = \Bigg\{ & \rho_e\left(\frac{u_e + u_c}{2}\right)^2 - \rho_c\left(\frac{u_c + u_w}{2}\right)^2 \\ & + \left(\frac{\rho_c + \rho_e + \rho_{ne} + \rho_n}{4}\right)\left(\frac{u_n + u_c}{2}\right)\left(\frac{v_e + v_c}{2}\right) \\ & - \left(\frac{\rho_s + \rho_{se} + \rho_e + \rho_c}{4}\right)\left(\frac{u_c + u_s}{2}\right)\left(\frac{v_{se} + v_s}{2}\right)\Bigg\} h\end{aligned} \tag{23}$$

$$\boldsymbol{\mathcal{F}}_{\text{press}} = (p_e - p_c)h \tag{24}$$

$$\boldsymbol{\mathcal{F}}_{\text{visc}} = a_c u + F_{\text{visc}}, \tag{25}$$

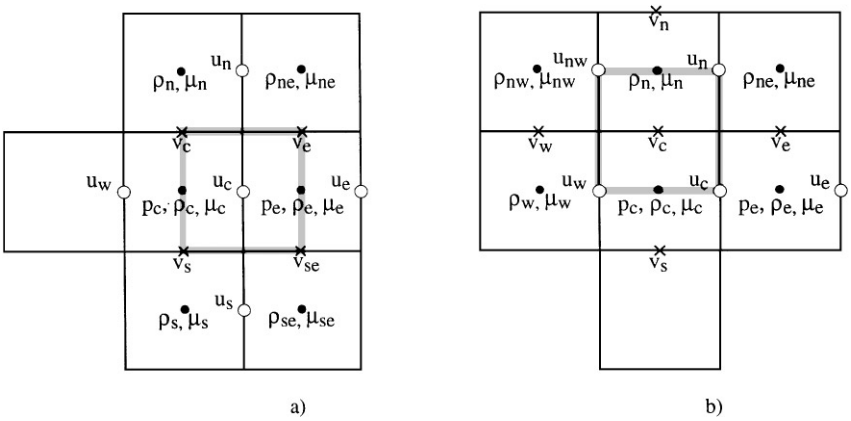

FIG. 4. Computational molecules used in the flow solver: (a) variables needed to solve the equations for u; (b) variables needed to solve the equations for v. (The shaded lines denote the control volumes.)

where h is the length of any of the faces of the control volume, and the diagonal and non-diagonal terms of $\mathcal{F}_{\text{visc}}$ can be written as

$$a_c u = \left[-2\mu_e - 2\mu_c - \left(\frac{\mu_c + \mu_e + \mu_{ne} + \mu_n}{4}\right) - \left(\frac{\mu_s + \mu_{se} + \mu_e + \mu_c}{4}\right)\right] u_c \quad (26)$$

$$F_{\text{visc}} = 2\mu_e u_e + 2\mu_c u_w + \left(\frac{\mu_c + \mu_e + \mu_{ne} + \mu_n}{4}\right)(u_n + v_e - v_c) - \left(\frac{\mu_s + \mu_{se} + \mu_e + \mu_c}{4}\right)(-u_s + v_{se} - v_s). \quad (27)$$

The formulation of the x-component of Eqs. (14), (19), and (20) is completed by setting ρ at the node for u_c to $(\rho_e + \rho_c)/2$ and $V = h^2$. Similar expressions can be written for the y-component of the velocity [30].

The equations for the pressure (Eqs. (21) and (22)) are a composite of the terms already defined. These equations are built by stepping through the faces of the control volume for the pressure, and setting the appropriate terms using the control volume of each velocity component encountered. For example, Eq. (21) can be written as

$$\left(\frac{(\rho_e + \rho_c)h_u^2}{2\Delta t} + \theta\left[2\mu_e + 2\mu_c + \frac{2\mu_c + 2\mu_e + \mu_{ne} + \mu_n + \mu_s + \mu_{se}}{4}\right]\right)^{-1}(\Delta p_e - \Delta p_c)h_p + \left(\frac{(\rho_n + \rho_c)h_v^2}{2\Delta t} + \theta\left[2\mu_n + 2\mu_c + \frac{2\mu_c + \mu_e + \mu_{ne} + 2\mu_n + \mu_w + \mu_{nw}}{4}\right]\right)^{-1} \times (\Delta p_n - \Delta p_c)h_p - \text{LHS}_W - \text{LHS}_S = u_c^0 h_p + v_c^0 h_p - \text{RHS}_W - \text{RHS}_S, \quad (28)$$

where the Δ in front of the pressure terms indicates the difference between the values at two iterations (time level n and k iteration 1 in this case); and h_u, h_v, and h_p are the sizes of one of the faces of the control volumes around the u-, v-, and p-nodes, respectively. The terms for the east and north faces have been written explicitly (recall that only the east and north faces are associated with each cell), whereas the remaining terms (LHS_W, LHS_S, RHS_W, and RHS_S) are evaluated by constructing similar computational molecules centered at the west and south neighbors.

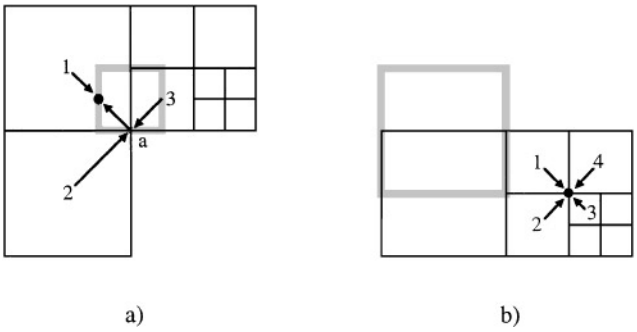

FIG. 5. Interpolations of cell-centered variables: (a) interpolation to a location inside a cell; (b) interpolation to a vertex.

Since all the discretized equations can be written in terms of the variables defined in this computational molecule, where the distance between each class of unknown is equal to h (the length of a side of the appropriate control volume), this computational molecule is used as the basis for the flow solver. Near refinement interfaces, where the size of adjacent control volumes varies, independent computational molecules are constructed with respect to each of the components of the velocity, and interpolations are defined to obtain the missing values. These interpolations have been divided into two main classes: (i) interpolations of the variables located at the cell centroid (pressure, viscosity, and density), and (ii) interpolations of the variables located at the cell faces (the x- and y-components of the velocity).

The first type of interpolation is sketched in Fig. 5. When the interpolated value lies inside a computational cell, as in Fig. 5a, the three closest values that enclose the point of interest when connected (1, 2, and 3 in the figure) are used to form a triangular linear element for the interpolation. Instead of computing the shape functions for such an element, the regular structure of the grids is used to simplify the expressions for the interpolations reducing the number of arithmetic operations to a linear interpolation to the vertex closest to the location of the interpolation followed by an average with the value at the cell centroid (Fig. 5a). For example, following the notation in Fig. 5a,

$$p_a = \frac{1}{s_{23}}(p_3 s_{2a} + p_2 s_{3a}) \tag{29}$$

$$p_* = \frac{1}{2}(p_1 + p_a), \tag{30}$$

where s_{ij} is the distance between i and j, which can be determined from the refinement levels of the respective cells. If cell 2 is at the same refinement level as cell 3, then $s_{2a}/s_{23} = s_{3a}/s_{23} = 1/2$. If cell 2 is more refined than cell 3, then $s_{2a}/s_{23} = 1/3$ and $s_{3a}/s_{23} = 2/3$. Finally, if cell 2 is less refined than cell 3, then $s_{2a}/s_{23} = 2/3$ and $s_{3a}/s_{23} = 1/3$. Again, due to the restrictions in refinement imposed by the grid-generation code, these are the only values the distance fractions can attain. Equations (29) and (30) produce an expression for the interpolated value which is exactly the same as if the shape functions were used.

If the position of interpolation of the cell-centered variable lies at the vertex of a cell, as shown in Fig. 5b, the four cells sharing the vertex are used in the interpolation. The variables

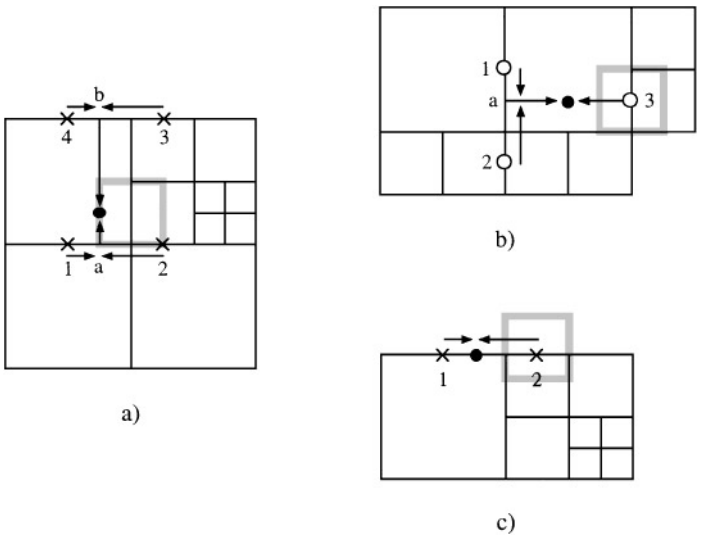

FIG. 6. Interpolations of face-centered variables: (a) interpolation to a location inside a cell using four nodes; (b) interpolation to a location inside a cell using three nodes; (c) interpolation along a face.

diagonal to each other are interpolated linearly to obtain two values at the vertex, and the two values are averaged; i.e.,

$$p_* = \frac{1}{2}\left[\left(\frac{s_{*3}}{s_{13}}p_1 + \frac{s_{*1}}{s_{13}}p_3\right) + \left(\frac{s_{*4}}{s_{24}}p_2 + \frac{s_{*2}}{s_{24}}p_4\right)\right]. \tag{31}$$

This interpolation is consistent with the implementation of the interpolation within a cell defined above.

The interpolations of the unknowns located at the cell faces (u and v) can also be performed in a combination of linear interpolations and averages. The most complicated case is sketched in Fig. 6a, where the interpolated value is found as follows:

$$v_a = \frac{s_{2a}}{s_{12}}v_1 + \frac{s_{1a}}{s_{12}}v_2 \tag{32}$$

$$v_b = \frac{s_{4a}}{s_{34}}v_3 + \frac{s_{3a}}{s_{34}}v_4 \tag{33}$$

$$v_* = \frac{s_{b*}}{s_{ab}}v_a + \frac{s_{a*}}{s_{ab}}v_b. \tag{34}$$

The rest of the cases are subsets of these operations (e.g., Figs. 6b and 6c). In the cases where three values are used in the interpolation, as in Fig. 6b, the interpolation coincides with that of a linear triangular element.

Similar interpolations are also used to transfer state variables from parent to children and vice versa in the adaptation of the grid. When a cell is unrefined the cell-centered variables are interpolated using Eq. (31) (where 1, 2, 3, 4 correspond to the NW, SW, SE, NE children, respectively), and the face variables are interpolated using an equation similar to Eq. (32) for each face of the parent cell (where the values on the children's faces which overlap with the parent's faces are used in the interpolation). When a cell is refined, Eqs. (29), (30) are used

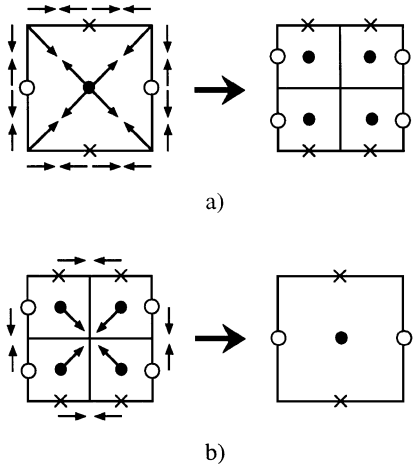

FIG. 7. Interpolations during grid adaptation: (a) refinement (arrows that do not begin at a node use information from neighboring cells); (b) unrefinement.

four times (once with respect to each vertex of the parent cell) to obtain the cell-centered variables of the children, and equations similar to (32)–(34) (depending on the neighbors of the parent cell) are used to obtain the face variables of the children. These operations are sketched in Fig. 7.

3.5. *Boundary Conditions*

The current capabilities of the code are such that the boundaries must align with the Cartesian grid; therefore, only vertical and horizontal boundaries are allowed. Due to the staggered mesh arrangement, the x-component of the velocity has boundary values which lie on the vertical boundaries and are half a cell past horizontal ones, while the opposite is true for the y-component of the velocity. Pressure boundary values are needed at ghost points half a cell past all the boundaries.

Velocity components which lie on the boundary are set to a constant at inflow and solid boundaries. At outflow and far field boundaries, these velocity components are evaluated in such a way that mass is conserved in the cell in which they lie. For the convective terms, the ghost values of the velocities at inflow and solid boundaries are computed so that a linear interpolation with the closest internal value gives the value specified at the boundary (reflection boundary conditions). These values, however, are not used in the computation of the viscous terms to avoid deterioration in the global accuracy of the solution [31]; instead, the gradient of the velocity components at the wall is derived by using a second-order, one-sided difference approximation. Substitution into the central difference term used for the viscous terms produces the following ghost value for v,

$$v_g = \left(\frac{2\Delta x_{gb}}{\Delta x_{21}}\frac{\Delta x_{2b}}{\Delta x_{1b}} - \frac{\Delta x_{gb}}{\Delta x_{1b}}\right)v_1 - \frac{2\Delta x_{gb}}{\Delta x_{21}}\frac{\Delta x_{1b}}{\Delta x_{2b}}v_2 + \left[1 + \frac{\Delta x_{gb}}{\Delta x_{1b}} - \frac{2\Delta x_{gb}}{\Delta x_{21}}\left(\frac{\Delta x_{2b}}{\Delta x_{1b}} - \frac{\Delta x_{1b}}{\Delta x_{2b}}\right)\right]v_b, \tag{35}$$

where v_b, v_1, and v_2 are the values of the y-component of the velocity at the boundary, a cell just inside the boundary, and the second interior cell in a direction normal to the

boundary, respectively; Δx_{ij} is the distance between nodes i and j. A similar equation can be written for u. Finally, the ghost values of the velocities at outflow and far field boundaries are set equal to the closest interior value. This is equivalent to saying that the gradient of the velocity normal to the boundary is equal to zero.

Since the equations for the auxiliary velocities (Eqs. (14) and (15)) are consistent with the momentum equation (Eq. (2)), the boundary conditions described above are also valid for the auxiliary velocities. Using this fact, the boundary conditions for the pressure at solid and inflow boundaries can be derived by setting up the correction equations (Eqs. (19) and (20)) at the boundary and canceling the appropriate terms. Assuming $F^k_{\text{visc}} - F^{k-1}_{\text{visc}} \approx 0$ at the boundaries, and that at time $= 0$, $p_g = p_1$, then the pressure at the ghost cell can be set equal to the internal value. At outflow and far field boundaries, the value of the pressure is specified.

The computational molecule of Fig. 4 requires ghost values for the density and the viscosity. These values are simply set equal to the value at the cell closest to the boundary. Far from the interfaces, this corresponds to an extension of the domain. Near the interfaces, this corresponds to reflecting boundary conditions (in the problems studied, the interfaces only approach solid boundaries).

The boundary conditions for the pressure and the velocity are computed implicitly. This is done by expressing the boundary values as linear functions of the internal variables, e.g.,

$$u_{\text{bdry}} = a_0 + \sum_{i=1}^{N} a_i u_i, \tag{36}$$

where u_i are the internal variables and a_i are constant coefficients, and storing the coefficient, as well as pointers to the structures containing the internal variables (which also contain pointers to the matrix location of the variables) during the evaluation of the boundary conditions. The boundary values are then directly incorporated into the algorithm described in the previous sections.

3.6. *Extension to Axisymmetric Geometries*

The discrete divergence of a quantity, $\mathbf{Q}$, in axisymmetric geometries (where the line of symmetry is $x = 0$) can be written as

$$\frac{1}{y_c V} \sum_{\text{faces}} \{\mathbf{Q} \cdot \vec{n} y_f \Delta S\}, \tag{37}$$

where y_f and y_c are the y-coordinates at the midpoint of the face and at the centroid of the control volume, respectively. Using this definition, the finite volume formulation of the equations can easily be extended to axisymmetric geometries as follows: Eqs. (11) and (12) become

$$\sum_{\text{faces}} (\vec{v} \cdot \vec{n} y_f \Delta S) = 0 \tag{38}$$

$$\begin{aligned} \frac{(\rho \vec{v} y_c V)^{n+1} - (\rho \vec{v} y_c V)^n}{\Delta t} = &- \sum_{\text{faces}} (\bar{\bar{F}}_{\text{conv}} \cdot \vec{n} y_f \Delta S)^n - \sum_{\text{faces}} (\bar{\bar{F}}_{\text{press}} \cdot \vec{n} y_f \Delta S)^{n+1} \\ &+ \theta \sum_{\text{faces}} (\bar{\bar{F}}_{\text{visc}} \cdot \vec{n} y_f \Delta S)^{n+1} + (1-\theta) \sum_{\text{faces}} (\bar{\bar{F}}_{\text{visc}} \cdot \vec{n} y_f \Delta S)^n \\ &+ \vec{f} V + \vec{H} V, \end{aligned} \tag{39}$$

where

$$\vec{H} = \begin{bmatrix} 0 \\ \dfrac{p_n + p_c}{2} - \dfrac{(\mu_n + \mu_c)v_c}{y_c} \end{bmatrix}. \tag{40}$$

With respect to the derivation of the PISO algorithm described in Section 3.4.2, the pressure source team is included in $\mathcal{F}_{\text{press}}$, while the velocity source term is treated as $\mathcal{F}_{\text{visc}}$ and is included in the diagonal terms, a_c. With this redefinition of terms, and including y_c in the time derivative terms and y_f in the summation of fluxes, Eqs. (14), (19), and (20) have the same form as before. The pressure equations (21), (22) include an additional y_f as a multiplier for the terms in the summations.

To generalize the code, the algorithm was actually implemented including all of the terms needed for the axisymmetric equations. To simulate two-dimensional problems, y_f and y_c are set to 1 and $\vec{H}$ is set to $\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$.

The expression for the surface force can also be easily extended to axisymmetric geometries. For example, Eq. (5) becomes

$$\vec{f}_{S_i} = \frac{1}{y_c}[(\sigma \vec{e}_t y)_{i,1} - (\sigma \vec{e}_t y)_{i,0} - \sigma_c \Delta s_i \vec{j}]. \tag{41}$$

4. VALIDATION AND RESULTS

In this section the results of simulations of several problems are presented. The first two problems are the deformation of a two-dimensional drop under shear flow and the oscillations of an axisymmetric drop driven by surface tension. These problems are used to validate the code and to provide estimates for the efficiency and convergence behavior of the method. For these test cases, the surface force is due to the interface between fluids and is defined and discretized as in Eqs. (4), (5), and (41). The last two problems are a typical single-cell-mechanics experiment (modeled as axisymmetric) and the adhesion of two biological cells under shear flow (modeled as two-dimensional). The resultant surface force for each of these cases will be described in Sections 4.3 and 4.4. These two problems demonstrate the capabilities of the method to study the mechanics and adhesion of biological cells.

4.1. *Deformation of a 2-D Drop under Shear*

In this problem, a two-dimensional drop deforms in a linear flow field caused by the relative motion of parallel walls (the effects of gravity are neglected). The drop is initially circular and the flow field is generated by moving the upper wall while keeping the bottom one stationary. The east and west boundaries are periodic. This test case is relevant to the study of cell adhesion since the same boundary conditions can be used to model the parallel plate flow chambers commonly used experimentally. The parameters of the problem are set as follows: the velocity of the upper wall, $v_w = 3$ cm/s; the density ratio between the drop and the outer fluid, $\rho_d/\rho_o = 2$, the viscosity ratio between the drop and the outer fluid, $\mu_d/\mu_o = 1$; the surface tension coefficient, $\sigma = 1\,\text{g/s}^2$; and the initial radius of the drop, $r_d = 1$cm. The computational domain is a square with $-4 \le x \le 4$ and $-4 \le y \le 4$. On output, the shape of the drop, the pressure contours, and the contours of the y-component

of the velocity (v) are examined. The y-component of the velocity was chosen instead of the speed because the drop distorts the linear velocity profile only slightly, and the effects are best seen in the y-component.

The results from this problem were compared to the results using a similar method implemented on structured grids (this latter implementation has been validated and used extensively [e.g., [15, 32, 33]]). The problem was run on each code using identical grids of 128 by 128 cells (all the cells in the code of this work were set to a refinement level of 8) on a square domain 8 cm in length, and excellent agreement was found between the results [33]. The adaptively refined code was found to use about 4 times more computational time per cell than the structured code due to the overhead of local refinement. However, since the adaptively refined code is capable of achieving similar results as the structured code with about 16 times fewer computational cells (as will be discussed in the next paragraph) the overall efficiency of the adaptively refined code is about 4 times better.

The effects of refinement on the solution were studied by simulating this problem on three different grids: one grid with local refinement near the interface (largest refinement level of 8, and smallest of 5), and two without (a fine mesh where all the cells are at refinement level 8, and a coarse mesh where the cells are at 5). The coarse mesh has 256 cells, the locally refined mesh contains 998, and the fine mesh has 16,384. These grids are shown in Fig. 8. The corresponding pressure and y-velocity contours at time $= 2.4$ s are shown in Fig. 9. This figure shows how the solution obtained using the refined mesh is much closer to the solution obtained using the finer mesh than the coarser mesh, particularly near the location of highest curvature in the drop. The pressure is distorted slightly by the refinement interfaces, where there is a slight loss of consistency in the discretization of second derivatives due to the linear interpolations used. The value of the pressure gradient, however, is only affected locally (right at the refinement interface), and the velocities are not affected significantly. The inconsistency at the refinement interfaces could be corrected by using quadratic interpolations instead of linear, but these interpolations would complicate the setup, would have to be linearized for the PISO algorithm, and could lead to positivity problems [34].

Figure 10 shows an overlay of the shape of the drops at time $= 2.4$ s for each grid. The shape obtained using the coarser mesh is more elongated, while the other two are almost indistinguishable.

4.2. *Oscillating Drop*

In this problem, the surface of a drop is perturbed and oscillates due to surface tension forces. These oscillations have been analyzed mathematically by Lamb [35]. To simulate the problem, the drop was initialized to an ellipsoid and far field conditions were enforced at all boundaries of the domain. The initial conditions selected correspond to the simplest and most important mode of oscillation (a wave number of two). This problem has been used as a test case in the literature by others modeling multiple fluids [18, 32]. The existence of an analytical solution provides a rigorous test especially for the interface tracking method and its implementation. The parameters were set as follows: the density inside the drop, $\rho_i = 1$ g/cm^3; the density of the external fluid, $\rho_o = 0.01$ g/cm^3; the viscosity inside the drop, $\mu_i = 0.01$ poises (g/cm s); the viscosity of the external fluid, $\mu_o = 0.001$ poises, the undisturbed radius of the drop, $R = 1$ cm; and the amplitude of the perturbation, $A = 0.03$ cm. This problem was solved in a square computational domain with $-2.4 \le x \le 2.4$ and

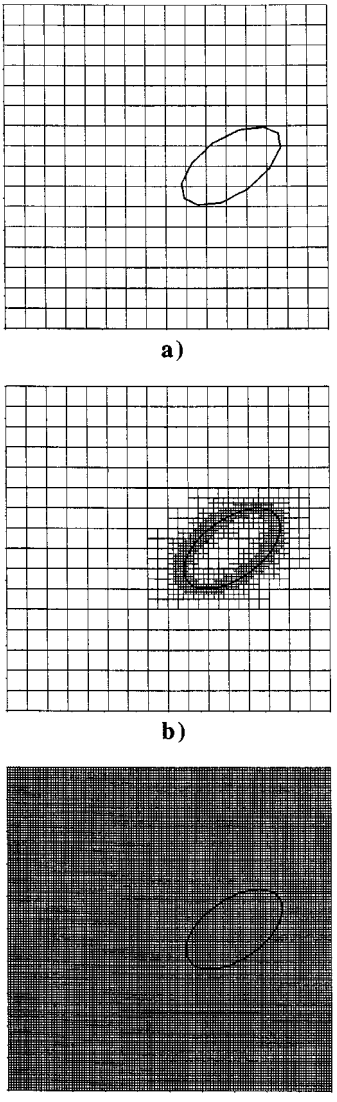

FIG. 8. Grids used to test the advantages of using refined grids (at time = 2.4 s): (a) RL = 5, 5 (all cells are at refinement level 5, $h = 0.5$); (b) RL = 5, 8 (the refinement level varies from 5 to 8, h varies from 0.5 to 0.0625); and (c) RL = 8, 8 (all cells are at refinement level 8, $h = 0.0625$).

$-2.4 \leq y \leq 2.4$, using adaptively refined meshes similar to that shown in Fig. 8b (but with different refinement levels).

The position of the highest point on the drop was tracked as a function of time, and the results using a 4, 8 mesh for an axisymmetric drop with $\sigma = 5\,\mathrm{g/s^2}$ (dyn/cm) are shown in Fig. 11. The period of the oscillations was computed by averaging the distances between peaks in this figure. The results for two-dimensional and axisymmetric drops with different surface tension coefficients ($2\,\mathrm{g/s^2}$ and $5\,\mathrm{g/s^2}$) are compared to the analytical solutions in Table 1. As seen in this table, the numerical solutions are in excellent agreement with the analytical ones.

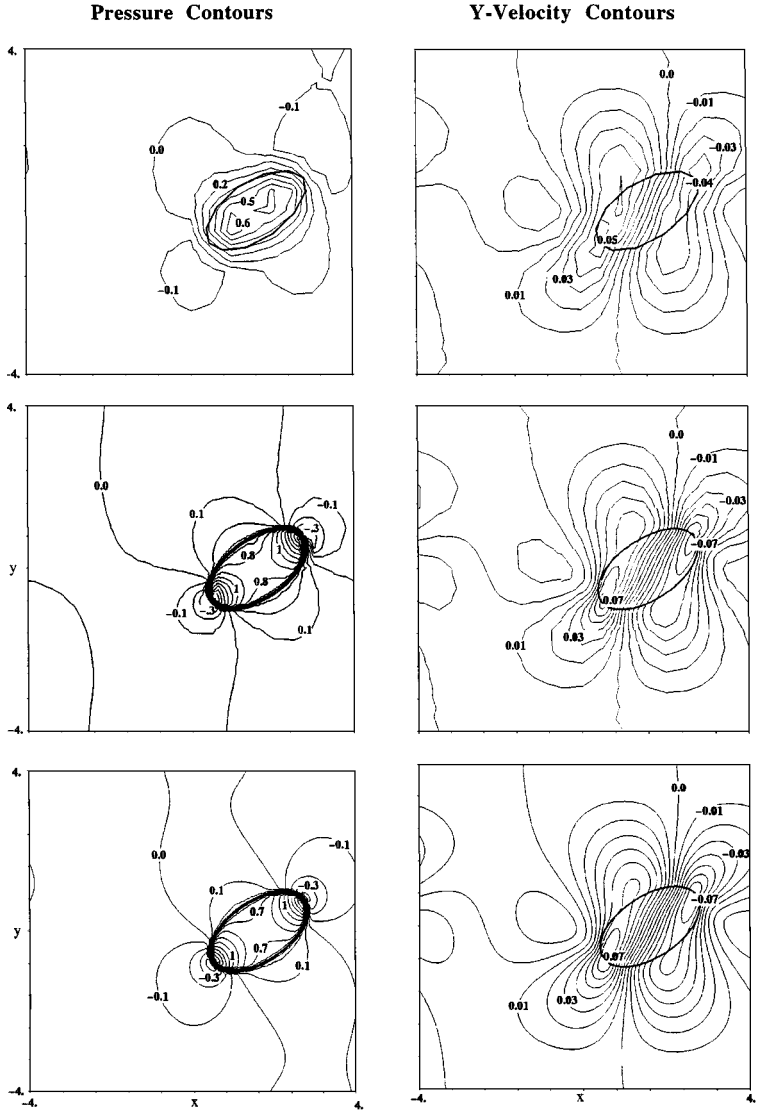

FIG. 9. Pressure and *y*-velocity contours obtained using a 5, 5 grid (top row), a 5, 8 grid (middle row), and an 8, 8 grid (bottom row); time = 2.4 s.

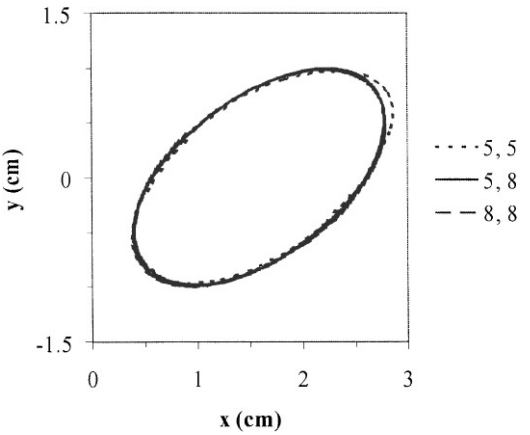

FIG. 10. Shapes of the drop at time = 2.4 s obtained using a 5, 5 grid, a 5, 8 grid, and an 8, 8 grid.

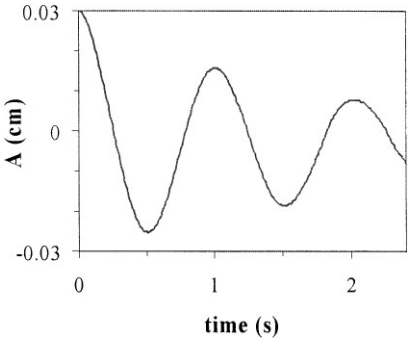

FIG. 11. Amplitude, $A(t)$, of the oscillations of a drop as a function of time for an axisymmetric drop with $\sigma = 5\,\mathrm{g/s^2}$. The remaining parameters are $\rho_i = 1\,\mathrm{g/cm^3}$, $\rho_o = 0.01\,\mathrm{g/cm^3}$, $\mu_i = 0.01$ poises, $\mu_o = 0.001$ poises, $R = 1\,\mathrm{cm}$, and $A(0) = 0.03\,\mathrm{cm}$.

This test case was next used to examine the convergence rate of the solution with respect to mesh size. In order to isolate the effect of the mesh size on the accuracy, the time step was set to a constant ($= 0.01$ s). The two-dimensional problem with $\sigma = 5\,\mathrm{g/s^2}$ was run on four different meshes, all with a lowest refinement level of 4, and the highest refinement level varied from 5 to 8. The period of the oscillations computed on each grid, the relative error, and the mesh size are tabulated on Table 2. The $\log_{10}$ of the error is plotted as a function of the $\log_{10}$ of a measure of the mesh size, $h_{\mathrm{approx}} = (\text{number of cells})^{-1/2}$, in Fig. 12. This figure shows that the error decreases monotonically with decreasing mesh size and that the convergence of the solution with respect to mesh size is close to linear (except for the rightmost segment, the endpoint of which is from a very coarse mesh). This convergence behavior is the best typically seen when interfaces are resolved on stationary grids. Tracking the interface explicitly reduces the error, but does not change the order. The inconsistencies near refinement interfaces discussed in the previous section also contribute to a decrease in the overall convergence rate of the solution. However, this drawback is offset by the fact that the same degree of accuracy can be obtained with significantly fewer cells using local refinement.

4.3. *Simulation of Cell-Entry Micropipet Experiment*

One of the most common techniques used to study the deformation of cells is the cell-entry micropipet assay. This assay consists of pulling a portion of the cell or the entire cell into a

TABLE 1
Comparison of the Period of the Oscillations of a Drop Obtained from the Simulations to the Analytical Solutions [35]

$\sigma\,(\mathrm{g/s^2})$	$\lambda_{\mathrm{numerical}}$	$\lambda_{\mathrm{analytical}}$	Relative error
	Two-dimensional		
2	1.847	1.814	1.8%
5	1.161	1.147	1.2%
	Axisymmetric		
2	1.605	1.576	1.8%
5	1.012	0.997	1.5%

TABLE 2
Error in the Period of the Oscillations of a Two-Dimensional Drop as a Function of Mesh Size ($\lambda_{analytical} = 1.147$, $\sigma = 5$ g/s^2, $h_{max} = 0.3125$)

Refinement levels	Number of cells	$h_{approx} = (\#\text{ cells})^{-1/2}$	h_{min}	$\lambda_{numerical}$	Relative error (%)
4, 5	160	0.0790569	0.15625	1.244	8.45
4, 6	304	0.0573498	0.078125	1.174	2.35
4, 7	676	0.0384577	0.039063	1.166	1.66
4, 8	1456	0.0262106	0.019531	1.161	1.22

micropipet of set diameter with a constant suction pressure. The experiments are recorded on video, and the distance between the leading edge of the cell and the tip of the pipet (projection length or L_p) is plotted as a function of time. From this plot and by assuming a constitutive relation between applied stress and cellular deformation, mechanical parameters for the cell can be calculated. This technique has been used in conjunction with several models to study artificial lipid vesicles [11] and several types of cells including red blood cells [11], leukocytes (neutrophils [36–39] and granulocytes [40]), and lymphocytes [41].

This assay is simulated in axisymmetric coordinates using parallel walls to represent the micropipet and initializing the cell to a sphere (cell without a nucleus) or two concentric spheres (cell with a nucleus) near the entrance of the pipet. Outflow boundary conditions are enforced at the end of pipet and far field conditions are implemented at the remaining boundaries. The resultant force on the cellular membranes is computed by assuming the membranes hold isotropic tensions only, i.e.,

$$\sigma = T_0 + K\frac{\Delta A}{A_0}, \tag{42}$$

where T_0 represents the interfacial tension due to the phase boundary between hydrophobic membrane components and water based surroundings, and the second term is due to the

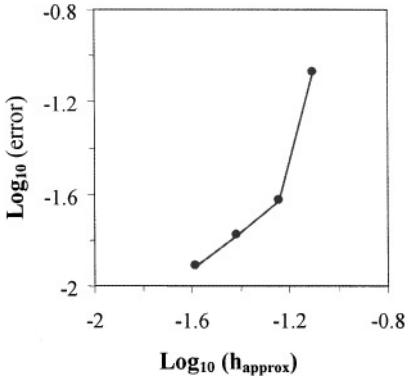

FIG. 12. Logarithmic plot of the error in the period of the oscillations as a function of an approximate mesh size: (number of cells)$^{-1/2}$. The number of cells for the (4, 5), (4, 6), (4, 7), and 4, 8 grids are 160, 304, 676, and 1456, respectively.

elasticity of the plasma membrane and associated cortex [42]. K is the isothermal area compressibility modulus, and $\Delta A/A_0$ is the fractional area change of the membrane with respect to the unstressed area, A_0. This unstressed area is set by the initial discretization of the membrane. Several biological membranes have been successfully modeled using Eq. (42), at least in some circumstances [11, 39]. The resultant surface force is then

$$\vec{f}_S = \vec{f}_{\text{mech}} = \frac{\partial(\sigma \vec{e}_t)}{\partial s}\, ds \tag{43}$$

and is discretized using Eq. (41). A similar model was used by Jan to study the effects of surfactants in contaminated bubbles [26].

The parameters in the simulations are set in accordance with the experiments of Needman and Hochmuth [43]: the cell radius is 4 μm, the pipet radius is 2 μm, and suction pressures of 1 and 2 kPa are used. The density of all fluids is set to a constant, $\rho = 10^7 \text{g/cm}^3$ (this constant is seven orders of magnitude higher than the observed density to allow the use of a larger time step, but since the flow is near the Stokes limit, the effect of this larger density on the solution is minimal [30]). The viscosity of the outer media is set to four orders of magnitude smaller than the cytoplasmic viscosity, $\mu_{\text{cytoplasm}}$ (larger viscosity ratios did not change the solution significantly [30]).

Three mechanical models for the cell are used: (a) a Newtonian fluid surrounded by a prestressed cortical shell ($K = 0$), (b) a Newtonian fluid surrounded by an elastic membrane, and (c) a Newtonian fluid with a more viscous nucleus inside. The first mechanical model has been used by experimentalists to analyze micropipet data for leukocytes [39, 40, 43] and it predicts the behavior of these cells at large deformations, but not at small deformations. Parameters on the same order as the values obtained by these experimentalists were used for the first run: $T_0 = 0.04$ dyn/cm, and $\mu_{\text{cytoplasm}} = 10^3$ dyn s/cm^2. Figure 13 shows snapshots in time of the pressure and speed contours of a cell entering a pipet with a suction pressure of 2 kPa using this model. As expected, two flow regimes are observed. In the first regime (time $= 0.003$ s in Fig. 13) the cell is not in contact with the pipet and the flow is controlled by the fluid outside the cell. A pressure gradient exists inside the pipet, and the velocity of the fluid is a maximum in the gap between the walls at the tip of the pipet and the cell. When the cell makes contact with the pipet, the flow inside the pipet becomes plugged and gains control of the problem (time $= 0.03$ s). The experimental data are obtained once this second flow regime is established. The pressure of the fluid inside the pipet (outside the cell) becomes constant, a pressure gradient develops inside the cell, and the maximum pressure occurs at the tip of the pipet where the cell makes contact; the maximum speed decreases and the flow between the pipet wall and the cell surface is minimal. These trends continue throughout the remaining frames of Fig. 13. At later times, contrary to experimental evidence (e.g., [43]), the trailing edge of the cell bulges into the pipet while a peripheral portion of the cell remains near the tip. Figure 14 shows how a cell with a higher surface tension enters the pipet entirely as is seen experimentally [(e.g., [43]). However, the surface tension for leukocytes has been measured to be as low as in the simulation of Fig. 13, suggesting that the Newtonian drop model does not completely capture the mechanical behavior of these cells even at large deformation.

One source of numerical error in these simulations is the current implementation of the velocity adjustments performed to avoid contact of the interface with the wall (discussed in Section 3.3). At each time step, these velocity adjustments cause a small loss in the volume of the cell (about 0.0001%). However, due to the large number of times that these

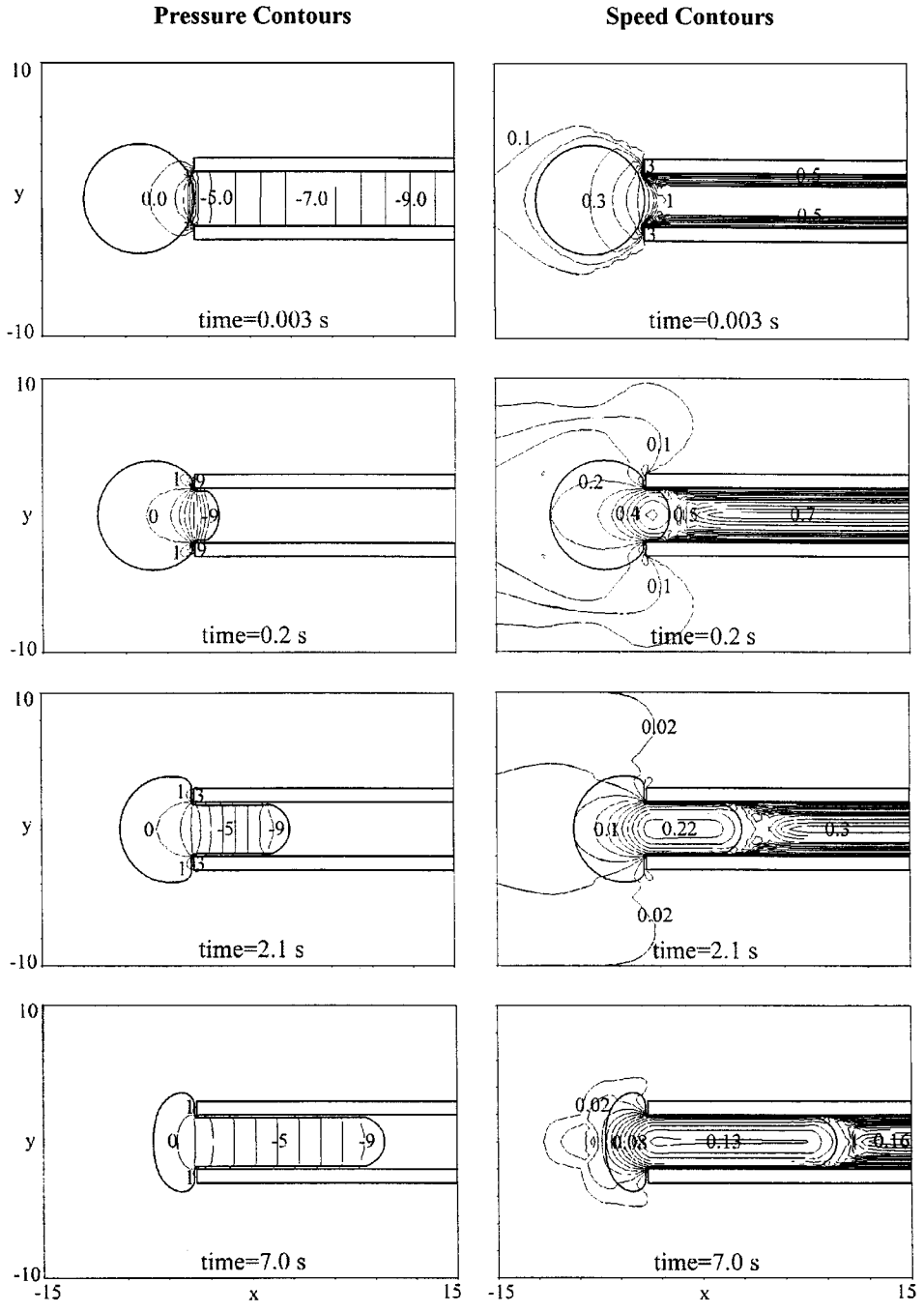

FIG. 13. Pressure (left) and speed (right) contours of a cell modeled as a Newtonian drop entering a pipet with a suction pressure of 1 kPa. The pressure is in units of $\mathrm{mg}/(\mu\mathrm{mds}^2) = 10$ kPa and the speed has units of μm/ds. The mechanical parameters of the cell are $T_0 = 0.04$ dyne/cm, $K = 0$, and $\mu_{\mathrm{cytoplasm}} = 10^3$dyne s/cm^2. The grid used refinement levels of 6, 8, and 9 (6 for the background, 8 for intersected cells, and 9 for proximity of the interface to the pipet).

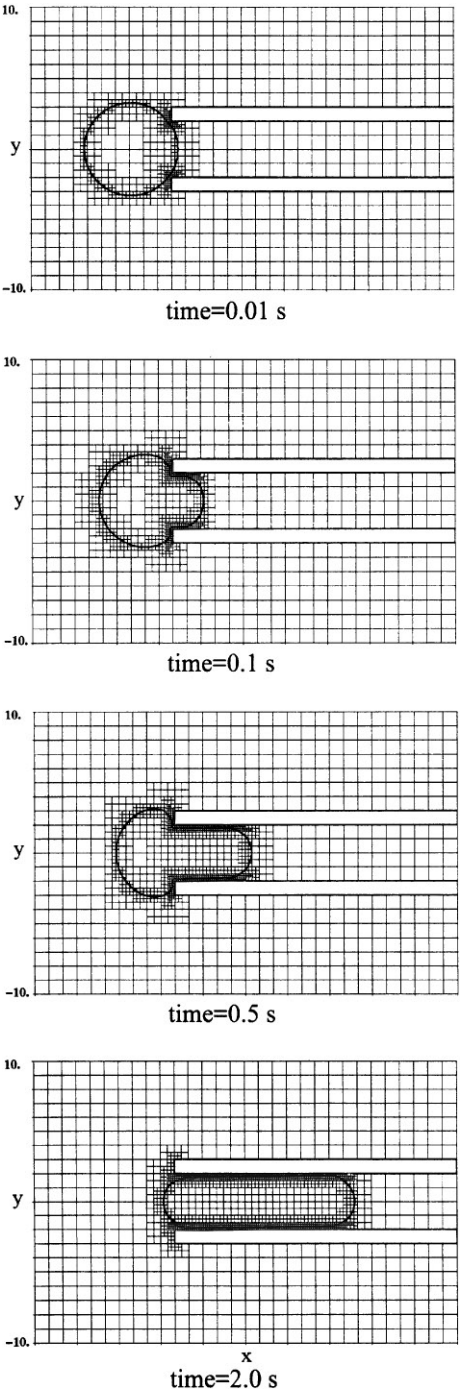

FIG. 14. Cell modeled as a Newtonian drop completely entering a pipet with a suction pressure of 2.5 kPa. The mechanical parameters of the cell are $T_0 = 1$ dyne/cm, $K = 0$, and $\mu_{\text{cytoplasm}} = 10^3$ dynes/cm^2. RL = 6, 8, 9 (6 for the background, 8 for intersected cells, and 9 for proximity of the interface to the pipet).

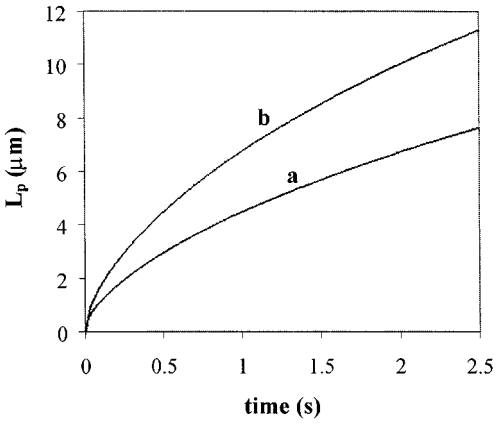

FIG. 15. Projection length vs time for a cell modeled as a Newtonian drop (mechanical parameters as in Fig. 13) aspirated at two different suction pressures: (a) 1 kPa; (b) 2 kPa.

operations are performed in the micropipet simulations, this error accumulates leading to volume losses of about 15%. Decreasing the size of the smallest computational cells (the ones between the interface and the pipet) by half reduces this volume loss by more than 50%. To avoid increasing the number of computational cells, the velocity adjustments on the interface must be communicated to the incompressible fluids. One way to achieve this is described in Section 3.3.

The Newtonian-drop model is used to compare the effects of suction pressure on the projection length of the cell into the pipet (Fig. 15). Figure 15 shows that doubling of the suction pressure increases the rate of entry by about two. This trend was seen experimentally by Needham and Hochmuth [43].

The other two mechanical models were compared to the Newtonian-drop model using a suction pressure of 2 kPa. In the model with the elastic membrane the parameters are $T_0 = 0.04$ dyn/cm, $K = 100$ dyn/cm and $\mu_{\text{cytoplasm}} = 10^3$ dyn s/cm^2. In the model of a cell with a nucleus the parameters for the cytoplasm and external membrane are $\mu_{\text{cytoplasm}} = 10^2$ dyn s/cm^2, and $T_0 = K = 0.04$ dyn/cm; and the parameters for the nucleus and the nuclear membrane are $\mu_{\text{nucleus}} = 10^3$ dyn s/cm^2, $T_0 = 0.04$ dyn/cm and $K = 2$ dyn/cm. The value of the elastic modulus for the second model is on the order of the values measured for lipid bilayers and red blood cells [11]. The parameters in the last model are on the order of those measured for lymphocytes [41]. The different mechanical models affect both the shape of the cells and the contact angle between the tip of the pipet and the outer portion of the cell, as seen in Fig. 16. Figure 17 shows the projection length of the cell as a function of time for the three models. As shown in this figure, compared to the Newtonian-drop model the elasticity of the membrane retards the entry of the cell into the pipet and this effect increases as the cell becomes more elongated. This behavior is similar to the behavior of red blood cells entering a pipet [44]. The cell with a less viscous cytoplasm and a more viscous nucleus enters the pipet more rapidly initially, and slows down as the nucleus starts deforming into the pipet. These changes in rate of entry are observed in micropipet experiments with leukocytes ([e.g., [43]). However, due to the low surface tension and the lower cytoplasmic viscosity in this model, at later times a portion of the cell extends past the tip outside the pipet (contrary to experimental observations). There is evidence that in some cells cytoplasmic elements

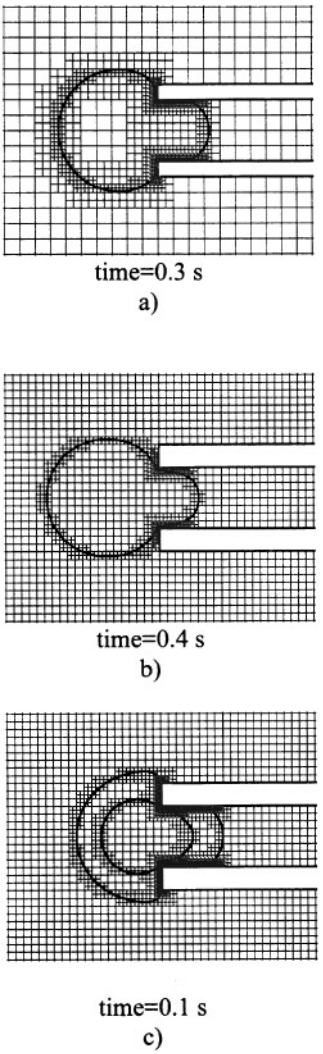

FIG. 16. Comparison of the shape of a cell entering a micropipet with a suction pressure of 2 kPa using three different mechanical models: (a) a Newtonian fluid with constant surface tension (RL = 6, 8, 9); (b) a Newtonian fluid enclosed by an elastic membrane (RL = 7, 8, 9); and (c) a Newtonian fluid with a more viscous nucleus (RL = 7, 8, 10; a refinement level of 10 is also used for proximity of the interfaces). (See text for problem parameters.)

connect the outer membrane to the nuclear membrane [45] and such elements could prevent the outer membrane from extending outside the pipet. These cytoplasmic elements could be incorporated in the methods of this paper using elastic links which connect discrete points on the cell membrane to discrete points on the nuclear membrane.

4.4. *Simulation of Cells Adhering Under Flow*

Specific cell adhesion frequently occurs in circulatory systems. In this section a two-dimensional version of this case is presented. As mentioned in the Introduction, biological cells adhere by forming bonds between mobile membrane molecules, while the membranes

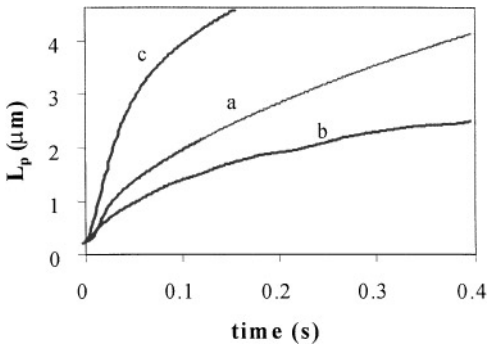

FIG. 17. Comparison of the projection length of a cell entering a pipet as a function of time using three different mechanical models (the letters correspond to the models in Fig. 16).

deform and colloidal forces act to repel the cells. The resultant surface force for each cell is

$$\vec{f}_S = \vec{f}_{\text{mech}} + \vec{f}_{\text{bonds}} + \vec{f}_{ns}, \tag{44}$$

where $\vec{f}_{\text{mech}}$ is due to the membrane's resistance to deformation (as in Eqs. (42) and (43)), $\vec{f}_{\text{bond}}$ is due to molecular bonds and $\vec{f}_{ns}$ results from the colloidal interactions. The number of bonds per segment is found by discretizing the reaction–diffusion equations on the interfaces, using reaction rates which are functions of the distance between the membranes (as proposed by Dembo *et al.* [6]). Since the membranes move arbitrarily and are discretized independently of one another, a segment from one membrane can interact with more than one segment on the other membrane. Hence, the reaction term and the non-specific force are computed by visiting all the segments on the other membrane; if the distance between the centroids of the two segments (l_m) is less than a threshold (l_{th}), the segment pair is given the opportunity to interact. If bonds form during the interaction, a link is established between the segments. The reaction term used in the diffusion–reaction equations is the sum over all the segments on the other membrane. The force exerted by each bond link on each segment is computed as

$$\vec{f}_{\text{bonds}_i} = n_{ik}\kappa_b\left(|\vec{l}_{m_{ik}}| - l_0\right)\frac{\vec{l}_{m_{ik}}}{|\vec{l}_{m_{ik}}|}, \tag{45}$$

where n_{ik} is the number of bonds between the ith segment of one membrane and the kth segment of the other, κ_b is a spring constant assigned to the bond, $\vec{l}_{m_{ik}}$ is a vector from the centroid of the ith segment of one membrane to the centroid of the kth segment of the other, and l_0 is the pre-stressed length of the bond. The idea of treating molecular bonds as springs was proposed by Bell *et al.* [3] and has been used successfully by many other researchers. The force due to the non-specific interactions is computed as

$$\vec{f}_{ns_i} = -\lambda_{ss}\left(\frac{1}{|\vec{l}_{m_{ik}}|} + \frac{1}{\delta}\right)\exp\left(-\frac{|\vec{l}_{m_{ik}}|}{\delta}\right)\frac{\vec{l}_{m_{ik}}}{|\vec{l}_{m_{ik}}|^2}\Delta s_i, \tag{46}$$

where δ is a measure of the combined thickness of the glycocalyx of both cells, and λ_{ss} is a measure of the ease with which the polymer layer between the cells can be compressed. This expression is also due to Bell *et al.* [3]. The total adhesive force on the ith segment is

TABLE 3
Mechanical Properties for the Cells in the Simulations of Cell Adhesion

	Cell 1 (lower)			Nucleus of cell 1			Cell 2 (upper)			Nucleus of cell 2		
Run	μ	T_0	K	μ	T_0	K	μ	T_0	K	μ	T_0	K
1	0.2	100	100	1.0	200	200	0.2	100	100	1.0	200	200
2	0.2	50	50	—	—	—	0.2	150	150	1.0	200	200

Note. The units of viscosity are ng/(μm ms) ($= 10^{-2}$ poises); the units for T_0 and K are ng/ms^2 ($= 10^{-3}$ dyne/cm).

computed by summing the contributions of all other segments on the opposite membrane (summing over k). For details on the implementation of this procedure see [30].

The adhesion of two deformable cells under shear flow is simulated using a square domain 40 μm in length, with periodic east and left boundaries. No-slip boundary conditions are enforced at the north and south boundaries, where the velocities are set to -25 and 25 μm/ms, respectively. The resulting shear is comparable to that in an arteriole [46]. The cells are initialized to circles 4 μm in radius and in very close proximity so molecular interactions can commence immediately (i.e., $\min(l_{m_{ik}}) \leq l_0$). One cell is placed higher than the other so the velocity gradient moves them past one another. Two cell pairs are examined: (i) identical nucleated cells, and (ii) one nucleated cell with a more flaccid cell lacking a nucleus. The mechanical properties of the cells and their nuclei are listed in Table 3. The density of all fluids is set to 0.1 ng/μm^3 ($= 100$ g/cm^3) and the viscosity of the immersing fluid is set to 0.1 ng/(μm ms) ($= 10$ dyn s/cm^2). The adhesion parameters (reaction rates, spring parameters for the bonds, receptor number, diffusivity, colloidal parameters) are set to typical values from the literature adjusted for two-dimensional computations when appropriate.

Figure 18 shows a close-up, near the area of contact, of the adaptively refined mesh used for run 2 (a similar mesh was used for the first run). The computational cells in the area of contact between the cells are 128 times smaller than the largest cells in the grid.

Figure 19 shows the evolution of the adhesion for the two cell pairs. As expected, the shear flow causes the cells to move past one another, elongate, and eventually detach. Figures 20 and 21 show plots of the bond density as a function of a coordinate along the membrane of the cells at different times for runs 1 and 2, respectively. The figures show that, as expected, the density of the bonds in a localized region of the surfaces of the cell (the contact area) increases as time passes. The curves for the two cells in run 1 are almost identical. The bond distribution is almost symmetric, with the majority of bonds located near the center of the area of contact. The contact area is smallest, while the maximum concentration of bonds is largest at time $= 0.8$ s; by time $= 1.0$ s all bonds have detached. The curves for the cells in run 2, on the other hand, are significantly different from one another. In this case, the distribution of bonds shifts with time toward one side of the contact area, and the majority of the bonds in the more flaccid cell concentrate in a very small portion of the membrane. At time $= 1$ s, the cells are still attached, but the contact area has begun to decrease.

Figure 22 shows the total number of bonds as a function of time for the two runs. Initially (time $\leqq 0.1$ ms), the total number of bonds is the same for the two cell pairs until the cells have deformed significantly and the cell mechanics begins to play a role. At this time, the curve for the cells with different mechanical properties lags the other. In both curves, the

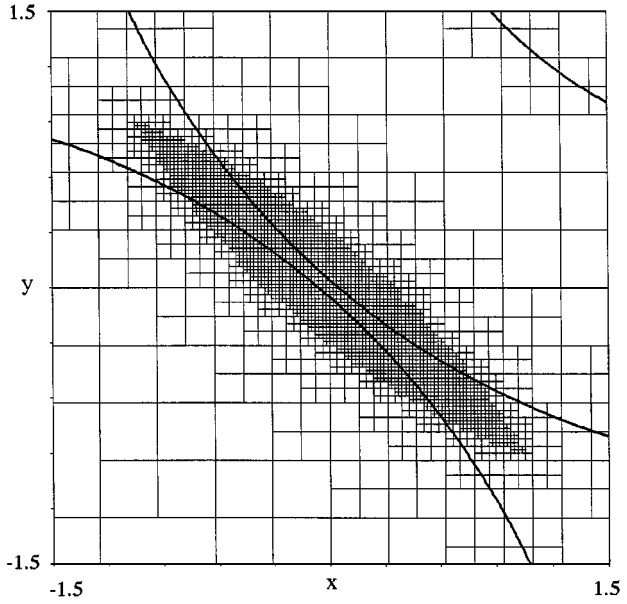

FIG. 18. Representative grid used for the simulations of cell adhesion zoomed near the area of contact.

number of bonds increases to a maximum (the same value for both curves), after which a sharp decrease is observed. Also in both cases, the rate of bond increase is non-monotonic. The sharpest increase in this rate seems to correspond to the time when the shear begins to pull apart the cells (compare the frames in the third row of Fig. 19). However, more studies are needed to determine the cause of this behavior.

5. CONCLUSIONS

One of the major goals of investigators in the area of cell adhesion is to establish a quantitative relationship between molecular parameters and the outcome of cell adhesion. Three major difficulties complicate the search for such a relationship: (i) the relevance of many cellular and extracellular events (external forces, cell mechanics and deformation, diffusion and specific binding of molecules, and non-specific interactions); (ii) the different length scales on which these events occur; and (iii) uncertainties in the mechanics of cell deformation. Similarly, one of the major goals of researchers studying cell mechanics is to elucidate the constitutive equations describing the relationship between the forces experienced by cells and their deformation. A major difficulty encountered by these researchers is that the analysis of experimental data requires *a priori* knowledge of this constitutive relationship.

A multi-fluid code with adaptive refinement, staggered variable arrangement, and arbitrarily deforming fronts tracked with a Lagrangian–Eulerian method was developed to specifically address those difficulties. The generation of different mechanical models for the cell is allowed by a modular treatment of membrane mechanics and fluid bodies. The method was validated with a drop deforming in shear and the oscillations on the surface of a drop. Results agreed with other numerical results and analytical solutions. While thorough convergence rate and efficiency tests were not performed, they are sufficient to establish the

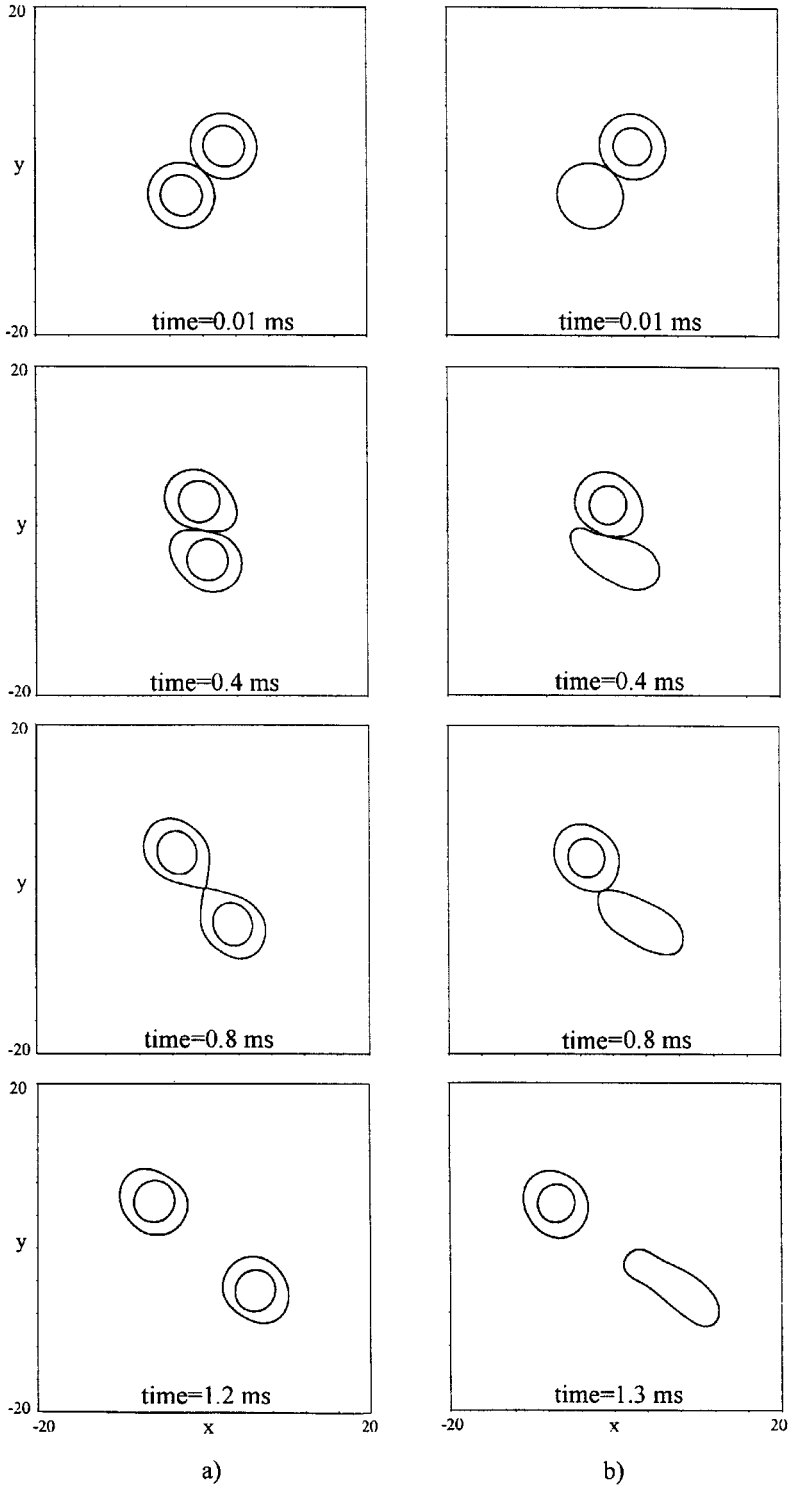

FIG. 19. Evolution of the adhesion of two cell pairs deforming under shear: (a) identical nucleated cells (run 1); (b) a nucleated cell with higher surface tension and elastic modulus (top) and a cell with no nucleus and lower surface tension (bottom) (run 2). RL = 6, 9, 12 (6 for the background, 9 for intersected cells, and 12 for proximity of the interfaces).

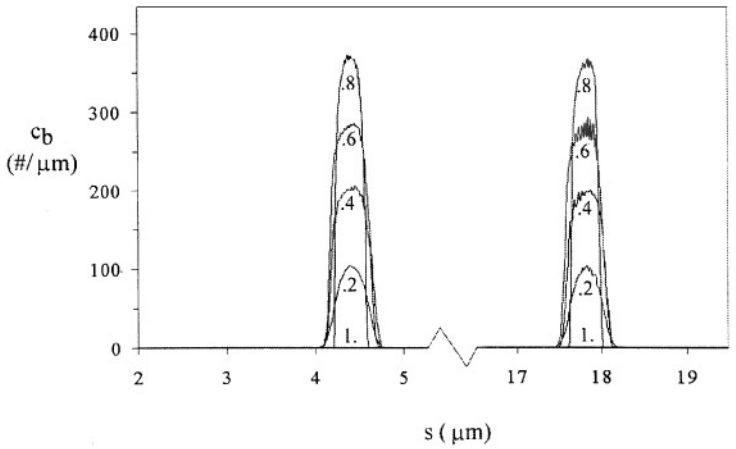

FIG. 20. Temporal evolution of the bond density on the surfaces of two identical cells adhering under shear (run 1). s is a coordinate along the length of the membrane of each cell, beginning at $\theta = 0$ on the undeformed cell and increasing in a counterclockwise direction (for clarity, the curves for each cell were slightly shifted to align the peaks). The curves on the left correspond to the lower cell and the curves on the right are for the upper cell. Each curve is marked with the time in ms.

overall consistency of the method and the net efficiency benefit of local refinement. More rigorous testing of these elements for the grid generation methods used in this work have been done by others [22, 23].

Cell-entry micropipet experiments (a common tool for studying cell mechanics) and the adhesion of cells under flow (a common situation encountered both experimentally and under physiological conditions) were simulated to exemplify the capabilities of the code. Three very different models for the cell mechanics were analyzed with only minor changes to the code. The different models affect both the shape and the rate of entry of cells into a micropipet. The elasticity of the membrane decreases the rate of entry of the cell into the pipet, and this decrease continues as the cell becomes more elongated. This behavior agrees qualitatively with that seen for red blood cells. The presence of a

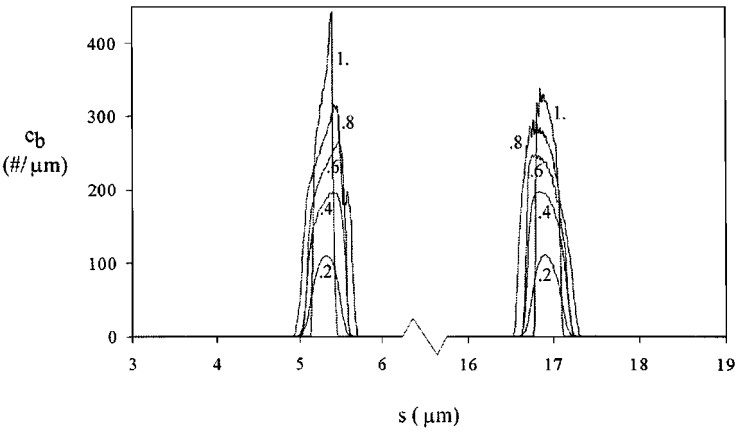

FIG. 21. Temporal evolution of the bond density on the surfaces of two different cells adhering under shear (run 2). s is a coordinate along the membrane as described in Fig. 20.

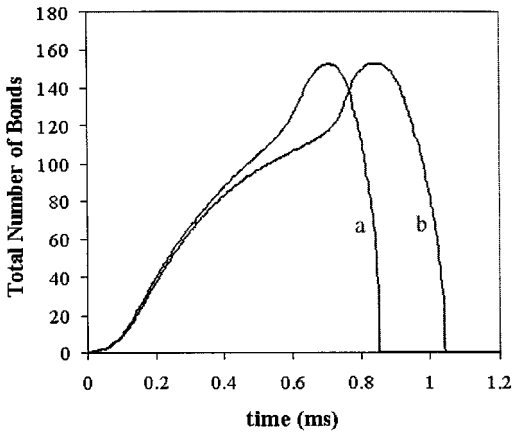

FIG. 22. Comparison of the total number of bonds as a function of time during the adhesion of (a) two identical cells (run 1), and (b) cells with different mechanical properties (run 2).

more viscous nucleus causes a non-monotonic rate of entry of the cell into the pipet. This behavior agrees qualitatively with that seen for neutrophils. The evolution of the shape of the cell as it enters the pipet, as well as the rate of entry of the cell into the pipet, can be compared directly with experimental data. In addition, the code provides information about the spatial variations in fluid pressure and velocity that is not currently available from experimental methods, but that agrees with the expected flow patterns of the problem.

The simulations in Section 4.4 show the capabilities of the code as a tool to study cell adhesion. The code computes the number and distribution of molecular bonds on the cell surface, the distribution of free molecules, and the shape of the cell at every time step. In addition, the code provides information about the spatial variation in pressure and fluid velocity (results not shown). A relationship between events occurring on the scale of the cell diameter and those occurring at the gap between the cells has been successfully established. For the parameters tried, the mechanical properties of the cells were shown to affect the temporal evolution of the contact area, as well as the total number of bonds between the two cells at a given time. This latter effect is very important since the number of bonds can determine the chemical signal between the cells and, therefore, the subsequent cellular response.

This method can be used to study the passive deformation of specific cells and find an appropriate mechanical model. It can be used to decide when simple models for cell adhesion are valid and will allow the impact of cell deformation on detachment force to be determined. Finally, the method can be used to generate information about transmembrane molecular bonds from simple “macroscopic” adhesion assays (e.g., cell separation in shear flow).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Dr. Sami Bayyuk in adapting his grid-generation code for the purposes of this work. This work was supported by ONR Fellowship N000 14-90-J-1778, a Rackham School of Graduate Studies Merit Fellowship, and NIH Cellular Biotechnology Training Grant T32 GM 08353 to G.A., and a National Science Foundation PYI award to J.J.L.

REFERENCES

1. V. Glaser, Work on cell-adhesion-based interactions beginning to bear fruit, *Genetic Engineering News*, 6 (Nov. 15, 1995).
2. P. D. Richardson, Physical and molecular biochemical aspects of cell adhesion: Viewpoints approaching a mutual understanding, in *Principles of Cell Adhesion*, edited by P.D. Richardson and M. Steiner (CRC Press, Boca Raton, FL, 1995).
3. G. Bell, M. Dembo, and P. Bongrand, Cell adhesion: Competition between nonspecific repulsion and specific bonding, *Biophys. J.* **45**, 1051 (1984).
4. E. Evans, Detailed mechanics of membrane–membrane adhesion and separation. I. Continuum of molecular cross-bridges, *Biophys. J.* **48**, 185 (1985).
5. E. Evans, Detailed mechanics of membrane–membrane adhesion and separation. II. Discrete kinetically trapped molecular cross-bridges, *Biophys. J.* **48**, 175 (1985).
6. M. Dembo, D.C. Torney, K, Saxman, and D. Hammer, The reaction-limited kinetics of membrane-to-surface adhesion and detachment, *Proc. R. Soc. London B* **234**, 55 (1988).
7. A. Tozeren, Cell–cell, cell substrate adhesion: Theoretical and experimental considerations, *J. Biomech. Eng.* **112**, 311 (1990).
8. C. Zhu, A thermodynamic and biomechanical theory of cell adhesion. Part I: General formulism, *J. Theor. Biol.* **150**, 27 (1991).
9. D. A., Hammer, and D. A. Lauffenburger, A dynamical model for receptor-mediated cell adhesion to surfaces, *Biophys. J.* **52**, 475 (1987).
10. D. A. Hammer and S. A. Apte, Simulation of cell rolling and adhesion on surfaces in shear flow: General results and analysis of selectin-mediated neutrophil adhesion, *Biophys. J.* **63**, 35 (1992).
11. E. A. Evans, and R. Skalak, *Mechanics and Thermodynamics of Biomembranes* (CRC Press, Boca Raton, FL, 1980).
12. V. C. Mow, F. Guilak, R. Tran-Son-Tay, and R. M. Hochmuth, Eds., *Cell Mechanics and Cellular Engineering* (Springer-Verlag, New York, 1994).
13. H. Zhou and C. Pozrikidis, Deformation of liquid capsules with incompressible interfaces in simple shear flow, *J. Fluid Mech.* **283**, 175 (1995).
14. A. L. Fogelson, Continuum models of platelet aggregation: Formulation and mechanical properties, *SIAM J. Appl. Math.* **52**(4), 1089 (1992).
15. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible flows *J. Comput. Phys.* **100**, 25 (1992).
16. S. A. Bayyuk, K. G. Powell, and B. van Leer, A simulation technique for 2-D unsteady inviscid flows around arbitrarily moving and deforming bodies of arbitrary geometries. AIAA Paper 93-3391-CP, in *AIAA 11th CFD Conference*, Orlando, FL, 1993.
17. W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith, *Computational Fluid Dynamics with Moving Boundaries*, Series in Computational and Physical Processes in Mechanics and Thermal Sciences (Taylor and Francis, Washington, DC, 1996).
18. M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).
19. H. Haj-Hariri, Q. Shi, and A. Borhan, in *Boundary Elements XVII*, edited by C. A. Brebbia, S. Kim, T. A. Osswald, and H. Power (Computational Mechanics Publications, Billerica, MA), p. 637.
20. H. Haj-Hariri, Q. Shi, and Q. Shi. Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers, *Phys. Fluids* **9**(4), 845 (1997).
21. M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. Welcome, An adaptive level set approach for incompressible two-phase flows, in *Proceedings of the ASME Fluids Engineering Summer Meeting: Forum on Advances in Numerical Modeling of Free Surface and Interface Dynamics* (1996), Vol. 3, p. 355.
22. D. De Zeeuw, *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*, Ph.D. thesis University of Michigan, Ann Arbor (1993).

23. S. A. Bayyuk, *Flows about Moving Boundaries and Arbitrary Geometries*, Ph.D. thesis, University of Michigan, Ann Arbor (1997).
24. D. De Zeeuw and K. G. Powell, An adaptively-refined cartesian mesh solver for the Euler equations, *J. Comput. Phys.* **104**, 55 (1992).
25. C. S. Peskin and D. M. McQueen, Modeling prosthetic heart valves for numerical analysis of blood flow in the heart, *J. Comput. Phys*. **37**, 113 (1980).
26. Y. L. Jan, *Computational Studies of Bubble Dynamics*, Ph.D. thesis, University of Michigan, Ann Arbor (1994).
27. R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, *J. Comput. Phys*. **62**, 40 (1985).
28. J. B. Bell, P. Colella, and H. M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys*. **85**, 257 (1989).
29. Y. Saad, *SPARSKIT: A basic tool kit for sparse matrix computations*, Technical Report 90-20, Research Institute for Advanced Computer science, NASA Ames Research Center, Moffet Field, CA (1990).
30. G. Agresar, *A Computational Environment for the Study of Circulating Cell Mechanics and Adhesion*, Ph.D. thesis, University of Michigan, Ann Arbor (1996).
31. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*, Springer Series in Computational Physics (Springer-Verlag, New York, 1983).
32. M. R. Nobari, Y.-J, Jan, and G. Tryggvason, Head-on collision of drops—A numerical investigation, *Phys. Fluids* **8**(1), 1 (1996).
33. A. Esmaeeli and G. Tryggvason, An inverse energy cascade in two-dimensional, low Reynolds number bubbly flows, *J. Fluid Mech*. **314**, 315 (1996).
34. W. Coirier and K. G. Powell, Solution-adaptive Cartesian cell approach for viscous and inviscid flows, *AIAA J*. **34**, 938 (1996).
35. H. Lamb, *Hydrodynamics* (Dover, New York, 1932), p. 471.
36. U. Bagge and P-I. Bränemark, White blood cell rheology, *Adv. Microcirc*. **7**, 1 (1977).
37. G. W. Schmid-Schönbein, K. P. Sung, H. Tözeren, R. Skalak, and S. Chien, Passive mechanical properties of human leukocytes, *Biophys. J*. **36**, 243 (1981).
38. C. Dong and R. Skalak, Leukocyte deformability: Finite element modeling of large viscoelastic deformation, *J. Theor. Biol*. **158**, 173 (1992).
39. D. N. Needham and R. M. Hochmuth, A sensitive measure of surface stress in the resting neutrophil, *Biophys. J*. **61**, 1664 (1992).
40. E. Evans and A. Yeung, Apparent viscosity and cortical tension of blood granulocytes determined by micropipet aspiration, *Biophys. J*. **56**, 151 (1989).
41. R. Tran-Son-Tay, H. P. Ting-Beall, D. V. Zhelev, and R. M. Hochmuth, Viscous behavior of leukocytes, in *Cell Mechanics and Cellular Engineering*, edited by V. C. Mow, F. Guilak, R. Tran-Son-Tay, and R. M. Hochmuth (Springer-Verlag, New York, 1994), p. 22.
42. D. Bray, J. Heath, and D. Moss, The membrane-associated 'cortex' of animal cells: Its structure and mechanical properties, *J. Cell Sci. Suppl*. **4**, 71 (1986).
43. D. Needman and R. M. Hochmuth, Rapid flow of passive neutrophils into a 4 μm pipet and measurement of cytoplasmic viscosity, *J. Biomech. Eng*. **112**, 269 (1990).
44. S. Chien, K.-L. P. Sung, R. Skalak, S. Usami, and A. Tözeren, Theoretical and experimental studies on viscoelastic properties of erythrocyte membrane, *Biophys. J*. **24**, 463 (1978).
45. D. E. Ingber, Cellular tensegrity: Defining new rules of biological design that govern the cytoskeleton, *J. Cell Science* **104**, 613 (1993).
46. T. J. Mueller, Application of numerical methods to physiological flows, in *Numerical Methods in Fluid Dynamics*, edited by H. J. Wirz and J. J. Smolderen (Hemisphere, Washington, DC, 1978).