JOURNAL OF COMPUTATIONAL PHYSICS 56, 111–134 (1984)

A Mathematical Model and Numerical Method for Studying Platelet Adhesion and Aggregation during Blood Clotting

AARON L. FOGELSON

Department of Mathematics, University of California,
Berkeley, California 94720 and
Applied Mathematics Group, Lawrence Berkeley Laboratory,
Berkeley, California 94720

Received October 25, 1983; revised February 7, 1984

The repair of small blood vessels and the pathological growth of internal blood clots involve the formation of platelet aggregates adhering to portions of the vessel wall. Our microscopic model represents blood by a suspension of discrete massless platelets in a viscous incompressible fluid. Platelets are initially noncohesive: however, if stimulated by an above-threshold concentration of the chemical ADP or by contact with the adhesive injured region of the vessel wall, they become cohesive and secrete more ADP into the fluid. Cohesion between platelets and adhesion of a platelet to the injured wall are modeled by creating elastic links. Repulsive forces prevent a platelet from coming too close to another platelet or to the wall. The forces affect the fluid motion in the neighborhood of an aggregate. The platelets and secreted ADP both move by fluid advection and diffusion. The equations of the model are studied numerically in two dimensions. The platelet forces are calculated implicitly by minimizing a nonlinear energy function. Our minimization scheme merges Gill and Murray's (*Math. Programming* 7 (1974), 311) modified Newton's method with elements of the Yale sparse matrix package. The stream-function formulation of the Stokes' equations for the fluid motion under the influence of platelet forces is solved using Bjorstad's biharmonic solver ("Numerical Solution of the Biharmonic Equation," Ph. D. Thesis, Stanford University, 1980). The ADP transport equation is solved with an alternating-direction implicit scheme. A linked-list data structure is introduced to keep track of changing platelet states and changing configurations of interplatelet links. Results of calculations with healthy platelets and with diseased platelets are presented.

1. INTRODUCTION

The early stages of the repair of a small blood vessel or of the pathological growth of internal blood clots involve the formation of aggregates of blood platelets adhering to portions of the vessel wall. We present here both a mathematical model and a numerical method for studying this aggregation process which, in particular, allow us to examine the interaction of the growing aggregates with the local fluid dynamics of the blood.

0021-9991/84 $3.00

Copyright © 1984 by Academic Press, Inc.
All rights of reproduction in any form reserved.

A brief discussion of the relevant biological phenomena will help motivate the model and numerical method. Platelets are cells suspended in the blood. They are present in enormous numbers (250,000/mm^3) yet small volume concentrations (less than 1%). Platelets are essentially neutrally buoyant with respect to the blood as a whole. Circulating platelets are normally nonadherent to one another or to the blood vessel wall. However, when an injury to a blood vessel occurs, adhesive platelet-activating tissue embedded in the vessel wall is exposed to the blood. Platelets quickly adhere to the wall near the injury and these platelets secrete an assortment of chemicals into the blood plasma. Other platelets attach to these platelets and to one another and these also release their stores of chemicals. As a result of the platelets cohering, a loose aggregate of platelets forms at the injury site. While there is not conclusive evidence that the released chemicals potentiate the aggregation *in vivo*, such a hypothesis is supported by *in vitro* studies in which a sufficiently high concentration of the chemical adenosine diphosphate (ADP), one of the released chemicals, can activate a platelet, inducing it to release its chemical stores and to become capable of adhering to other activated platelets [12, 21]. The events in the aggregation process occur in a moving fluid (the blood plasma) or at a fluid–solid interface (the vessel wall) so it is not surprising that a substantial amount of both clinical and experimental evidence indicates an important role for the blood's fluid dynamics in platelet aggregation. In particular, regions of relative flow stasis and regions in which a flow disturbance can be expected, such as near branchings or constrictions of a vessel, are indicated as preferential sites of pathological aggregate growth [14].

We have modeled platelet aggregation on a microscopic scale, representing blood by a viscous incompressible fluid in which discrete massless platelets are suspended. The blood flows through a vessel, a portion of which we designate as injured. Upon contact with the injury or when stimulated by a sufficiently high concentration of the chemical ADP, a platelet becomes activated and releases more ADP into the fluid. The platelets and ADP are both assumed to move by fluid advection and by diffusion. Activation of a platelet also enables the platelet to cohere with other activated platelets. We model this cohesion by creating an elastic link between cohering platelets. The link generates forces which resist motions tending to separate the platelets. These link forces affect the fluid motion in the neighborhood of the cohering platelets. Other interplatelet forces, which we describe later, also affect the local fluid motion. It is through the action of these forces on the fluid dynamics, rather than through a changing geometry, that an aggregate's presence is manifested. Our system is fully coupled in the sense that the fluid carries the ADP and platelets while interplatelet forces, potentiated by ADP-induced activation of the platelets, determine the local flow.

The numerical method we have constructed to study the model utilizes grid-based finite-difference schemes to solve the fluid dynamics and ADP-transport equations. The platelets are tracked independently of the grid using a simple Euler scheme to solve for the advective motion and a random walk technique to simulate the diffusive motion. Explicit evaluation of the interplatelet forces produces large instabilities in the resulting platelet motions. We therefore use an approximately implicit scheme to determine these forces. Solving for the forces involves minimizing a nonlinear energy function φ which depends on all of the platelet positions, a total of perhaps several hundred variables. The minimization is carried out using Gill and Murray's modified Newton's algorithm [5, 6] which we merged with elements of the Yale sparse matrix package [4]. The resulting algorithm can handle the sometimes indefinite Hessian matrix of φ while exploiting the fact that typically fewer than 10% of this matrix's elements are nonzero. An important element of our numerical method is a platelet data structure, based on the linked-list idea [13], which permits us to efficiently access and update information about the platelet states (e.g., activated or not) and about the configuration of interplatelet elastic links. A similar data structure might be useful in other particle methods, especially if there is a rapid turnover in the population of particles during a calculation or if particles can make transitions between several states.

2. Model: Assumptions and Equations

We model the aggregation of platelets in a very narrow blood vessel (diameter $\approx$ $50\,\mu$m) and base the model on interactions among four types of objects: discrete platelets, a fluid continuum representing the rest of the blood, a single activating chemical representing ADP, and the walls of the blood vessel.

The fluid is viscous, homogeneous, and incompressible. Based on the vessel radius and physiological blood-flow velocities in vessels of this size, the Reynolds number is less than 0.1. We assume that inertial effects are negligible and describe the flow by Stokes' equations

$$0 = -\nabla p + \Delta \mathbf{u} + \mathbf{f}; \qquad \nabla \cdot \mathbf{u} = 0, \tag{1}$$

where $\mathbf{u}$ is the velocity and p the pressure. The force density $\mathbf{f}(\mathbf{x}, t)$ arises from interactions among the platelets and will be discussed below.

We assume that the blood vessel walls are structurally rigid. Portions of the wall are designated as injured and therefore adhesive in a sense we will make precise below. The remainder of the wall is nonadhesive.

Taking advantage of the neutral buoyancy of actual platelets, we make the idealization that platelets are massless point particles that move primarily at the local fluid velocity. Two things distinguish them mechanically from other fluid particles. One of these is that, in addition to their advective motion, individual platelets move randomly relative to the flow. This is intended to simulate the diffusion-like motion of platelets which is observed experimentally in flowing blood and which is thought to result from local "stirring" induced by tumbling and colliding among the much larger red blood cells which occupy 45% of the blood volume [8]. This additional platelet motion is important, for it provides a mechanism for platelet transport to the vessel wall. We let $\mathbf{x}_k(t)$ represent the location of the kth platelet at time t and decribe its motion with the "particulate advection–diffusion" equation

$$d\mathbf{x}_k = \mathbf{u}(\mathbf{x}_k(t), t)\, dt + d\mathbf{B}_k \tag{2}$$

where $\mathbf{B}_k(t)$ is a Brownian motion path with diffusion coefficient equal to the platelet diffusion coefficient σ_p. This equation holds for $k = 1, 2, \ldots, N(t)$, where $N(t)$ is the number of platelets in the vessel segment at time t. The second way platelets differ from fluid particles is that the force density f in the Stokes' equations can be nonzero only in the neighborhood of a platelet (see below).

Each platelet is assumed to exist in one of two states: nonactivated or activated. A nonactivated platelet contains a store of ADP and is noncohesive. An activated platelet is devoid of ADP and is "sticky," i.e., it can cohere with other activated platelets. We assume that once a platelet is activated, it remains so.

Activation of a platelet occurs if the platelet encounters a concentration of ADP that is above a prescribed threshold value c_T or if the platelet touches the injured region of the blood vessel wall IW. This may be formalized as follows. Let

$$t'_k = \min\{t : c(\mathbf{x}_k(t), t) \geqslant c_T\} \tag{3}$$

and

$$t''_k = \min\left\{t : \operatorname{dist}(\mathbf{x}_k(t), IW) \leqslant \frac{d}{2}\right\}. \tag{4}$$

Here d is a distance which we think of as a platelet diameter. In the definition of t'_k and t''_k, we make the convention that the minimum of the empty set is ∞. Then platelet k is activated at the time t_k given by

$$t_k = \min\{t'_k, t''_k\}. \tag{5}$$

At this time the platelet immediately becomes sticky and begins to release its ADP into the fluid. The time course of ADP release is assumed to be described by a smooth function $\delta_\tau(t)$ of integral 1 and support in a time interval of length order τ. We note that the smooth release of ADP precludes the possibility of a platelet activating itself. Note also that we have introduced a distance d for the platelet diameter. Although a platelet moves as a point particle, it is useful to give it an effective volume when describing its interactions with the wall or with other platelets as well as when describing the release reaction.

Each time a platelet is activated, ADP is released into the fluid. The mathematical expression for the source created by the release of ADP is

$$s(\mathbf{x}, t) = \sum_k A\, \delta_d(\mathbf{x} - \mathbf{x}_k(t))\, \delta_\tau(t - t_k). \tag{6}$$

Here A is the total amount of ADP released by a single platelet, $\delta_d(\mathbf{r})$ is a spherically symmetric function with integral 1 and support in a sphere of diameter d, and t_k and

$\delta_\tau(t)$ are as described above. Note that, in (6), the sum at time t is, in effect, a sum over those platelets activated during a time interval of length τ prior to time t.

Four types of forces, two attractive forces and two repulsive forces, can act on a model platelet. The attractive forces arise from adhesion of a platelet to the injured wall or from the cohesion of two activated, hence sticky, platelets. If two activated platelets come within a platelet diameter (d) of one another. a mechanical link is created between them. The effect of this link is analogous to that of a rubber band looped around the pair of platelets which generates a force on each of these platelets to resist motions increasing the distance between them. The force on the kth platelet due to such a link connecting it to the mth platelet is assumed to have the form

$$S_L(\|\mathbf{x}_k - \mathbf{x}_m\| - d)\, H(\|\mathbf{x}_k - \mathbf{x}_m\| - d)\frac{\mathbf{x}_m - \mathbf{x}_k}{\|\mathbf{x}_m - \mathbf{x}_k\|}. \tag{7}$$

Here S_L is the stiffness of the interplatelet link (a constant for now), $H(\cdot)$ is the Heaviside function, and $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$. A similar elastic link is created between any platelet (activated or not) that touches the injured wall and its point of first contact. We allow at most one link between a given platelet and the wall and at most one link between a given pair of platelets. A platelet may, however, form links with many distinct platelets. This permits the growth of sizable platelet aggregates. The two repulsive forces are intended to simulate the effect of the platelet's nonzero volume. Any platelet coming less than a platelet radius from the wall experiences a repulsive force acting normal to the wall. Any two platelets closer together than a platelet diameter are mutually repellent. Both of these forces are assumed to have forms analogous to that in (7). Let $\mathbf{f}_k$ be the resultant of all the forces acting on the kth platelet. Note that $\mathbf{f}_k$ is a function only of the positions of the platelets $\mathbf{x}_1,..., \mathbf{x}_N$. The force density $\mathbf{f}$ in the Stokes' equations (1) is generated by spreading $\mathbf{f}_k$ to a region around $\mathbf{x}_k$ that is the size of a real platelet.

$$\mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t; \mathbf{x}_1,..., \mathbf{x}_N) = \sum_{k=1}^{N(t)} \mathbf{f}_k(\mathbf{x}_1(t),..., \mathbf{x}_N(t))\, \delta_d(\mathbf{x} - \mathbf{x}_k(t)). \tag{8}$$

The great advantage of treating platelets as point particles which can generate local forces is that the geometry of the domain in which the equations of the model are solved remains constant in time. Even as platelet aggregates develop, we regard the entire vessel as filled with fluid and solve the equations in the full domain. The presence of the aggregates is manifested solely through the forces generated by the platelets making up the aggregates. This technique was patterned closely after Peskin's representation of heart wall muscle by a set of discrete point forces [16].

The final assumption we make concerns the transport of ADP, which we take to be described by the advection–diffusion equation,

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \sigma \Delta c + s(\mathbf{x}, t), \tag{9}$$

where $c(\mathbf{x}, t)$ is the concentration of ADP, σ is the ADP diffusion coefficient, and $s(\mathbf{x}, t)$ is the source term defined in (6). We have retained the time derivative here, while discarding it in the fluid dynamics equations (1), because $\mu/\rho \gg \sigma$, where μ is the blood viscosity and ρ is the blood mass density.

The state of our model system at any time t is given by the platelet positions $\mathbf{x}_1, ..., \mathbf{x}_N$, the ADP concentration $c(\mathbf{x}, t)$, the sets of nonactivated and activated platelets (and a list of activation times for the activated platelets), the set of pairs of linked platelets, and the set of platelets linked to the wall. Then Eq. (6) gives $s(\mathbf{x}, t)$, and expressions like Eqs. (7) and (8) give $\mathbf{f}(\mathbf{x}, t)$. We obtain $\mathbf{u}(\mathbf{x}, t)$ by solving Eq. (1) with suitable boundary conditions. The evolution of the system is described by Eqs. (2) and (9) along with boundary conditions on $c(\mathbf{x}, t)$ and by the conditions for activating and linking platelets.

The boundary conditions hat we impose depend on the experimental or physiological situation we wish to stimulate. For the numerical experiments that we describe in this paper, we impose the no-slip condition on $\mathbf{u}$ at the vessel walls and stipulate that both the incoming and outgoing flows be Poiseulle. The no-slip condition implies that there is no advective flux of ADP across the vessel walls and we assume that there is also no diffusive flux, so $\partial c/\partial n = 0$ along the vessel walls. One expects that over time most of the ADP that diffuses upstream will be carried back into the domain by the flow. We think therefore that it is reasonable to impose the more stringent condition of zero net flux pointwise along the upstream boundary, i.e., $uc - \sigma(\partial c/\partial x) = 0$. The downstream boundary condition for c is chosen based on the premise that, because the nondimensionalized ADP diffusion coefficient is much smaller than 1, most of the ADP flux across this boundary would be advective flux. We assume that $(\partial c/\partial t) + u(\partial c/\partial x) = 0$ for a point (x, y, z) on the downstream boundary where $u = u(y, z)$ is the prescribed Poiseulle flow. This formulation for the downstream boundary condition avoids the occurrence there of a concentration boundary layer.

3. Numerical Method: Introduction

We will now describe the numerical scheme used to study the model. We restrict the model to two spatial dimensions to economize on computer time and memory. We note, however, that with the exception of the solution of the Stokes' equations, our numerical scheme is identical to that which we would use for three-dimensional calculations. Our domain is the rectangle $\Omega = \{(x, y) : 0 \leqslant x \leqslant x_{\max}, 0 \leqslant y \leqslant y_{\max}\}$. The sides $y = 0$ and $y = y_{\max}$ represent the vessel walls; $x = 0$ and $x = x_{\max}$ are, respectively, the inlet and outlet of the vessel segment. We place a uniform square mesh of size h over Ω and divide time into timesteps of size Δt. The Eulerian variables $\mathbf{u}, \mathbf{f}, c$, and s are defined at points of the mesh and we use the usual notation $\mathbf{u}_{ij}^n = \mathbf{u}(ih, jh, n\Delta t)$, etc. The platelets are tracked independently of the mesh and we use the notation $\mathbf{x}_k^n = \mathbf{x}_k(n\Delta t)$ to denote the position of the kth platelet at time $n\Delta t$.

Since the platelet positions generally do not coincide with mesh points, values of $\mathbf{u}$

and c must be defined at the location of each platelet for use in Eq. (2) and to check the activation condition (3). This interpolation is accomplished using Peskin's [16] discrete approximation $D_{ij}(\mathbf{x})$ to the two-dimensional delta function. The function $D_{ij}(\mathbf{x})$ has support consisting of those mesh points within a square of side $4h$ centered on $\mathbf{x}$; $D_{ij}(\mathbf{x})$ is order h^{-2} at $\mathbf{x}$; and $\sum_{ij} h^2 D_{ij}(\mathbf{x}) = 1$ for any $\mathbf{x}$. We define

$$\mathbf{u}(\mathbf{x}_k^n) = \sum_{ij} \mathbf{u}_{ij}^{n+1} D_{ij}(\mathbf{x}_k^n) h^2, \tag{10}$$

which is an approximation to the identity

$$\mathbf{u}(\mathbf{x}_k(t), t) = \int_\Omega \mathbf{u}(\mathbf{x}, t)\, \delta(\mathbf{x} - \mathbf{x}_k(t))\, d\mathbf{x}.$$

An equation similar to Eq. (10) is used to define c at each nonactivated platelet's position. The function $D_{ij}(\mathbf{x})$ is also a natural discretization for our function $\delta_d(\mathbf{x})$ which appears in the definitions of $\mathbf{f}(\mathbf{x}, t)$ and $s(\mathbf{x}, t)$. We therefore use

$$\mathbf{f}_{ij} = \sum_{k=1}^{N} \mathbf{f}_k(\mathbf{x}_1, \dots, \mathbf{x}_N)\, D_{ij}(\mathbf{x}_k) \tag{11}$$

$$s_{ij}^n = \sum_{k=1}^{N} A\, D_{ij}(\mathbf{x}_k^n)\, \delta_k^n \tag{12}$$

to define values of $\mathbf{f}$ and s at points of the mesh. The mesh size currently in use is such that the area of the support of $D_{ij}(\mathbf{x})$ agrees well with the area of a platelet. In defining s_{ij}^n, we assume that all of a platelet's ADP is released during the timestep in which the platelet is activated. So,

$$\begin{aligned} \delta_k^n = 1 \quad & \text{if platelet } k \text{ is activated in the } n\text{th timestep,} \\ 0 \quad & \text{otherwise.} \end{aligned}$$

The superscripts denoting time were omitted in Eq. (11) for reasons that will become clear later. We remark that the relationship between the support of D_{ij} and the size of a platelet can be maintained as $h \to 0$ by including more mesh points in the support of D_{ij}. Similarly, if Δt were diminished, the release of ADP by a platelet would be made to occur over a correspondingly larger number of timesteps.

4. Overview of Numerical Calculations

We will now give an overview of the operations performed to advance the model system from time $n\Delta t$ to time $(n + 1)\Delta t$. Details about the methods used appear in the following sections.

New platelets are introduced randomly at the upstream boundary at a prescribed average rate and with their y coordinates chosen from a prescribed distribution. Next, values $\mathbf{f}_k^*$ for the forces acting on the platelets are calculated using an approximately implicit scheme to avoid numerical instabilities in the platelet motion. Using these $\mathbf{f}_k^*$'s in the right-hand side of Eq. (11), the force density $\mathbf{f}_{ij}^{n+1}$ is calculated at points of the mesh and, using these values, the new velocity field $\mathbf{u}_{ij}^{n+1}$ is determined. After interpolating the velocities to the platelet positions using Eq. (10), the platelets are moved to new positions

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \mathbf{u}_k^{n+1}(\mathbf{x}_k^n)\,\Delta t + (\sigma_p \Delta t)^{1/2}\,\mathbf{R}, \tag{13}$$

where $\mathbf{R}$ is a Gaussian-distributed random variable with mean 0 and variance 2. Platelets which reach the downstream boundary are removed from the calculations.

After moving the platelets, tests are performed to determine which changes to make in the states of the platelets and in the connections between platelets. If condition (3) is satisfied at the position $\mathbf{x}_k^{n+1}$ of nonactivated platelet k, then this platelet is activated and a suitable contribution is made to the source term defined in Eq. (6). Next, each platelet is examined to see whether it should be linked to the injured wall. If a nonactivated platelet becomes linked to the wall, it is also activated and contributes to the source term (6). In the final type of test, pairs of activated platelets are examined to see whether the platelets should be linked together; i.e., whether these platelets are sufficiently close together but are not already linked to one another. We do a preliminary spatial sorting of the platelets to reduce the number of pairwise calculations.

After the activating and linking are completed, the advection–diffusion equation, including new sources, is solved to determine new values of the ADP concentration. This completes the calculations for this timestep.

5. Fluid Dynamics

For the moment, assume that the force density $\mathbf{f}$ is known. In two dimensions, the incompressibility of the flow implies that a stream function ψ exists such that, if $\mathbf{u} = (u, v)$, then

$$u = \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x}. \tag{14}$$

Taking the curl on both sides of Eq. (1), which eliminates the pressure term, and making use of Eq. (14), we obtain the scalar equation

$$\Delta\Delta\psi = \hat{\mathbf{z}} \cdot \operatorname{curl} \mathbf{f} \tag{15}$$

where $\hat{\mathbf{z}}$ is a unit vector normal to the plane of the flow. The no-slip condition at the vessel walls and the prescribed parabolic flow profiles at the vessel inlet and outlet easily yield boundary conditions for ψ and its normal derivative on all of the boundaries. To solve Eq. (15) with these boundary conditions, we use a finite-difference-based biharmonic solver for rectangular domains developed by Bjorstad [1]. Then u and v are obtained at mesh points by centered-difference approximations to Eq. (14).

6. Interplatelet Forces

The forces $\mathbf{f}_k$ generated by platelet interactions are calculated implicitly to avoid instabilities in the platelet motions induced by these forces. An approximately implicit scheme, similar to one introduced by Peskin [17], proves sufficient to damp out the instabilities. We define points $\mathbf{x}_1^*, ..., \mathbf{x}_N^*$ by the system of equations

$$\mathbf{x}_k^* = \mathbf{x}_k^n + \lambda\, \Delta t\, \mathbf{f}_k(\mathbf{x}_1^*, ..., \mathbf{x}_N^*) \tag{16}$$

for $k = 1, ..., N$. The parameter λ in Eq. (16) is the magnitude of the velocity induced by a unit force and was estimated numerically. The approximation built into Eq. (16) is that the fluid velocity at $\mathbf{x}_k$ is proportional to the force $\mathbf{f}_k$ and is independent of the forces $\mathbf{f}_m$ for $m \neq k$. Actually, the velocity at each point is a functional of all the forces, with $\mathbf{f}_k$ typically being the largest contributor to the velocity at $\mathbf{x}_k$. Insofar as the above approximation is valid, $\mathbf{x}_1^*, ..., \mathbf{x}_N^*$ is the platelet configuration at the end of the timestep. Equation (16) is **implicit** because the forces are calculated from the (unknown) final configuration rather than from the given configuration $\mathbf{x}_1^n, ..., \mathbf{x}_N^n$. Using the solution of Eq. (16), we define

$$\mathbf{f}_k^* = \mathbf{f}_k(\mathbf{x}_1^*, ..., \mathbf{x}_N^*). \tag{17}$$

We use these new forces in Eq. (11) to define the force density for the Stokes' equations. The new velocity field due to the influence of all the interplatelet forces is then calculated and the platelets are moved according to Eq. (13). We emphasize that the points $\mathbf{x}_k^*$ are used only to calculate $\mathbf{f}_k^*$.

Before describing the means of solving Eq. (16), we introduce the notation $\mathbf{X} = (\mathbf{x}_1^*, ..., \mathbf{x}_N^*)$; $\mathbf{X}^0 = (\mathbf{x}_1^n, ..., \mathbf{x}_N^n)$; and $\mathbf{F} = (\mathbf{f}_1, ..., \mathbf{f}_N)$ and note that the system in Eq. (16) can then be expressed

$$0 = \mathbf{X} - \mathbf{X}^0 - \lambda\, \Delta t\, \mathbf{F}(\mathbf{X}). \tag{18}$$

Although the Jacobian matrix of the (nonlinear) function $\mathbf{X} - \lambda\, \Delta t\, \mathbf{F}(\mathbf{X})$ is symmetric, it is often indefinite in our calculations, so the multidimensional Newton–Raphson method is inappropriate for solving Eq. (18). Instead, following Peskin [17], a differentiable "energy function" $E(\mathbf{X})$ is introduced such that the following conditions are satisfied:

(i) $E(\mathbf{X}) \geqslant 0$

(ii) $E(\mathbf{X}) \to 0$ as $\|\mathbf{X}\| \to \infty$

(iii) $F(\mathbf{X}) = -\operatorname{grad} E(\mathbf{X})$.

A minimum point of the function $\varphi(\mathbf{X}) = \frac{1}{2} \|\mathbf{X} - \mathbf{X}^0\|^2 + \lambda \Delta t\, E(\mathbf{X})$ is then a solution of Eq. (16). We use Gill and Murray's modified Newton's method (MNM) [5, 6] to seek a minimum point of φ. This method can handle the symmetric but indefinite Hessian matrices that we encounter (the Hessian of φ is identical to the Jacobian of $\mathbf{X} - \lambda \Delta t\, \mathbf{F}(\mathbf{X})$ mentioned before).

In brief, Gill and Murray's method works as follows. Let $\mathbf{g}$ be the gradient of φ and G be the Hessian matrix of φ. A sequence $\{\mathbf{X}^{(k)}\}$ of approximations to a minimum point of φ is defined iteratively by the equations

$$(G^{(k)} + \Lambda^{(k)})\, \mathbf{p}^{(k)} = -\mathbf{g}^{(k)} \tag{19}$$

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}. \tag{20}$$

In Eq. (19), $\Lambda^{(k)}$ is a nonnegative diagonal matrix to be discussed below. The vector $\mathbf{p}^{(k)}$ is called a search direction and $\alpha^{(k)}$ is a positive scalar chosen to achieve a "sufficient decrease" in $\varphi^{(k+1)}$ relative to $\varphi^{(k)}$. The process of choosing $\alpha^{(k)}$ is called linesearch. We employ a linesearch algorithm similar to that described in [7].

The solution of Eqs. (19) is facilitated by the symmetric Gaussian decomposition

$$(G^{(k)} + \Lambda^{(k)}) = U^{(k)T} D^{(k)} U^{(k)}, \tag{21}$$

where $U^{(k)}$ is a unit upper-triangular matrix and $D^{(k)}$ is a diagonal matrix. $\Lambda^{(k)}$ is constructed during the row-by-row factorization in such a way that:

(i) $G^{(k)} + \Lambda^{(k)}$ is positive definite;

(ii) diagonal elements of $D^{(k)}$ are bounded away from zero by a positive constant δ;

(iii) the inequality $|d_{jj} u_{jk}^2| \leqslant \beta^2$ holds for $k > j$, where β^2 is a preselected constant.

Condition (i) guarantees that $\mathbf{p}^{(k)}$ gives a descent direction for φ as can be seen by premultiplying Eq. (19) by $\mathbf{p}^{(k)T}$. Conditions (ii) and (iii) assure the numerical stability of the factorization and the subsequent forward and back substitution used in solving Eq. (19).

Provided β^2 has been suitably chosen, $\Lambda^{(k)}$ is automatically set to zero if $G^{(k)}$ is itself sufficiently positive definite in a sense defined by Gill and Murray [5]. The modified Newton's method therefore reduces to Newton's method, where $G^{(k)}$ is positive definite (e.g., in the neighborhood of a minimum point of φ), and thus it has the locally quadratic rate of convergence exhibited by the Newton's method.

In the absence of sparsity, the factorization (21) as described by Gill and Murray requires order $n^3/6$ operations and order n^2 storage locations where n is the order of the Hessian matrix $G^{(k)}$. Factorization of several Hessian matrices is usually necessary in each time step. Recall that, for our work, n is twice the number of platelets present in the domain; it has been as large as 250 in calculations to date and we envision doing experiments in which it might grow to order 1000. Fortunately the

Hessian matrices that we encounter are sparse. To see why this is so, we note that $G^{(k)}$ has a natural 2×2 block structure in which each block corresponds to the spatial coordinates of a pair of platelets. The elements of an off-diagonal block are nonzero only if the pair of platelets in question is linked directly together or if the two platelets are close enough to repel each other. We find that, in practice, fewer than 10% of the entries of G are nonzero. For each platelet, the number of direct interactions with other platelets is limited by the platelet's effective nonzero volume. Therefore, the number of nonzeros in $G^{(k)}$ should increase only linearly with n as we go to larger calculations.

We exploited this sparseness, to reduce substantially the amount of computational work and storage necessary for the factorization, by merging the MNM with the sparse symmetric Gaussian elimination algorithm of the Yale sparse matrix package [4]. The latter algorithm was designed to solve efficiently a linear system involving a sparse, positive definite, symmetric (but otherwise unstructured) coefficient matrix. By making the changes in this algorithm shown in Appendix A, we extended its use to performing the MNM's modified factorization for our symmetric but indefinite Hessian matrices.

The Yale sparse matrix algorithm exploits the sparseness of a matrix G in three ways:

(i) A compact row-by-row storage scheme for G (and its factor U), developed by Gustavson [11], reduces the storage requirements for G to little more than twice the number of nonzero elements. This row-wise storage is compatable with the row-by-row factorization necessary for the MNM.

(ii) Only nonzero elements of G and U are involved in the arithmetic operations of the factorization. The data structure associated with the algorithm, of which the row-by-row storage scheme is a part, permits this restriction to nonzero elements with little testing of whether a given element is nonzero.

(iii) A symmetric reordering of the rows and columns of G is done prior to the factorization to reduce the incidence of "fill-in"; i.e., of indices (i, j) such that $g_{ij} = 0$ but $u_{ij} \neq 0$. This reduces both the work and storage required for the factorization.

Details of the factorization are given in [2, 3] and a graph-theoretic treatment of the minimum-degree algorithm used to reorder G may be found in [18]. A peculiar consequence of combining the reordering algorithm of the Yale code with the modified factorization of the MNM is that the matrix $A^{(k)}$ in Eq. (19) is dependent on the ordering of G. This is so because when a row of $D^{(k)}$ and $U^{(k)}$ (hence of $A^{(k)}$) is modified in the row-by-row factorization of the MNM, all subsequent rows can be affected.

7. Chemical Transport

The advection–diffusion equation describing the transport of ADP is approximated by a stable, implicit alternating-direction scheme. We introduce the usual difference operators D^+, D^-, and D^0 defined by

$$(D^+f)_j=\frac{(f_{j+1}-f_j)}{h}$$

$$(D^-f)_j=\frac{(f_j-f_{j-1})}{h}$$

$$(D^0f)_j=\frac{(f_{j+1}-f_{j-1})}{2h}.$$

We use a subscript x or y to indicate with respect to which variable the differencing is done. We let c^n be the mesh function with values c^n_{ij} and we similarly define c^*, c^{n+1}, s^{n+1}, u^{n+1}, and v^{n+1}. Then the scheme we use can be expressed:

$$\left(I+\frac{\Delta t}{2}(u^{n+1}D^0_x+D^0_xu^{n+1})-\sigma\,\Delta t\,D^+_x D^-_x\right)c^*=c^n+\Delta t\,s^{n+1} \tag{22}$$

$$\left(I+\frac{\Delta t}{2}(v^{n+1}D^0_y+D^0_yv^{n+1})-\sigma\,\Delta t\,D^+_y D^-_y\right)c^{n+1}=c^*. \tag{23}$$

In these equations, (u^{n+1}, v^{n+1}) are known from the solution of the Stokes' equations, c^n is the known ADP concentration field at the end of the last timestep, and s^{n+1} is the known source term given by Eq. (12) due to ADP released during the current timestep. Equation (22) (along with approximations to the boundary conditions) is solved first yielding values of c^* at all mesh points; then Eq. (23) is solved for the new ADP concentrations c^{n+1}_{ij}.

All of the boundary conditions for c discussed in Section 2 can be put into the form

$$d_1\frac{\partial c}{\partial n}+d_2c=d_3, \tag{24}$$

where $d_1=0$ or 1, d_2 and d_3 are known functions of x or y, and $\partial c/\partial n$ is the derivative of c normal to the boundary. We approximate these boundary conditions at, say, $x=0$, by choosing $c^*_{0,j}$ so that the parabola through $(0, c^*_{0,j})$, (h, c^*_{1j}), and $(2h, c^*_{2j})$ satisfies Eq. (24) at $x=0$. This preserves the second-order spatial accuracy of the difference equations.

If Eq. (22) is written componentwise, it is easy to see that it decouples into equations which allow $c^*_{1j},..., c^*_{Ij}$ to be determined separately for each j ($j=1,..., J$). These equations for $c^*_{1j},..., c^*_{Ij}$ combined with the approximations to Eq. (24) yield an almost tridiagonal system of the form shown in Fig. 1. In that figure, $\mathbf{r}_j$ is a known

$$\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 0 & \cdots & 0 \\ \gamma_1 & & & & & \vdots \\ 0 & & A & & & \\ \vdots & & & & & \gamma_2 \\ 0 & 0 & \cdots & 0 & \beta_2 \quad \beta_1 & \beta_0 \end{pmatrix} \begin{pmatrix} c^*_{0,j} \\ c^*_{1j} \\ \vdots \\ c^*_{Ij} \\ c^*_{I+1,j} \end{pmatrix} = \begin{pmatrix} r_{0j} \\ r_{1j} \\ \vdots \\ r_{Ij} \\ r_{I+1,j} \end{pmatrix}$$

FIG. 1. The form of the linear systems for the finite-difference approximation of the ADP advection–diffusion equation.

vector, A is a tridiagonal matrix, and the α's, β's, and γ's may be nonzero. The solution of this system can be accomplished by solving three tridiagonal systems all with the same coefficient matrix A and a 2×2 system for $c^*_{0,j}$ and $c^*_{I+1,j}$. After obtaining c^*_{ij} for $i = 0, 1,..., I + 1$ and $j = 0, 1,..., J + 1$, Eq. (23) yields similar easily solved systems for c^{n+1}_{ij}.

For the scheme given by Eqs. (22) and (23), it is important that the inequality

$$\frac{\sigma}{Uh} > \frac{1}{2}, \tag{25}$$

in which U is a characteristic flow speed, be satisfied (see, e.g., [15]). In regions where this inequality is violated, the solution to the difference scheme exhibits nonphysical spatial oscillations with wavelength of order h. Having chosen a practical value of h, the inequality constrained us to using an artificially large value of σ for our calculations. We are currently exploring the possibility of using particle methods to overcome this difficulty.

8. LINKED-LIST DATA STRUCTURE

Our need to keep track of continually-turning-over populations of platelets, their changing positions and states, and the changing configuration of interplatelet elastic links poses for us an information storage problem different from those usually encountered in, say, solving differential equations on a grid. We need to be able to store and access this information readily and to perform the specific tests, discussed in Section 4, that are necessary to update it, without wasting a lot of storage space or computational time in searching. We will describe elements of a data structure based on the idea of "linked-lists" [13] that can efficiently handle this problem. It is worth remarking that similar data structures might be profitably used in vortex methods and other particle methods involving a large turnover of particles.

We begin by describing a FORTRAN implementation of the simplest form of linked list. Suppose we have a set of items numbered $1, 2,..., N$ partitioned into disjoint categories labeled $A, B,...$. A pointer PTA, PTB,..., gives the first item in each category. An array LIST of dimension N stores the rest of the information as follows: given item I, LIST(I) is the next item in the same category. The last item in each category is indicated by a 0.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
LIST	13	10	5	9	11	2	6	0	14	0	8	15	12	16	0	0

FIG. 2. An example of a linked-list whose elements are partitioned into four categories with pointers PNA = 3, PA = 7, PL = 4, and PAN = 1.

To illustrate the use and advantages of a linked-list data structure, we consider how we monitor the platelets currently in our domain. First, suppose we have a means of readily transferring an item from one category to another within a linked list. We will show how this can be done later. Each platelet currently in the domain has been assigned an identifying number or tag. These tags are stored in a linked list which is partitioned into four categories holding, respectively, the tags of platelets which are: nonactivated; activated but not physically linked to another platelet or the wall; activated and physically linked; and finally a category of available tag numbers, i.e., numbers not currently assigned to a platelet in the domain. The following example may help make this clear. Let PNA, PA, PL, and PAN be the pointers for the categories "nonactivated platelets," "activated platelets," "linked platelets," and "available numbers," respectively. Suppose that the pointers and LIST are currently as depicted in Fig. 2. Then the current contents of the four categories are:

NA	3, 5, 11, 8
A	7, 6, 2, 10
L	4, 9, 14, 16
AN	1, 13, 12, 15.

When a new platelet is introduced into the domain at the upstream boundary, it is assigned the first number in the category of available numbers, and this number is transferred to the nonactivated-platelet category. (So, in our example, item 1 would be transferred to the nonactivated-platelet category.) If, subsequently, the platelet becomes activated or physically linked, the number is transferred to the appropriate category. If the platelet exits the domain, its number is transferred to the category of available numbers for later use tagging a new platelet. We see that the linked-list structure allows us to **reuse** storage space. This is important because, while several thousand platelets may pass through the domain during the course of a numerical experiment, there are typically fewer than two hundred platelets in the domain at any given time. The actual savings in storage space is several times the amount suggested because for each platelet we need several pieces of information, e.g., x and y coordinates, identification numbers of the platelets to which it is physically linked, etc. Not only do we save storage space, but **the amount of searching is drastically reduced** since we do not have to search through several thousand storage locations to locate the platelets currently in the domain. Further reductions in searching occur because the platelets are stored by category. For example, recall that we perform a test for each nonactivated platelet to see whether it should be activated. This or

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
LIST			7	10	4		5	0		11	13		8			

FIG. 3. A single category stored in a linked-list. PTA = 3 is the pointer for this category.

another operation performed on all items in a category, say category A, can be done with no searching by accessing the category as done in the following loop:

```
   LP = PTA
 5 IF(LP.EQ.0) GO TO 10
   —OPERATION ON ITEM LP—
   LP = LIST(LP)
   GO TO 5
10 ....
```

The process of transferring an item I from category A to category B can be done in two steps, first removing item I from category A and then inserting it in category B. Adding item I to the beginning of category B is achieved easily with the instructions

```
LIST(I) = PTB
PTB = I
```

Deleting item I from category A is more difficult in that it requires more information. To make the chain of items in category A "skip over" item I, we need to know the item in this category just before item I, and to replace the I stored there by LIST(I). This may be clearer from an example. Suppose category A is as depicted in Fig. 3, with PTA = 3, so that the chain of items in category A is 3, 7, 5, 4, 10, 11, 13, 8. To remove item 4, we would replace 4 in LIST(5) by LIST(4) = 10. Then the chain would be 3, 7, 5, 10, 11, 13, 8. The problem, in general, is to determine which item comes just before item I. If we are sweeping through the entire category and perhaps transferring some items, then it is easy to store temporarily the number of the preceding item. If, however, we want the ability to transfer a **given** item without sweeping through the category, we need to use a more elaborate form of linked-list structure called a "doubly-linked list" which we will now describe.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
LIST			7	10	4		5	0		11	13		8			
LISTR			0	5	7		3	13		4	10		11			

FIG. 4. Storage of a single category in a doubly-linked list. The items are stored in the order 3, 7, 5, 4, 10, 11, 13, 8 in LIST and in the reverse order 8, 13, 11, 10, 4, 5, 7, 3 in LISTR.

Consider in addition to the array LIST described above, an array LISTR (R for reverse) containing the same items as LIST, partitioned into the same categories, but with the order within each category reversed relative to that in LIST. Continuing to use the above example, we would have the situation shown in Fig. 4 with PTA = 3. The item that comes just before item 4 in LIST is the item that comes just **after** item 4 in LISTR, i.e., LISTR(4) = 5. So, to remove item 4 from category A, we can use the following instructions, which also remove item 4 from category A in LISTR:

$$\begin{aligned}&\mathrm{LIST(LISTR(4)) = LIST(4)}\\&\mathrm{LISTR(LIST(4)) = LISTR(4)}\end{aligned}$$

To complete the transfer of item 4 to the beginning of category B, we use the following:

$$\begin{aligned}&\mathrm{LIST(4) = PTB}\\&\mathrm{LISTR(PTB) = 4}\\&\mathrm{PTB = 4}\end{aligned}$$

We used a doubly-linked list to store the categories of platelet states described above. We used a similar doubly-linked list, indexed by identifying tags assigned to the [redacted] collection of points (e.g., platelet positions) in a region. A category consists of the subset of points in a spatial cell and there is an array of pointers, one pointer corresponding to each cell.

9. Results

The results of two numerical experiments are depicted in Figs. 5 and 6. In the experiment shown in Fig. 5, platelets carried an amount of ADP (the parameter A in Eq. (6)) in the physiological range for healthy platelets. For the experiment shown in Fig. 6, the amount of ADP each platelet could release was substantially reduced. Reduced ability to secrete ADP is a characteristic of some platelet diseases including "storage-pool disease" [22, 23].

The domain for the experiments was a rectangle representing a 100 μm length of a 50 μm diameter blood vessel. The interval from 25 μm to 75 μm along the vessel on both top and bottom walls was designated as injured. Parameter values in the physiological range were used for the incoming flow velocity, the platelet number density, and the platelet diameter, while the values used for the ADP and platelet diffusion coefficients were larger than those found in physiological situations. The former was chosen 20 times the biological value in order to satisfy inequality (25) with a practical choice of the mesh size h.

We used a uniform square grid with 81 points along the length of the vessel and 41 points across it. The timestep used corresponds to 0.0381 ms and each experiment was allowed to run for 7800 timesteps. Each took approximately 170 CPU minutes and used $155{,}000_8$ words of core memory on a CDC 7600.

The pictures in Fig. 5 show the instantaneous state of the model system at increments of 300 timesteps. The generally horizontal solid lines are flow streamlines separated by a constant increment in ψ. The flow velocity is therefore higher where the streamlines are closer together. The dotted line is the curve along which the ADP concentration equals the activation-threshold concentration c_T. The above-threshold region is generally to the right of this curve. Asterisks (*) represent nonactivated platelets and A's represent activated ones. A line segment connecting a platelet to another platelet or to the injured wall represents an elastic link.

The initial state for the experiment was a uniform distribution of platelets in the domain and an identically zero ADP concentration. By frame (a), five platelets had adhered to the top and bottom injuries and induced a chain reaction of activation that moved the threshold-concentration boundary upstream of the injury. The fluid motion was noticably disturbed in a neighborhood of each of these five platelets. Over the next seven frames, a monolayer of adherent platelets formed over much of the injury, and small aggregates began to form towards the upstream ends of both injuries. These aggregates grew substantially during the remainder of the experiment and directed the blood flow away from the downstream portions of the injuries, thus blocking continued growth of the downstream aggregates.

Figure 6 shows the state of adhesion and aggregation in the second experiment after 7800 timesteps, the same time as that of the last picture in Fig. 5. This experiment differed from the first only in that the amount of ADP that each platelet could release was reduced by a factor of 6. The released ADP was not sufficient to maintain an above-threshold ADP concentration in the vicinity of either injury. We see that, as a result, aggregation was severely reduced; indeed it was almost entirely limited to direct adhesion of the platelets to the injury, which, we recall, does not depend on prior activation by ADP.

10. Conclusions

We have presented a mathematical model of platelet aggregation, based, in part, on a mechanism of platelet activation by a platelet-released agent (ADP). The model was able to generate extensive aggregate development showing that platelet aggregation may indeed occur via such a mechanism. It also was able to mimic the greatly reduced aggregation levels which characterize the blood of patients suffering from platelet "storage-pool disease."

A major feature of our numerical method is its versatility. We are currently able to simulate two of the best experimental procedures used to study the influence of fluid dynamics on platelet aggregation [9, 10, 20, 21]. We expect to be able to do further numerical experiments which parallel other physical experiments and, perhaps more

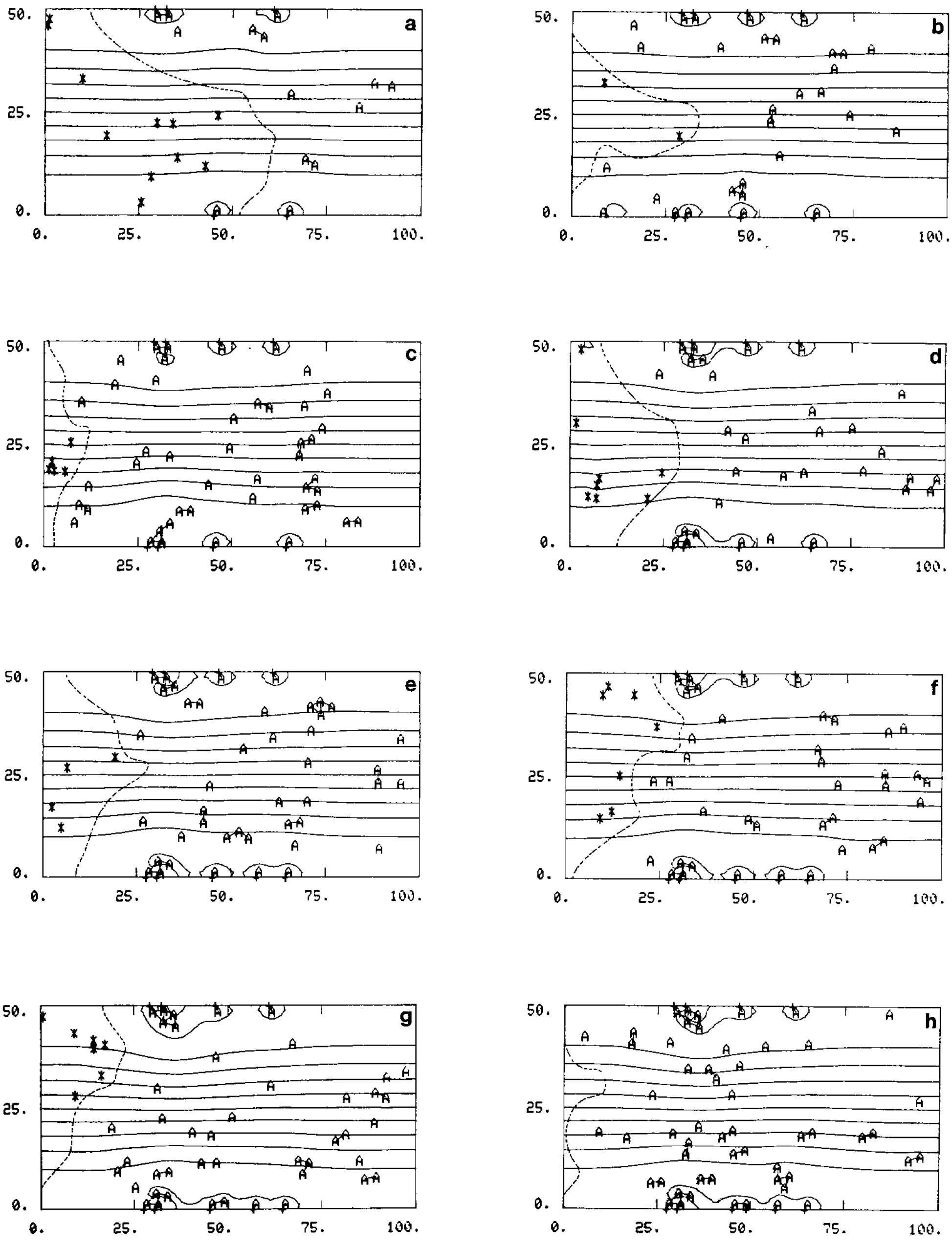
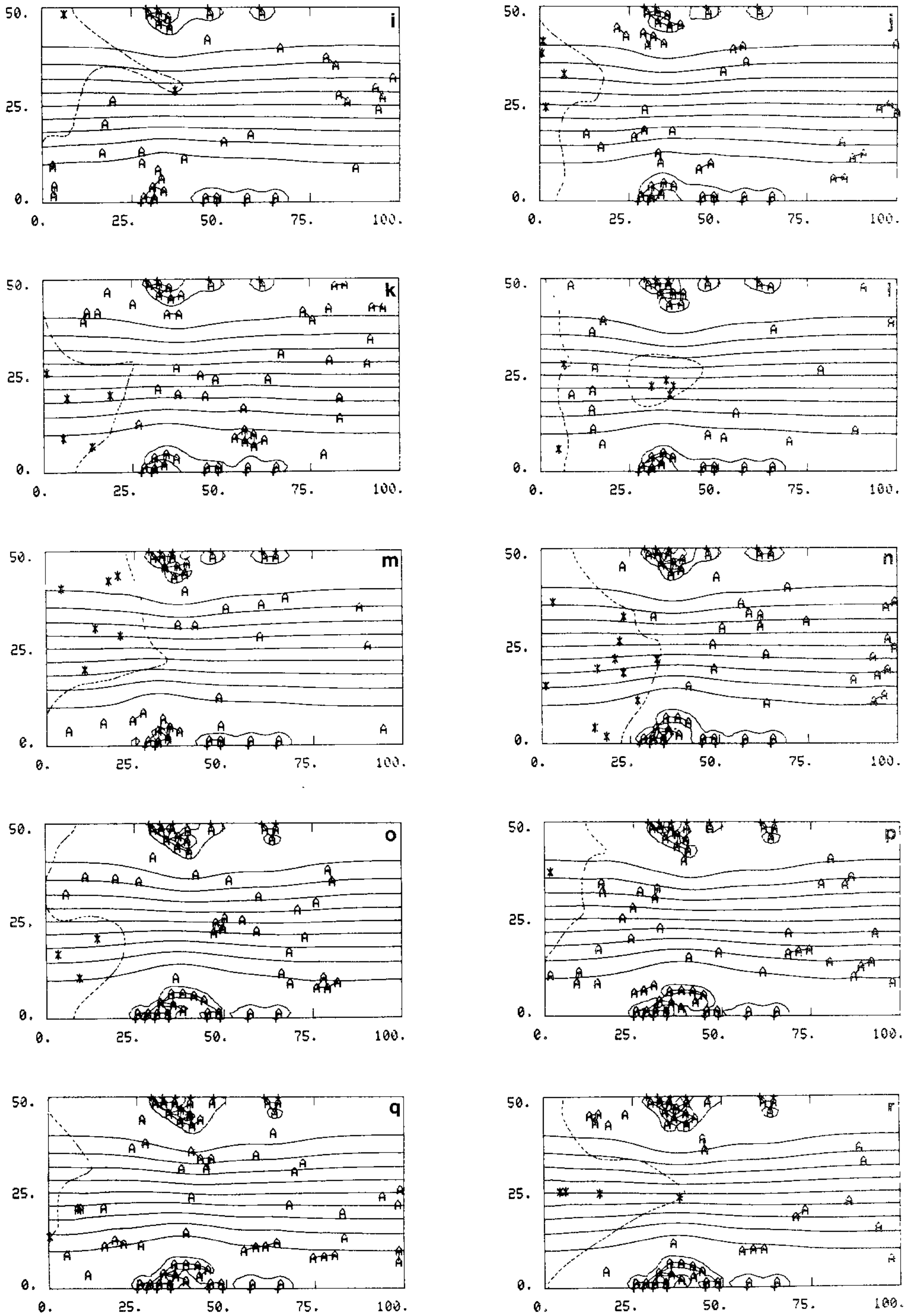
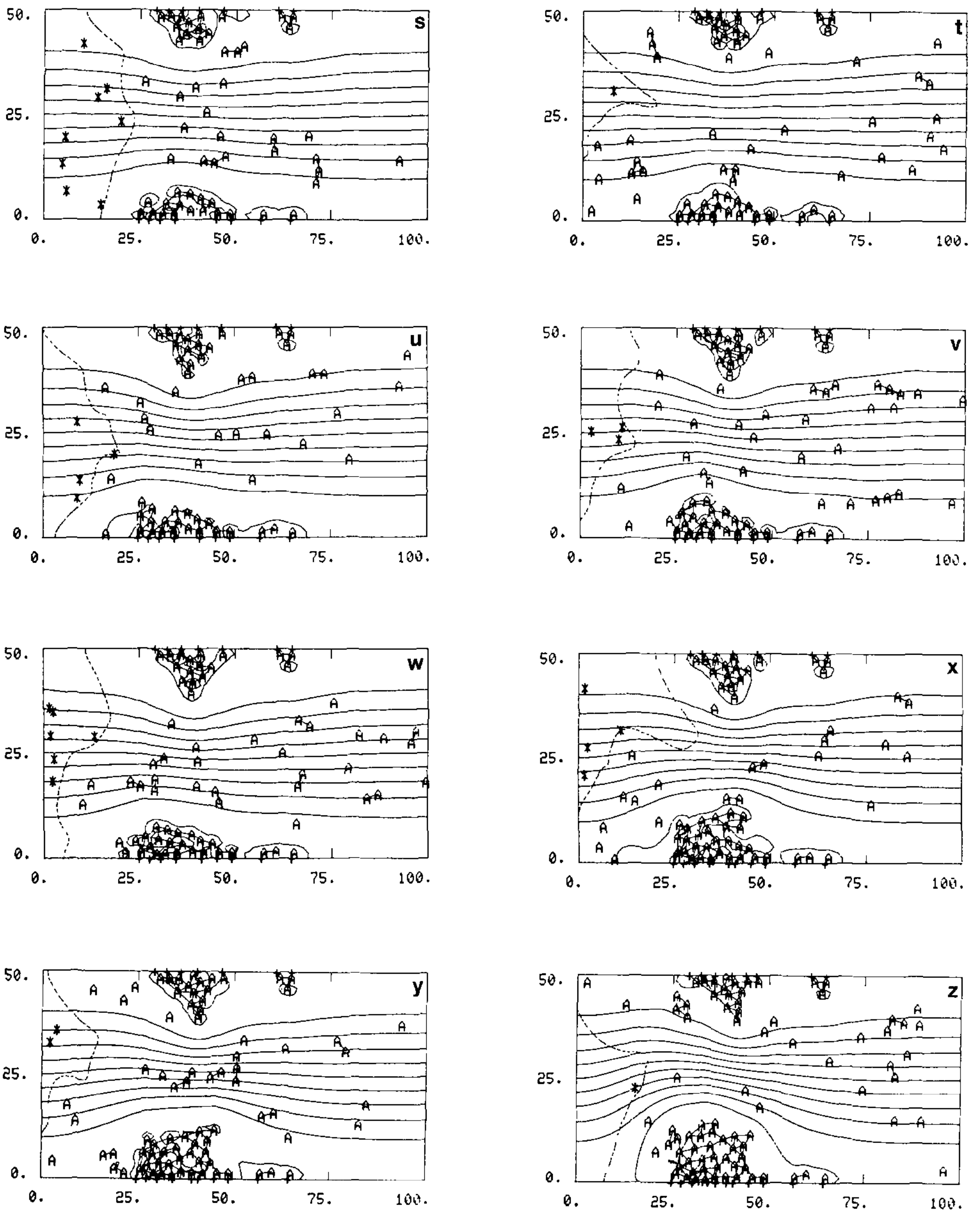


FIG. 5. Formation of aggregates by platelets secreting a normal amount of ADP. Extensive aggregates develop along both injuries. The asymmetry results from the random component of the platelet motion.


i
j
k
l
m
n
o
p
q
r
0.
25.
50.
0.
25.
50.
75.
100.
A


s
t
u
v
w
x
y
z
50.
25.
0.
0.
25.
50.
75.
100.
A

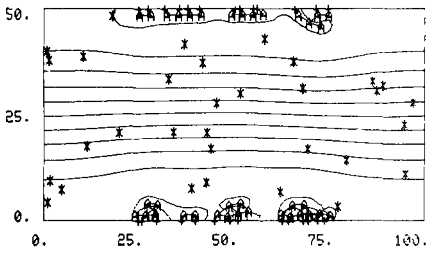


FIG. 6. Formation of aggregates by platelets secreting a pathologically low amount of ADP is severely curtailed. The time is the same as in the last frame in Fig. 5. Aggregation is restricted almost entirely to direct adhesion of platelets to the injured wall.

importantly, to study numerically situations as yet beyond the reach of laboratory experimentation. Of particular interest would be calculations showing the effect on aggregation of protrusions from the vessel wall (e.g., atheroschlerotic plaques) and of vessel bifurcations. The protrusions and the internal vessel boundaries occurring at the bifurcations (of two-dimensional vessels) can be modeled by arrays of point forces which act on the fluid in a manner similar to the interplatelet forces described before (also, see [16]). They can easily be fit into the framework of the existing numerical method.

Several limitations of the numerical method have probably affected our results. Two of the limitations, namely the imposition of upstream and downstream boundary conditions fairly close to the injury site and the restriction of the calculations to two dimensions, arise from considerations of economy. We hope eventually to reduce their impact by extending our calculation to longer segments of three-dimensional vessels. A third limitation is the constraint to use an artificially large ADP diffusion coefficient. This significantly alters the relative importance of diffusive and advective transport of ADP and allows ADP released near the injured wall to influence platelets further upstream and over a greater breadth of the channel than is warranted. This in turn may influence both the sites on the injury at which aggregates develop and the rate of aggregate growth. As mentioned earlier, we are currently exploring the use of particle methods to solve the ADP transport equation with the physiological value for the diffusion coefficient.

APPENDIX A: MERGER OF MODIFIED NEWTON'S METHOD WITH YALE SPARSE SYMMETRIC GAUSSIAN ELIMINATION ALGORITHM

The dense modified factorization developed by Gill and Murray determines one row of U and D during each of $(n-1)$ steps. Assuming the first $(k-1)$ rows are known, the kth row is determined as follows:

Define: $\bar{d}_k = \max \left\{ \left| g_{kk} - \sum_{m=1}^{k-1} c_{mk} u_{mk} \right|, \delta \right\}.$

Define: $c_{kj} = g_{kj} - \sum_{m=1}^{k-1} c_{mj} u_{mk}, \quad \text{for } j = k+1, \ldots, n.$

Define: $\Theta_k = \max_{k < j \leqslant n} \{ |c_{kj}| \}.$

Define:
$$d_{kk} = \bar{d}_k, \quad \text{if } \Theta_k^2 \leqslant \beta^2 \bar{d}_k,$$
$$= \frac{\Theta_k^2}{\beta^2}, \quad \text{if } \Theta_k^2 > \beta^2 \bar{d}_k.$$

Define: $u_{mk} = (c_{mk}/d_{kk}), \quad \text{for } m = 1, \ldots, k-1.$

Here δ and β^2 are preassigned positive constants.

Next we show how this procedure is incorporated into the numerical factorization algorithm of the Yale sparse matrix package. Other portions of the package were used unaltered. We follow the authors of the Yale package Eisenstat *et al.* [2, 3, 4], in introducing the sets:

(i) rg_k: *the set of columns* $j \geqslant k$ *for which* $g_{kj} \neq 0$.

(ii) ru_k: *the ordered set of columns* $j \geqslant k$ *for which* $u_{kj} \neq 0$.

(iii) cu_k: *the set of rows* $i < k$ *for which* $u_{ik} \neq 0$.

The sets rg_k are inputs into the Yale algorithm and are easily retrieved from the compact row-wise storage scheme. The sets ru_k, which detail the nonzero structure of U are determined in a symbolic factorization done prior to the numerical factorization which we describe below. These sets are also readily retrieved from the compact storage scheme. The sets cu_k are determined during the numerical factorization as needed; we omit this part of the description.

In the algorithm below, the lines that we introduced to have the modified factorization performed are indicated by asterisks (*).

SPARSE MATRIX $U^T DU$ FACTORIZATION

```
1.*  $\gamma := \max_k \{ | g_{kk} | \}$;
2.*  $\xi := \max_{j>k} \{ | g_{kj} | \}$;
3.*  $\beta^2 := \max \{ \gamma, \xi/n \}$;
4.   For $k := 1$ to $n$, do
5.     [For $j \in ru_k$ do
6.       [$c_{kj} := 0$];
7.     For $j \in \{ j' \in rg_k : j' > k \}$ do
8.       [$c_{kj} := g_{kj}$];
9.     $u_{kk} := 1$;
10.     $d_{kk} := g_{kk}$;
11.     For $i \in cu_k$ do
12.        [$u_{ik} := c_{ik}/d_{ii}$;
13.         $d_{kk} := d_{kk} - c_{ik}u_{ik}$;
14.     For $j \in \{j' \in ru_i : j' > k\}$ do
15.        [$c_{kj} := c_{kj} - c_{kj}u_{ik}$]];
16.*  $\Theta_k := \max_{m>k}\{|c_{km}|\}$;
17.*  $d_{kk} := \max\{|d_{kk}|, \delta, \Theta_k^2/\beta^2\}$].
```

This completes the modified factorization of G. Forward and back substitution exploiting the sets ru_k can now be done to solve the linear system while avoiding arithmetic with zeros of U. For details on the remainder of the algorithm, see [2–4].

Acknowledgments

The author is deeply grateful to Charles S. Peskin for his continued advice and encouragement. This work was supported at the Courant Institute of Mathematical Sciences in part by a SIMS Societal Fellowship from a grant to SIMS by the Alfred P. Sloan Foundation, in part by the NIH under research Grant HL17859, and in part by the DOE under contract DE-AC02-76ER03077 at the Courant Mathematics and Computing Laboratory of New York University. Continuation of this work at Lawrence Berkeley Laboratory is supported in part by the Director, Office of Basic Energy Sciences, Engineering, Mathematical, and Geosciences Division of the U.S. Department of Energy under contract DE-AC03-76SF00098 and in part by NSF Grant MCS-8211323.

References

1. P. E. Bjorstad, "Numerical Solution of the Biharmonic Equation." Ph.D. thesis, Stanford University, 1980.
2. S. C. Eisenstat, M. H. Shultz, and A. H. Sherman, Efficient Implementation of Sparse Symmetric Gaussian Elimination, *in* "Advances in Computer Methods for Partial Differential Equations" (R. Vichnevetsky, Ed.), AICA, 1975.
3. S. C. Eisenstat, M. H. Shultz, and A. H. Sherman, Application of Sparse Matrix Methods to Partial Differential Equations, *in* "Advances in Computer Methods for Partial Differential Equations" (R. Vichnevetsky, Ed.), AICA, 1975.
4. S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman, Yale Sparse Matrix Package, I. The Symmetric Codes, Yale University Department of Computer Science Report No. 112, New Haven, Conn., 1977.
5. P. E. Gill and W. Murray, *Math. Programming*, 7 (1974), 311.
6. P. E. Gill and W. Murray, "Newton-type Methods for Unconstrained and Linearly Constrained Optimization," Academic Press, New York, 1974.
7. P. E. Gill and W. Murray, "Safeguarded Steplength Algorithms for Optimization Using Descent Methods," National Physical Laboratory Report NAC 37, Teddington, England, 1974.
8. H. L. Goldsmith and T. Karino, Microscopic considerations: The motions of individual particles, *in* "Annals of the New York Academy of Sciences" Vol. 283, (L. Vroman and E. Leonard, Eds.), N.Y. Acad. Sci., New York, 1977.

9. E. F. Grabowski, *Microvasc. Res.* **16** (1978), 159.
10. E. F. Grabowski, J. T. Franta, and P. Didisheim, *Microvasc. Res.* **16** (1978), 183.
11. F. G. Gustavson, Basic techniques for solving sparse systems of linear equation, *in* "Sparse Matrices and Their Applications" (Rose and Willoughby, Eds.), Plenum, New York, 1972.
12. H. Holmsen, L. Salganicoff, and M. H. Fukami, Platelet behavior and biochemistry, *in* "Haemostasis: Biochemistry, Physiology, and Pathology" (Ogston and Bennett, Eds.), Wiley, New York, 1977.
13. D. E. Knuth, "The Art of Computer Programming," Vol. 1, Addison–Wesley, Reading, Mass., 1969.
14. H. Muller-Mohnssen, M. Kratzer, and W. Baldauf, Micro-thrombus formation in models of coronary arteries caused by stagnation point flow arising at the predilection sites of atheroschlerosis and thrombosis, *in* "Proceedings from The Role of Fluid Mechanics in Atherogenesis" (Nerem and Cornhill, Eds.), Ohio State University, Columbus, Ohio, 1978.
15. C. S. Peskin, "Flow Patterns Around Heart Valves: A Digital Computer Method for Solving the Equations of Motion," Ph.D. thesis, Albert Einstein College of Medicine, 1972.
16. C. S. Peskin, *J. Comput. Phys.* **25** (1977), 220.
17. C. S. Peskin and D. M. McQueen, *J. Comput. Phys.* **37** (1980), 113.
18. D. J. Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, *in* "Graph Theory and Computing" (R.C. Read, Ed.), Academic Press, New York, 1972.
19. V. T. Turitto and H. R. Baumgartner, *Microvasc. Res.* **9** (1975), 335.
20. V. T. Turitto, H. J. Weiss, and H. R. Baumgartner, *Microvasc. Res.* **19** (1980), 352.
21. H. J. Weiss, *New Engl. J. Med.* **293** (1975), 531.
22. H. J. Weiss, *New Engl. J. Med.* **293** (1975), 580.
23. H. J. Weiss, T. B. Tschopp, and H. R. Baumgartner, *New Engl. J. Med.* **293** (1975), 619.