JOURNAL OF COMPUTATIONAL PHYSICS 105, 33–46 (1993)

Improved Volume Conservation in the Computation of Flows with Immersed Elastic Boundaries

CHARLES S. PESKIN

Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, New York 10012

AND

BETH FELLER PRINTZ

Department of Physiology, Mt. Sinai School of Medicine, Gustave Levy Place, New York, New York 10029

Received April 30, 1990; revised June 1, 1992

This paper introduces a recipe for the construction of a finite-difference divergence operator the coefficients of which are completely determined once an interpolation scheme has been chosen. Substitution of this divergence operator for the previously used divergence based on central differences makes a dramatic improvement in the overall volume conservation that is observed in an immersed-boundary computation. This improvement is particularly important for computations in which an elastic boundary separates chambers containing fluid at substantially different pressures, a situation which is prominent in cardiac fluid dynamics during the contraction of the ventricles.

INTRODUCTION

This paper is concerned with an improvement in a method that was first introduced by Peskin for the study of flow patterns around heart valves [1–5]. This immersed-boundary method has since been applied to a variety of problems in two and three space dimensions, including blood flow in the heart [6, 7], the design of prosthetic cardiac valves [8, 9], platelet aggregation during blood clotting [10], aquatic animal locomotion [11], wave propagation in the cochlea [12], and the flow of suspensions [13].

Throughout this work, it was noted from time to time that volume conservation was not exact, and indeed that there was a systematic tendency for a closed pressurized chamber to lose volume slowly at a rate proportional to the pressure difference across its walls, almost as though the fluid were leaking out through a porous boundary. The obvious explanation that fluid was indeed escaping between the discrete marker particles that computationally define the wall was ruled out by two observations: (1) fluid markers placed halfway between the wall markers move with the wall and are not pushed out of it, and (2) increasing the density of wall markers (without refining the grid of the fluid computation) does not reduce the leak.

In any case, the effect was small enough to be tolerable in the applications described above. In the cardiac research, however, this was only because the computational experiments were primarily concerned with diastole, during which the heart walls are relaxed and the pressure in the cardiac chambers is low. In more recent work concerning ventricular systole, we have found that the volume loss is too large to ignore.

This paper reports on a change in the computational method that virtually eliminates the volume error in question. The key idea is the introduction of a new finite-difference divergence operator which is constructed in such a way that the interpolated velocity field in which the immersed boundary moves is more nearly divergence-free. Because of the modular character of the method, this change is easy to implement. It does not substantially increase the cost of the computations

SUMMARY OF THE IMMERSED-BOUNDARY METHOD

Consider a closed elastic curve immersed in a two-dimensional incompressible fluid. The equations of motion of the system may be written as follows:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p=\mu\nabla^2\mathbf{u}+\mathbf{F} \tag{1}$$

$$\nabla\cdot\mathbf{u}=0 \tag{2}$$

$$\mathbf{F}(\mathbf{x},t)=\int_0^L \mathbf{f}(s,t)\,\delta(\mathbf{x}-\mathbf{X}(s,t))\,ds \tag{3}$$

0021-9991/93 $5.00

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{u}(\mathbf{X}(s,t),t) = \int \mathbf{u}(\mathbf{x},t)\,\delta(\mathbf{x}-\mathbf{X}(s,t))\,d\mathbf{x} \quad (4)$$

$$\mathbf{f}(s,t) = \frac{\partial}{\partial s}(T\tau) \quad (5)$$

$$T = S\left(\left|\frac{\partial \mathbf{X}}{\partial s}\right| - 1\right) \quad (6)$$

$$\tau = \frac{\partial \mathbf{X}/\partial s}{|\partial \mathbf{X}/\partial s|}. \quad (7)$$

Here $\mathbf{u}(\mathbf{x}, t)$ is the fluid velocity, $p(\mathbf{x}, t)$ is the fluid pressure, and $\mathbf{F}(\mathbf{x}, t)$ is the force-density generated by the boundary in the fluid. ($\mathbf{F}(\mathbf{x}, t)$ is a δ-function layer, see Eq. (2)). The constants ρ and μ are the fluid density and viscosity. The boundary configuration is given by $\mathbf{x} = \mathbf{X}(s, t)$, $0 \leqslant s \leqslant L$, where fixed s marks a material point and L is the unstressed length of the boundary. Since the boundary is a closed curve, $\mathbf{X}(0, t) = \mathbf{X}(L, t)$. The force exerted by the element of boundary ds on the fluid is $\mathbf{f}(s, t)\,ds$. This force is computed from the boundary tension $T(s, t)$ and the unit tangent $\tau(s, t)$, see Eq. (5).

Since the formulation given above is somewhat unusual, it is important to remark that it can be reduced to a standard formulation in terms of boundary conditions which hold on the immersed elastic interface. The only technical assumption we need in order to make this reduction is that the fluid velocity is continuous across the interface. (Without this assumption Eq. (4) becomes ambiguous.) Then Eq. (4) is equivalent to the no-slip condition that the fluid sticks to the boundary. Note that this condition here plays the role of an equation of motion for the boundary rather than that of a constraint on the fluid motion, since the boundary motion is not specified in advance. The remaining boundary condition concerns the jump in stress across the immersed boundary. To find it, combine Eqs. (1) and (3) and write the result as follows

$$\rho \frac{Du_i}{Dt} = \frac{\partial \sigma_{ij}}{\partial x_j} + \int_0^L f_i(s)\,\delta(\mathbf{x}-\mathbf{X}(s,t))\,ds, \quad (8)$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla \quad (9)$$

is the material derivative and where

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (10)$$

is the fluid stress tensor. In the foregoing we have used the summation convention that a repeated index is summed from 1 to 2. Now integrate Eq. (8) over an arbitrary region $\Omega(t)$ that moves with the fluid. Let the boundary of Ω be denoted $\partial\Omega$, let $\mathbf{n}$ be the outward normal on $\partial\Omega$, and let dl be the arc length on $\partial\Omega$. The result is

$$\frac{d}{dt}\int_{\Omega(t)} \rho u_i\,d\mathbf{x} = \int_{\partial\Omega(t)} \sigma_{ij} n_j\,dl + \int_{\{s:\,\mathbf{X}(s,t)\in\Omega(t)\}} f_i(s,t)\,ds. \quad (11)$$

Finally, let $\Omega(t)$ collapse to a thin strip of finite length with the immersed boundary running down its center. Take the limit as the width of the strip approaches zero. Then

$$0 = \int ([\sigma_{ij} n_j]\,dl + f_i\,ds), \quad (12)$$

where $\mathbf{n}$ is now the normal on one side of the immersed boundary and where [ ] denotes the jump in a quantity across the immersed boundary. Since the domain of integration is arbitrary

$$[\sigma_{ij}]\,n_j = -f_i\frac{ds}{dl} = -\frac{f_i}{|\partial \mathbf{X}/\partial s|}. \quad (13)$$

Taking normal and tangential components, we get

$$n_i[\sigma_{ij}]\,n_j = -\frac{\mathbf{f}\cdot\mathbf{n}}{|\partial \mathbf{X}/\partial s|}, \quad (14)$$

$$\tau_i[\sigma_{ij}]\,n_j = -\frac{\mathbf{f}\cdot\tau}{|\partial \mathbf{X}/\partial s|}, \quad (15)$$

where $\tau = (\partial \mathbf{X}/\partial s)/|\partial \mathbf{X}/\partial s|$ is the unit tangent to the immersed boundary. Now

$$n_i[\sigma_{ij}]\,n_j = -[p] \quad (16)$$

and

$$\tau_i[\sigma_{ij}]\,n_j = \mu\tau\cdot[\mathbf{n}\cdot\nabla\mathbf{u}] = \mu\tau\cdot\left[\frac{\partial \mathbf{u}}{\partial n}\right], \quad (17)$$

where $\partial/\partial n$ denotes the normal derivative. We leave the proofs of Eqs. (16)–(17) as exercises for the reader with the following hints: To prove (16) make use of the identity $n_i n_j + \tau_i\tau_j = \delta_{ij}$, the fact that $\nabla\cdot\mathbf{u} = 0$, and the fact that $[\tau_i(\partial u_j/\partial x_i)] = [(\tau\cdot\nabla\mathbf{u})_j] = 0$ since $[\mathbf{u}] = 0$ and since $\tau\cdot\nabla$ is the *tangential* derivative. To prove (17), use the last of these facts and also $\tau\cdot\mathbf{n} = 0$.

It follows from Eqs. (14)–(17) that

$$[p] = \frac{\mathbf{f}\cdot\mathbf{n}}{|\partial \mathbf{X}/\partial s|} \quad (18)$$

$$\mu\tau\cdot\left[\frac{\partial \mathbf{u}}{\partial n}\right] = -\frac{\mathbf{f}\cdot\tau}{|\partial \mathbf{X}/\partial s|}, \quad (19)$$

which are the expected conditions for force balance at a massless interface along which the force $\mathbf{f}\,ds$ is applied to the interval ds of the boundary. (The factor $|\partial \mathbf{X}/\partial s|$ appears because s is not arc length in general.)

We have derived Eqs. (18)–(19) only to show that our formulation of the problem in terms of a delta-function layer of force reduces to well-known boundary conditions for the jump in stress at an interface. The strength of our formulation, however, is that it makes no explicit use of these jump conditions, which would be awkward to discretize along a moving boundary, the motion of which is unknown.

The discretization and numerical solution of Eqs. (1)–(7) proceed as follows [1–5]. Let each independent variable be restricted to a lattice of equally spaced points: $\mathbf{x} = (jh, kh)$, $s = l\,\Delta s$, $t = n\,\Delta t$, where j, k, l, n are integers and where $h = \Delta x = \Delta y$. In particular we shall use notations such as $\sum_{\mathbf{x}}$ and $\sum_s$ to denote sum over these computational lattices. Let D^+, D^-, D^0 denote the forward, backward, and centered divided difference operators. A subscript on such an operator indicates the variable to which it should be applied. Let the time step index n be written as a superscript, so that $\mathbf{u}^n(\mathbf{x}) = \mathbf{u}(\mathbf{x}, n\,\Delta t)$. The goal of the method is to compute $(\mathbf{u}^{n+1}, \mathbf{X}^{n+1})$ from $(\mathbf{u}^n, \mathbf{X}^n)$. This is done as follows:

(1) Find the boundary force $\mathbf{f}^n$ from the boundary configuration $\mathbf{X}^n$:

$$T^n = S(|D_s^+ \mathbf{X}^n| - 1) \tag{20}$$

$$\boldsymbol{\tau}^n = \frac{D_s^+ \mathbf{X}^n}{|D_s^+ \mathbf{X}^n|} \tag{21}$$

$$\mathbf{f}^n = D_s^-(T^n \boldsymbol{\tau}^n). \tag{22}$$

Remark. This is an *explicit* method for the computation of the boundary force. For more stable alternatives, see [1–5, 14].

(2) Apply the force $\mathbf{f}^n$ to the grid of the fluid computation:

$$\mathbf{F}^n(\mathbf{x}) = \sum_s \mathbf{f}^n(s)\,\delta_h(\mathbf{x} - \mathbf{X}^n(s))\,\Delta s, \tag{23}$$

where, for $\mathbf{x} = (x, y)$,

$$\delta_h(\mathbf{x}) = \delta_h(x)\,\delta_h(y) \tag{24}$$

and

$$\delta_h(x) = \begin{cases} \dfrac{1}{4h}\left(1 + \cos\dfrac{\pi x}{2h}\right) & |x| \leqslant 2h \\ 0 & |x| \geqslant 2h. \end{cases} \tag{25}$$

(3) Update the fluid velocity under the influence of the force density $\mathbf{F}^n$. This is done by Chorin's projection method [15, 16] which solves the following systems successively for $\mathbf{u}^{n+1,0}$, $\mathbf{u}^{n+1,1}$, $\mathbf{u}^{n+1,2}$, $(\mathbf{u}^{n+1}, p^{n+1})$:

$$\rho\,\frac{\mathbf{u}^{n+1,0} - \mathbf{u}^n}{\Delta t} = \mathbf{F}^n \tag{26}$$

$$\rho\left(\frac{\mathbf{u}^{n+1,1} - \mathbf{u}^{n+1,0}}{\Delta t} + u_x^n\,D_x^0 \mathbf{u}^{n+1,1}\right) = \mu D_x^+\,D_x^- \mathbf{u}^{n+1,1} \tag{27}$$

$$\rho\left(\frac{\mathbf{u}^{n+1,2} - \mathbf{u}^{n+1,1}}{\Delta t} + u_y^n\,D_y^0 \mathbf{u}^{n+1,2}\right) = \mu D_y^+\,D_y^- \mathbf{u}^{n+1,2} \tag{28}$$

$$\begin{gathered} \rho\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1,2}}{\Delta t}\right) + \mathbf{D}p^{n+1} = 0 \\ \mathbf{D}\cdot\mathbf{u}^{n+1} = 0. \end{gathered} \tag{29}$$

(The proper choice of $\mathbf{D}$ is the subject of this paper; see below.)

(4) Interpolate the new velocity to the old boundary positions and move the boundary points:

$$\mathbf{X}^{n+1}(s) = \mathbf{X}^n(s) + \Delta t \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x})\,\delta_h(\mathbf{x} - \mathbf{X}^n(s))\,h^2, \tag{30}$$

where δ_h is the approximate δ-function defined above, see Eqs. (24)–(25).

This completes the time step.

The vector difference operator $\mathbf{D}$ corresponding to the vector derivative operator ∇ has been left unspecified in the foregoing. In all previous work we used $\mathbf{D} = (D_x^0, D_y^0)$. An improved recipe for $\mathbf{D}$ is introduced in the next section.

CONSTRUCTION OF A DIVERGENCE OPERATOR BASED ON THE INTERPOLATION SCHEME

The velocity field in which the boundary points actually move is not the discrete velocity field $\mathbf{u}(\mathbf{x})$, which is defined only at grid points, but the interpolated velocity field $\mathbf{U}(\mathbf{X})$, which is defined everywhere by the formula

$$\mathbf{U}(\mathbf{X}) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\,\delta_h(\mathbf{x} - \mathbf{X})\,h^2. \tag{31}$$

Because of the properties of δ_h (see Eqs. (24)–(25)), $\mathbf{U}(\mathbf{X})$ is continuous and has continuous first derivatives. Thus $\nabla\cdot\mathbf{U}$ is well-defined, and we can construct the operator $\mathbf{D}$ in such a way that

$$(\mathbf{D}\cdot\mathbf{u})(\mathbf{x}) = h^{-2}\int_{B(\mathbf{x})} (\nabla\cdot\mathbf{U})(\mathbf{X})\,d\mathbf{X}, \tag{32}$$

where $B(\mathbf{x})$ is a square box of side h alligned parallel to the grid and centered on the grid point $\mathbf{x}$. In other words, $\mathbf{D}\cdot\mathbf{u}$ at the grid point $\mathbf{x}$ is defined as the average value of $\nabla\cdot\mathbf{U}$

over the box $B(\mathbf{x})$ centered on $\mathbf{x}$. For the actual evaluation of the operator $\mathbf{D}$, we rewrite Eq. (32) with the aid of the divergence theorem

$$(\mathbf{D}\cdot\mathbf{u})(\mathbf{x}) = h^{-2}\int_{-h/2}^{h/2} U_x(x+X,\, y+Y)\,dY\Big|_{X=-h/2}^{X=h/2} + h^{-2}\int_{-h/2}^{h/2} U_y(x+X,\, y+Y)\,dX\Big|_{Y=-h/2}^{Y=h/2}, \quad (33)$$

where we use the notation $\mathbf{x}=(x, y)$, $\mathbf{X}=(X, Y)$, $\mathbf{U}=(U_x, U_y)$. Substituting into this equation the expression for $\mathbf{U}$ from Eq. (3) and making use of the fact that δ_h is even, we find

$$\begin{aligned}(\mathbf{D}\cdot\mathbf{u}) &= \sum_{\mathbf{x}'} u_x(x', y')\,\delta_h(X+x-x')\Big|_{X=-h/2}^{X=h/2} \\ &\quad\times\int_{-h/2}^{h/2}\delta_h(Y+y-y')\,dY \\ &\quad+\sum_{\mathbf{x}'} u_y(x', y')\int_{-h/2}^{h/2}\delta_h(X+x-x') \\ &\quad\times dX\,\delta_h(Y+y-y')\Big|_{Y=-h/2}^{Y=h/2} \\ &= \sum_{\mathbf{x}'}[u_x(x', y')\,\gamma(x-x')\,\omega(y-y') \\ &\quad+u_y(x', y')\,\omega(x-x')\,\gamma(y-y')], \end{aligned} \quad (34)$$

where we have introduced the notation

$$\gamma(x) = \delta_h(x+X)\big|_{X=-h/2}^{X=h/2} \quad (35)$$

$$\omega(x) = \int_{-h/2}^{h/2}\delta_h(x+X)\,dX. \quad (36)$$

(We shall only make use of the values of γ and ω on the lattice points $x=jh$.) Note that

$$\sum_x \gamma(x) = \delta_h(\infty)-\delta_h(-\infty)=0 \quad (37)$$

$$\sum_x \omega(x) = \int_{-\infty}^{\infty}\delta_h(x)\,dx = 1. \quad (38)$$

Thus the values of γ are like coefficients of a difference operator and the values of ω are like coefficients of an averaging operator.

From Eq. (34) we may read off the form of the vector difference operator $\mathbf{D}$. Its components D_x and D_y are given by

$$(D_x\phi)(x, y) = \sum_{x'y'}\phi(x', y')\,\gamma(x-x')\,\omega(y-y') \quad (39)$$

$$(D_y\phi)(x, y) = \sum_{x'y'}\phi(x', y')\,\omega(x-x')\,\gamma(y-y'). \quad (40)$$

These equations (together with the definitions of γ and ω, Eqs. (35)–(36)) give a definite recipe for the construction of $\mathbf{D}$ in terms of any choice that one might happen to make of the function δ_h. In our particular case, the values of γ and ω are found as follows.

Since δ_h is even, γ is odd and ω is even. Also, since $\delta_h(x)=0$ for $|x|\geq 2h$, $\gamma(x)=\omega(x)=0$ for $|x|\geq 5h/2$. We have

$$\gamma(0) = \delta_h\left(\frac{h}{2}\right)-\delta_h\left(-\frac{h}{2}\right)=0 \quad (41)$$

$$-\gamma(-h)=\gamma(h)=\delta_h\left(\frac{3h}{2}\right)-\delta_h\left(\frac{h}{2}\right)=-\frac{1}{2h}\frac{\sqrt{2}}{2} \quad (42)$$

$$-\gamma(-2h)=\gamma(2h)=\delta_h\left(\frac{5h}{2}\right)-\delta_h\left(\frac{3h}{2}\right) = -\frac{1}{4h}\left(1-\frac{\sqrt{2}}{2}\right) \quad (43)$$

$$\gamma(jh)=0 \qquad \text{for} \quad |j|\geq 3 \quad (44)$$

$$\omega(0) = \int_{-h/2}^{h/2}\delta_h(x)\,dx = \frac{1}{4}+\frac{\sqrt{2}}{2\pi} \quad (45)$$

$$\omega(-h)=\omega(h)=\int_{h/2}^{3h/2}\delta_h(x)\,dx=\frac{1}{4} \quad (46)$$

$$\omega(-2h)=\omega(2h)=\int_{3h/2}^{2h}\delta_h(x)\,dx=\frac{1}{8}-\frac{\sqrt{2}}{4\pi} \quad (47)$$

$$\omega(jh)=0 \qquad \text{for} \quad |j|\geq 3. \quad (48)$$

This completes the construction of the vector difference operator $\mathbf{D}=(D_x, D_y)$. The value of $\mathbf{D}\cdot\mathbf{u}$ at a particular grid point $\mathbf{x}$ is a function of the values of $\mathbf{u}$ on a 5×5 stencil of the grid centered on $\mathbf{x}$.

IMPLEMENTATION OF THE PROJECTION OPERATION

Chorin's projection method gets its name from the system of Eqs. (17), which we rewrite here in the form

$$\rho\frac{\mathbf{u}-\mathbf{w}}{\Delta t}+\mathbf{D}p=0, \qquad \mathbf{D}\cdot\mathbf{u}=0 \quad (49)$$

These equations define the orthogonal projection of $\mathbf{w}$ onto the space of vector fields that are divergence-free in the sense that $\mathbf{D}\cdot\mathbf{u}=0$. This is the only place that the operator $\mathbf{D}$ appears in the computational method.

Given $\mathbf{w}$, the system of Eqs. (49) can be solved for the unknowns $(\mathbf{u}, p)$ in the following way. First evaluate

$$\mathbf{q} = -\frac{\rho}{\Delta t}\mathbf{D}\cdot\mathbf{w}. \quad (50)$$

Next, solve a discrete Poisson equation for p:

$$-\mathbf{D}\cdot\mathbf{D}p = q. \tag{51}$$

Finally, evaluate $\mathbf{u}$ as follows:

$$\mathbf{u} = \mathbf{w} - \frac{\Delta t}{\rho}\mathbf{D}p. \tag{52}$$

It is easy to check that this procedure constructs a solution to Eqs. (49). First note that Eq. (52) is just a rewritten version of the first equation in the system of Eqs. (49). Then, to show that $\mathbf{D}\cdot\mathbf{u} = 0$, apply $\mathbf{D}\cdot$ to both sides of Eq. (52) and make use of Eqs. (50)–(51).

We now consider the solution of the discrete Poisson equation $-\mathbf{D}\cdot\mathbf{D}p = q$. This will be done by the discrete Fourier transform.

Up to now we have not made any statement about the domain occupied by the fluid in which our elastic boundary is immersed. Suppose this domain is a square periodic box covered by an $N\times N$ grid. We define the discrete Fourier transform of a grid function $p_{jk} = p(jh, kh)$ by the formula

$$\hat{p}_{lm} = \sum_{jk} p_{jk} e^{-i(2\pi/N)(jl+km)}. \tag{53}$$

The inversion formula is

$$p_{jk} = \frac{1}{N^2}\sum_{lm} \hat{p}_{lm} e^{i(2\pi/N)(jl+km)} \tag{54}$$

and the discrete convolution theorem reads as follows:

$$\sum_{j'k'} p_{j'k'} q_{j-j',k-k'} = \frac{1}{N^2}\sum_{lm} \hat{p}_{lm}\hat{q}_{lm} e^{i(2\pi/N)(jl+km)}. \tag{55}$$

Applying this convolution theorem to the definition of $\mathbf{D}$, Eqs. (39)–(40), we see that

$$(D_x\phi)\hat{}_{lm} = \hat{\gamma}_l\hat{\omega}_m\hat{\phi}_{lm} \tag{56}$$

$$(D_y\phi)\hat{}_{lm} = \hat{\omega}_l\hat{\gamma}_m\hat{\phi}_{lm}. \tag{57}$$

These results may be summarized by the statement that

$$\hat{\mathbf{D}}_{lm} = (\hat{\gamma}_l\hat{\omega}_m, \hat{\omega}_l\hat{\gamma}_m), \tag{58}$$

which defines the Fourier transform of the vector difference operator $\mathbf{D}$. The Fourier transform of the discrete Poisson equation $-\mathbf{D}\cdot\mathbf{D}p = q$ therefore reads as follows:

$$-(\hat{\gamma}_l^2\hat{\omega}_m^2 + \hat{\omega}_l^2\hat{\gamma}_m^2)\,\hat{p}_{lm} = \hat{q}_{lm} \tag{59}$$

and, of course, the solution of this equation for $\hat{p}$ is

$$\hat{p}_{lm} = \frac{\hat{q}_{lm}}{-(\hat{\gamma}_l^2\hat{\omega}_m^2 + \hat{\omega}_l^2\hat{\gamma}_m^2)}. \tag{60}$$

It remains only to evaluate $\hat{\gamma}$ and $\hat{\omega}$ (and to be careful of those cases in which the denominator in Eq. (60) is zero!).

The values of γ and ω are listed in Eqs. (41)–(48). From these we conclude that

$$\begin{aligned}
\hat{\gamma}_l &= \frac{\sqrt{2}}{2}\frac{1}{2h}2i\sin\frac{2\pi l}{N} + \left(1-\frac{\sqrt{2}}{2}\right)\frac{1}{4h}2i\sin\frac{4\pi l}{N} \\
&= \frac{i}{h}\left(\sin\frac{2\pi l}{N}\right)\left(\frac{\sqrt{2}}{2} + \left(1-\frac{\sqrt{2}}{2}\right)\cos\frac{2\pi l}{N}\right)
\end{aligned} \tag{61}$$

$$\begin{aligned}
\hat{\omega}_l &= \left(\frac{1}{4}+\frac{\sqrt{2}}{2\pi}\right) + \frac{1}{4}2\cos\frac{2\pi l}{N} + \left(\frac{1}{8}-\frac{\sqrt{2}}{4\pi}\right)2\cos\frac{4\pi l}{N} \\
&= \left(\frac{1}{4}+\frac{\sqrt{2}}{2\pi}\right) + \frac{1}{2}\left(1-2\sin^2\frac{\pi l}{N}\right) \\
&\quad + \left(\frac{1}{4}-\frac{\sqrt{2}}{2\pi}\right)\left(1-2\sin^2\frac{2\pi l}{N}\right) \\
&= 1-\sin^2\frac{\pi l}{N} - \left(\frac{1}{2}-\frac{\sqrt{2}}{\pi}\right)4\sin^2\frac{\pi l}{N}\cos^2\frac{\pi l}{N} \\
&= \left(\cos^2\frac{\pi l}{N}\right)\left(1-2\left(1-\frac{2\sqrt{2}}{\pi}\right)\sin^2\frac{\pi l}{N}\right).
\end{aligned} \tag{62}$$

Note that $\hat{\gamma}_l = 0$ if and only if $\sin(2\pi l/N) = 0$. The other factor in $\hat{\gamma}_l$ cannot be zero, since $1-\sqrt{2}/2 < \sqrt{2}/2$. Similarly, $\hat{\omega}_l = 0$ if and only if $\cos^2\pi l/N = 0$. The other factor in $\hat{\omega}_l$ cannot be zero, since $2(1-2\sqrt{2}/\pi) < 1$. Thus $\hat{\gamma}_l = 0$ where $l = 0$, $N/2$ and $\hat{\omega}_l = 0$ when $l = N/2$. These are the only zeros of $\hat{\gamma}$ and $\hat{\omega}$ on the interval $[0\cdots(N-1)]$.

We now consider the denominator of Eq. (60), which may be written in the form $(-\hat{\gamma}_l^2)\,\hat{\omega}_m^2 + \hat{\omega}_l^2(-\hat{\gamma}_m^2)$. Since $\hat{\gamma}$ is imaginary and $\hat{\omega}$ is real, both terms in this expression are real and nonnegative. The expression as a whole can only be zero if both of its terms are zero. This will happen if $l = N/2$, if $m = N/2$, or if $(l, m) = (0, 0)$. There are no other cases. (We consider only those possibilities in which l and m are integers in the interval $[0\cdots(N-1)]$.)

The values of (l, m) for which the denominator in Eq. (60) is zero are precisely the values of (l, m) for which $\hat{\mathbf{D}}_{lm} = (0, 0)$. Since q is computed by applying $\mathbf{D}\cdot$ to some vector, the numerator in Eq. (60) is also zero at these wave numbers. Thus $\hat{p}_{lm}$ is undefined in these cases. Its value is of no consequence, however, since $\hat{\mathbf{D}}\hat{p}_{lm}$ is zero for these special wave numbers. In practice, we set $\hat{p}_{lm} = 0$ for $(l, m) = (0, 0)$, for $l = N/2$ (with any m), and for $m = N/2$ (with any l).

In the practical implementation of this method, we make

use of an array $\hat{a}_{lm}$, which is computed and stored during preprocessing, and which is defined as follows:

$$\hat{a}_{lm} = \begin{cases} 0, & (l, m) = (0, 0),\ l = N/2, \text{ or } m = N/2, \\ \dfrac{1}{(-\hat{\gamma}_l^2)(\hat{\omega}_m^2) + \hat{\omega}_l^2(-\hat{\gamma}_m^2)}, & \text{otherwise.} \end{cases} \tag{63}$$

Then, when it is time to compute $\hat{p}$ from $\hat{q}$, we simply use

$$\hat{p}_{lm} = \hat{a}_{lm}\hat{q}_{lm} \tag{64}$$

for all (l, m) including the special values noted above.

The overall algorithm for the solution of the discrete Poisson equation $-\mathbf{D} \cdot \mathbf{D}p = q$ may now be summarized as follows

$$p = (FFT)^{-1}(\hat{a}(FFT)(q)), \tag{65}$$

where (FFT) denotes the Fast Fourier Transform algorithm and where $(FFT)^{-1}$ stands for the version of that algorithm that applies the inverse Fourier transform. Because the array $\hat{a}$ is computed during preprocessing, the computational effort involved in the solution of the discrete Poisson equation is identical to what it was in our earlier work in which $\mathbf{D}$ was simply (D_x^0, D_y^0). The only change is that the array $\hat{a}$ now holds different numbers than before. Somewhat more work is required, however, in the steps that immediately precede and follow the solution of the discrete Poisson equation. These are the steps in which the operator $\mathbf{D}$ is directly applied to data, see Eqs. (50) and (52), and the additional work is a consequence of the larger stencil of $\mathbf{D}$.

DISCUSSION OF AN ALTERNATIVE APPROACH TO IMPROVED VOLUME CONSERVATION

Since we use a Fourier method to solve the projection equations (Eqs. (49)), the following alternative approach to improved volume conservation may occur to the reader: why not use a finite Fourier series to represent $\mathbf{w}$, perform a *continuous* projection of $\mathbf{w}$ to obtain a *divergence-free* Fourier series for $\mathbf{u}$, and then *evaluate* this series at points of the immersed boundary in order to define the velocity with which the boundary moves? The practical difficulty with this approach involves the last step: evaluating the Fourier series at the points of the immersed boundary. Since these points do not lie on a regular grid, the Fast Fourier Transform is not suitable for the purpose. Since the number of boundary points is $O(N)$ and the number of nonzero Fourier coefficients would be $O(N^2)$, the computational work required to evaluate the boundary velocity in this alternative manner would be $O(N^3)$. This alternative is therefore expensive and has not, in fact, been tried.

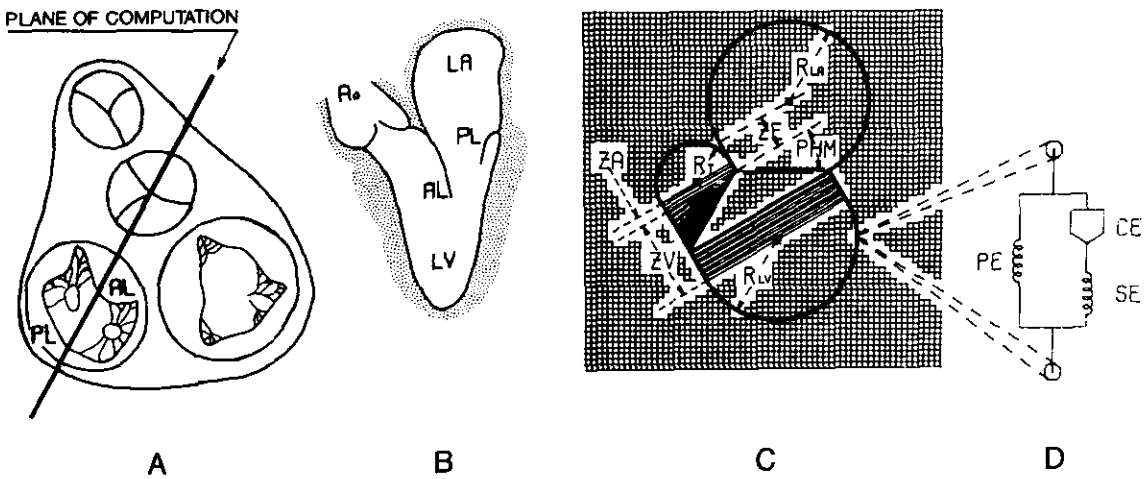

FIG. 1. The heart model. (A) View of the cardiac ventricles with the atria and arteries cut away. The heart is shown in ventricular diastole with the inflow valves open and the outflow valves closed. The plane of the two-dimensional computation is indicated by a heavy line. It bisects the posterior leaflet (PL) and the anterior leaflet (AL) of the mitral valve. Left heart structures cut by the plane of the computation are shown in (B). Besides the two leaflets of the mitral valve, they include the left atrium (LA), the left ventricle (LV), and the root of the aorta (Ao). The initial configuration of the computational model is shown in (C). The heart model is immersed in a 64×64 square lattice which is used for the fluid computation. Points of the heart wall move freely through this lattice without being constrained to lie at lattice intersections. Parameters (R_{LV}, ZV, ZA, R_T, ZE, R_{LA}, and PHM) control the initial geometry of the simulated left heart chamber. There is a source in the left atrium to model the pulmonary venous return, and a sink equipped with a valve in the aortic outflow tract to allow for the ejection of blood into the aorta during systole. There is also a source/sink external to the heart to accommodate changes in cardiac volume. Pairs of boundary points are connected by links which have the mechanical properties of cardiac muscle as indicated in (D); each link includes a parallel elastic element (PE), a series elastic element (SE) and a contractile element (CE). The elastic elements are springs and the contractile element generates a force which depends on its velocity of shortening, see [6] for details. The cross links shown in (C) generate force on the wall at their ends; they do not interfere with flow in the interior of the chamber.

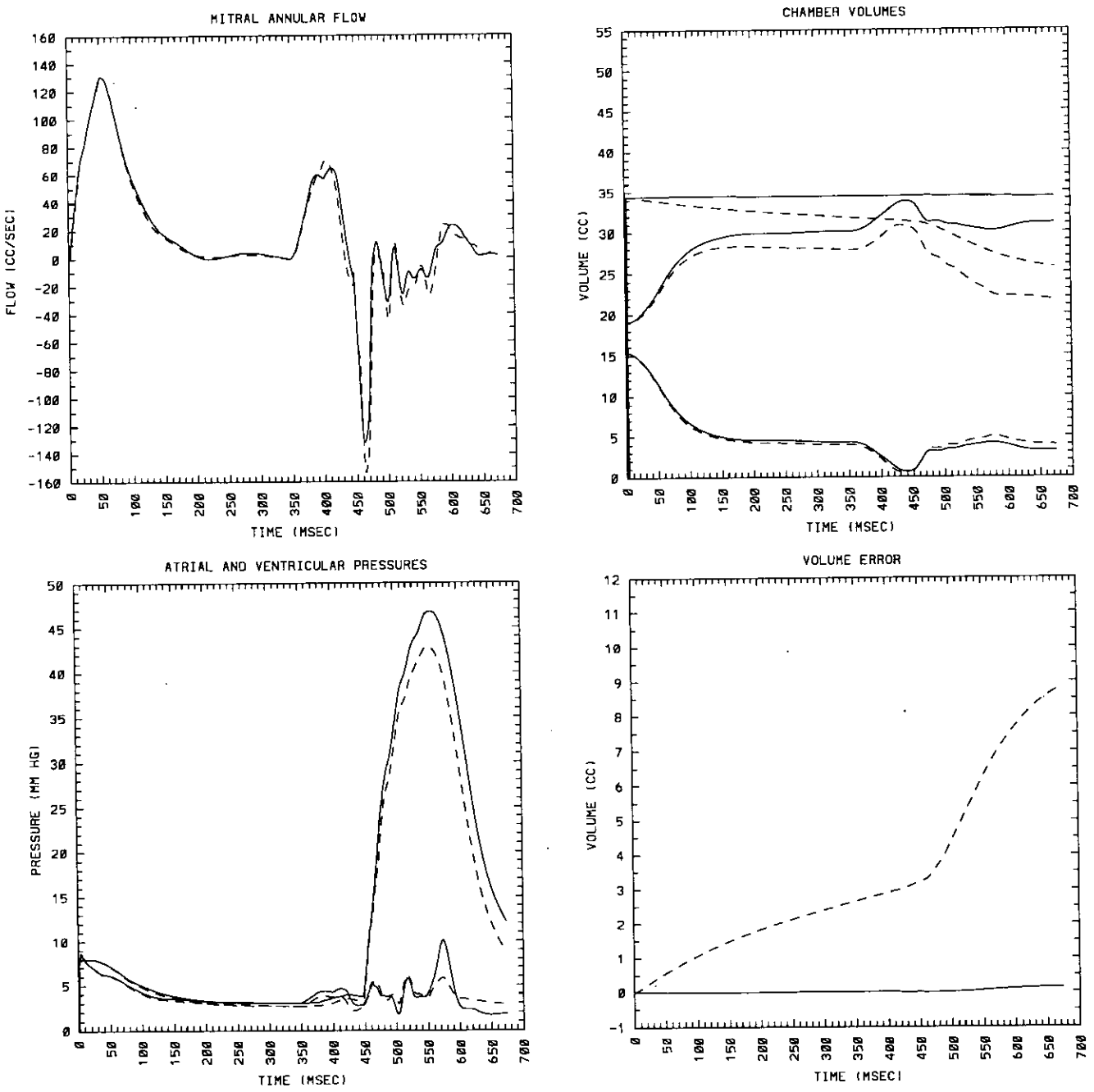

FIG. 2. Flow, pressure, and volume during one cardiac cycle with sources and sinks turned off. Broken curves are computed by the old method and solid curves by the method of this paper. The model includes the left atrium, the left ventricle, and the mitral valve (which guards the opening between the left atrium and the left ventricle). The upper left panel plots the flow (volume/time) through the mitral valve annulus. The early positive wave (0–150 ms) is associated with relaxation of the ventricle, and the later positive wave (350–450 ms) is associated with contraction of the atrium. The large negative deflection represents volume displaced by the closing mitral leaflets, and the oscillations which follow are the source of the first heart sound. The final positive wave represents unloading of stored volume by displacement of the closed valve as the ventricular pressure falls. The lower left panel plots the atrial and ventricular pressures. These run close together with the atrial pressure generally higher than the ventricular during the time interval 0–450 ms when the mitral valve is open. With the contraction of the ventricle at about 450 ms, the ventricular pressure rises dramatically but the atrial pressure is prevented from rising by the closed mitral valve. Oscillations of the valve are reflected in the atrial pressure trace. The upper right panel plots the volumes of the left atrium (lower curve) and left ventricle (middle curve). The uppermost curve is the sum of the other two. Note that this sum should be constant, since the sources and sinks have been turned off. In fact, the broken upper curve, which was obtained by the old method, shows a substantial leak which becomes particularly severe after about 450 ms when the ventricular pressure begins to rise. The solid upper curve, obtained by the new method, is essentially flat. Volume errors are plotted on an expanded scale in the lower right panel. Even here, the volume error of the new method is barely detectable.

RESULTS

Throughout this section, the term *old method* refers to the immersed boundary method with **D** given by central differences $\mathbf{D} = (D_x^0, D_y^0)$, whereas *new method* refers to the same overall algorithm but with **D** constructed from δ_h by the method of this paper (see Eqs. (39)–(40) and Eqs. (35)–(36)). We compare these two methods by applying them to the problem of blood flow in a two-dimensional model of the left side of the heart.

The heart model is shown in Fig. 1. It is described in detail in [6], but a brief description will be given here. The model includes a left atrium, a left ventricle, and a mitral valve (the inflow valve of the left ventricle). The aortic (outflow) valve is also modeled but in a less detailed way, see below.

The entire left heart is modeled as though it were floating in a fluid with the same density and viscosity as the blood in the cardiac chambers. This makes it possible to apply the immersed boundary method not only to the heart valve leaflets (which are actually immersed in blood) but also to the muscular heart walls. We make no effort to uncouple the internal and external fluid dynamics, so the external fluid does have some influence on the internal flow. This is not completely unrealistic, however: The interventricular septum actually has blood in both sides, and the outer walls of the heart are surrounded by tissue, much of which has the same density as blood. In any case, we find that the computed external pressure is relatively uniform, so the results could hardly be very different if we imposed a constant-pressure condition in lieu of the external fluid.

Besides being driven by the contraction and relaxation of the muscular heart walls (see [6] for details), the model is also equipped with an atrial source that models the pulmonary venous return and with an aortic sink located in the ventricular outflow tract that models the function of the aortic valve. (The aortic sink is a new feature of the model, which is otherwise essentially the same as that presented in [6].) A source/sink located outside the heart accommodates the changes in volume that occur during the cardiac cycle. The computational apparatus used to establish sources and sinks is described in [2].

Some parameters of the model are as follows: The diameter of the mitral annulus is (initially) 1.9 cm. The entire computational domain measures 7.36 cm × 7.36 cm. It is covered by a 64×64 mesh, so the mesh width is 0.115 cm. For output purposes, we need to convert *areas* in the two-dimensional model to *volumes* of the cardiac chambers. To do this we multiply by a characteristic length, 1.84 cm, which is 1/4 the length of computational domain. The duration of the cardiac cycle is 0.7 s, of which 0.45 s are occupied by ventricular diastole (relaxation) and 0.25 s by ventricular systole (contraction). The time step is 0.7 ms during ventricular diastole, and this is refined by a factor of 2 or 4 during ventricular systole. The density of blood is taken to be the same as that of water, 1 g/cm^3. The kinematic viscosity of blood, which is actually four times that of water or 0.04 cm^2/s, is artificially raised by a factor of 25 to 1.0 cm^2/s. This increase in viscosity is needed in order to achieve numerical stability without using an excessively fine grid. Evidence that inertial forces remain dominant over viscous forces despite the increased viscosity is presented in [6]. The orders of magnitude of velocity and pressure generated by the model can be gauged as follows:

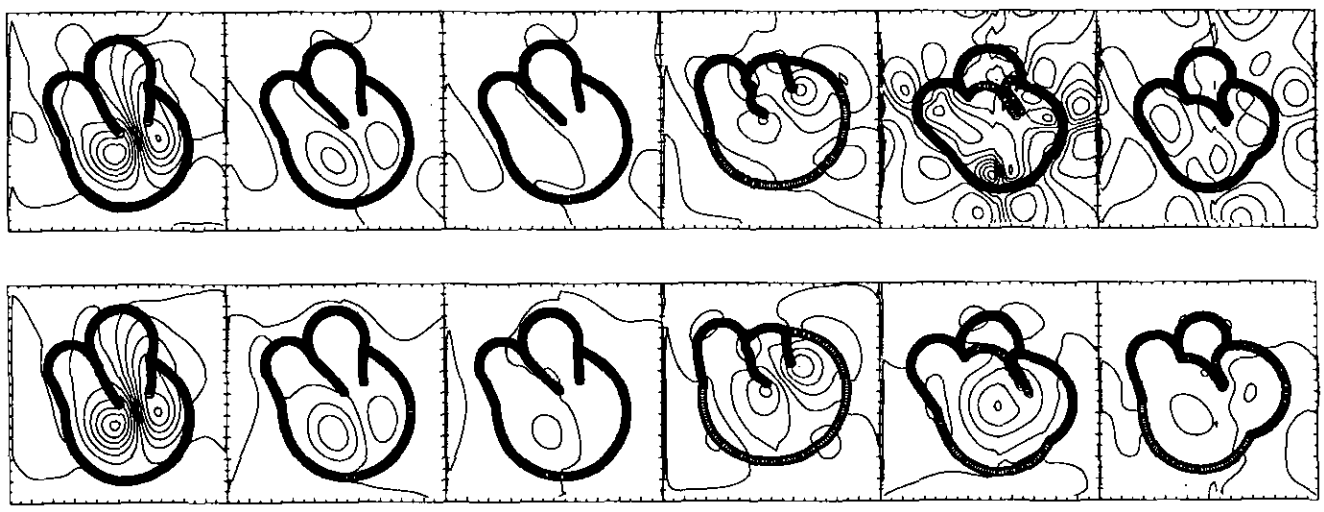

FIG. 3. Computed streamlines corresponding to the wave forms of Fig. 2. All sources and sinks have been turned off. Top row: old method; bottom row: new method. Each row shows selected time steps separated by 112.5 ms. First frame in each row depicts early ventricular diastole, and last frame in each row is at the end of ventricular systole. The heart volume is noticeably larger at the end of the computation performed by the new method (compare last frame, bottom row, with last frame, top row; for quantitative comparison, see Fig. 2). The extra streamlines in the last two frames of the old computation (top row) may be associated with the numerical leakiness of the old method.

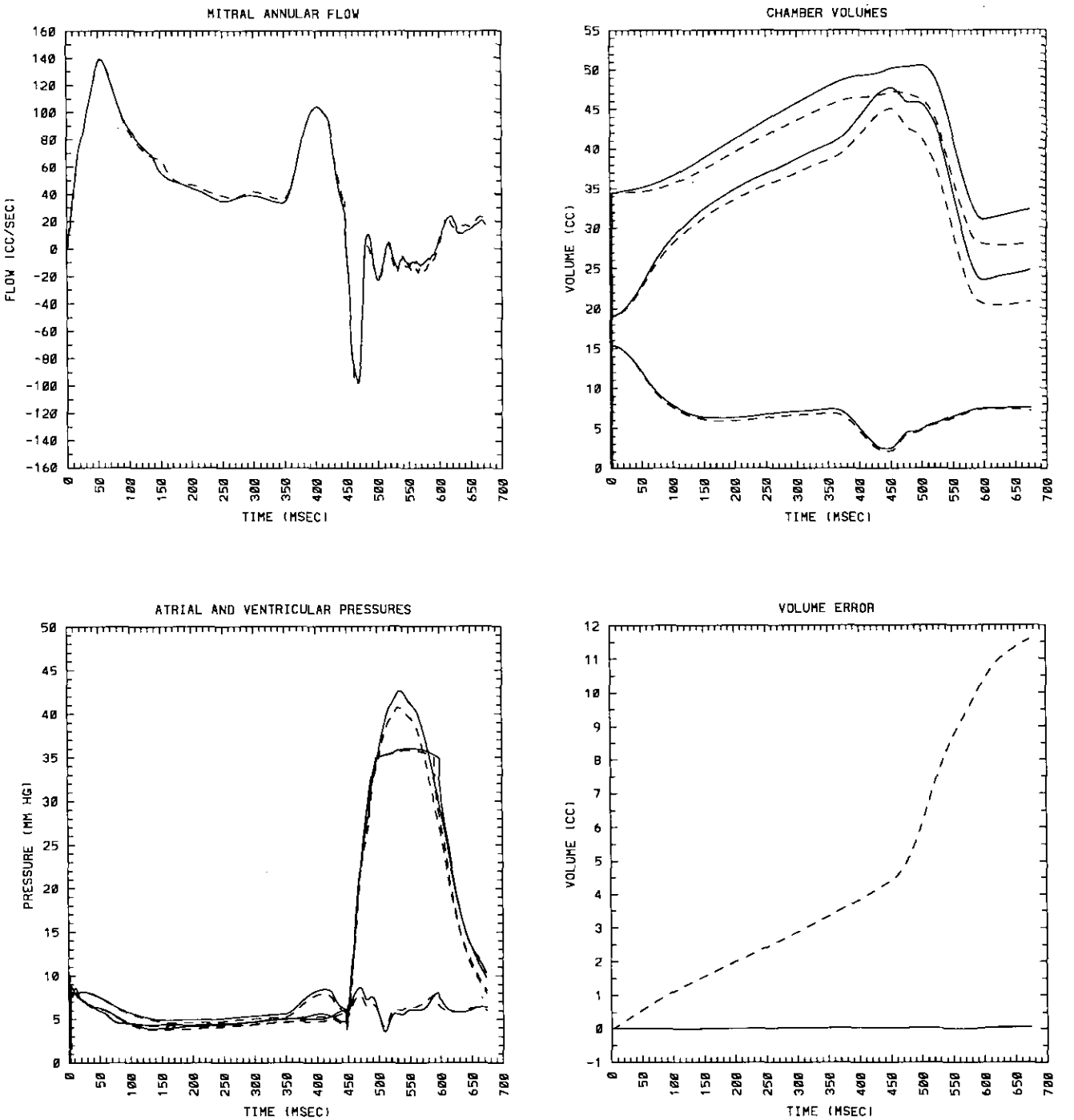

FIG. 4. Flow, pressure, and volume during the cardiac cycle with the sources and sinks turned on. This computational experiment is similar to that of Figs. 2 and 3 except that there is a source in the left atrium and a sink (equipped with a valve) in the left ventricular outflow tract. (A third sink/source outside the heart accommodates the resulting changes in cardiac volume.) All curves have the same meaning as in Fig. 2 (in particular, the solid curves were obtained with the new method and the broken curves with the old) except that there is a third pressure trace in the lower left panel which shows the pressure in the left ventricular outflow tract near the aortic valve. This pressure is more or less "clamped" by the aortic pressure when the aortic valve is open (about 480–600 ms). In this case there is no reason for the combined atrial and ventricular volume to be constant; one would expect it to rise because of pulmonary venous return when the aortic valve is closed and to fall because of ventricular ejection when the aortic valve is open. This is indeed what happens, see uppermost curve in the upper right panel. Nevertheless, one can define a volume error by taking the source and sink flows into account. This has been done in the lower right-hand panel. As in Fig. 2, the volume error of the new method is only just barely detectable, even on the expanded scale of this plot.

The spatially averaged velocity of flow through the mitral valve has a maximum value of about 40 cm/s, and the maximum pressure achieved by the model left ventricle during systole is about 45 mm Hg $= 6 \times 10^4$ dynes/cm^2. (The velocity is realistic, but the pressure is low by a factor of about 2.7.)

Figures 2–5 compare the old and new methods on a heart computation involving one full cardiac cycle. In Figs. 2 and 3 all sources and sinks have been turned off; this makes it straightforward to check on volume conservation because the sum of the chamber volumes should be constant. Figures 4 and 5 show the more realistic case in which the sources and sinks are turned on. Volume conservation can still be checked by allowing for the flow through the sources and sinks.

Figures 2 and 4 compare the old and new methods in terms of the computed waveforms of flow, pressure, and volume. In each of these figures the results obtained with the old method are shown as broken curves whereas the results obtained with the new method are shown as solid curves. Figures 3 and 5 show the selected streamline plots generated by the same computations. In each case, the top row of streamline plots was obtained with the old method and the bottom row with the new method.

Both in Fig. 2 and in Fig. 4, the upper right panel contains graphs of the atrial volume (lowest curve), the ventricular volume (middle curve), and the combined atrial and ventricular volumes (upper curve) as functions of time. In Fig. 2, the combined volume should be constant, since there are no sources or sinks. This is very nearly true for the new method (solid line) but the old method (broken line) shows a substantial leak, especially during ventricular systole ($T > 450$ ms). The volume error is plotted on an expanded scale in the lower right-hand panel of Fig. 2. Even on this expanded scale, the volume error of the new method (solid line) is hard to distinguish from zero. Figure 3 shows the streamline plots of the two computations depicted in Fig. 2. The top row shows results obtained with the old method and the bottom row shows results obtained with the new method. The improvement in volume conservation is evident by the end of the run.

In Fig. 4, there is no reason for the combined atrial and ventricular volume to be constant, since there are sources and sinks as discussed above, but we can still define a volume error by keeping track of the flows through the sources and sinks and comparing the change in combined chamber volume to the net contribution of the sources and sinks. The result of this calculation is shown in the lower right-hand panel of Fig. 4. Again the volume error of the new method (solid line) is only just barely detectable, even on the expanded scale of this plot. Figure 5 shows the streamline plots of the two computations depicted in Fig. 4. The format is the same as in Fig. 3. The loss of volume in the case of the old method (top row) is somewhat less evident here, because the ventricle is decreasing in volume anyway through the normal process of ejection through the aortic valve.

Since the volume errors of the new method are too small

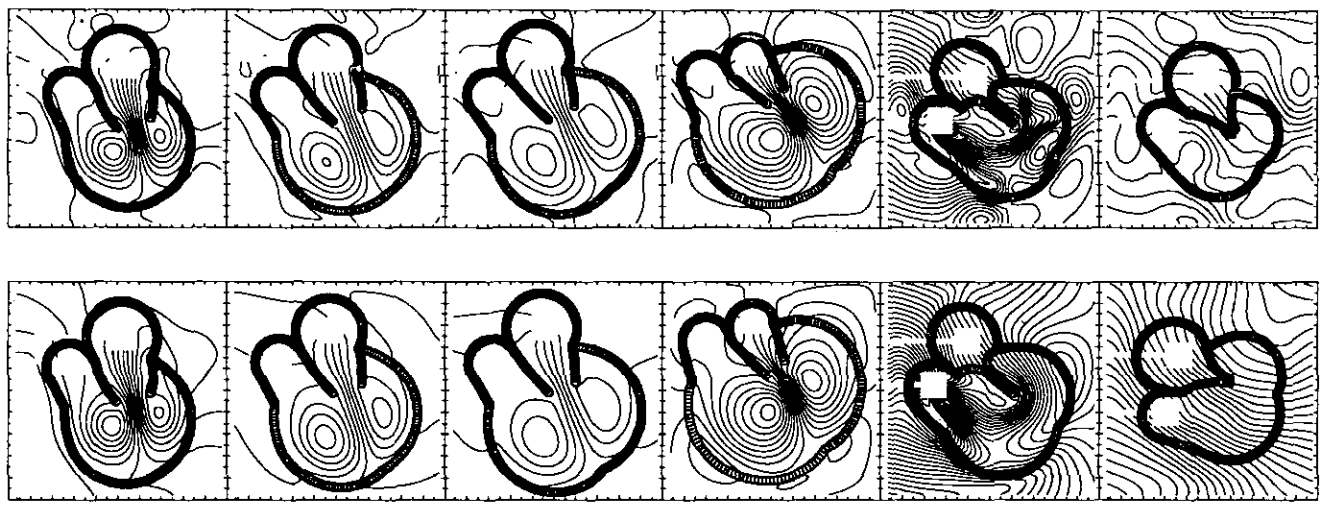

FIG. 5. Computed streamlines corresponding to the waveforms of Fig. 4. Format is the same as in Fig. 3 (old method at top, new method below). Note the source in the left atrium and the sink in the aortic outflow tract. These are most evident in the next to last frame of each row. The stream function is not defined in the region occupied by the source or the sink, since the divergence of the velocity is not zero there. To keep the stream function single valued, two cuts have been introduced, one connecting the atrial source to the external source/sink and the other connecting the aortic sink to the external source/sink. Rotation of the heart during systole appears to be associated with ejection; there is also translation which is not seen in the figure because the plotting program centers the heart in the frame. Unlike the real heart, which is tethered by arteries and veins, the model heart is floating unrestrained. The greater rotation (and translation, not shown) in the computation performed by the new method is probably explained by the greater ejection volume. The results in Fig. 4 show that the decrease in ventricular volume during systole is about the same for the two methods (about 22 cm^3) but that a substantial part of this (8 cm^3) was associated with the numerical leak in the case of the old method and was therefore not ejected through the aortic valve.

TABLE I

Volume Errors over One Cardiac Cycle

Sources and Sinks	Old method	New method	Old/New
Off	8.81 cm^3	0.132 cm^3	67
On	11.58 cm^3	0.0508 cm^3	225

to be read off easily from the figures, we list in Table I the volume errors at the end of each of the two computational experiments depicted in Figs. 2–5. The improvement achieved through the use of the new method can be measured by the ratio of the old volume error to the new volume error. In the two cases reported here, these improvement factors are about 67 and 225.

The left-hand panels in Figs. 2 and 4 show that other physiological variables are not very sensitive to the change in the method that we have made here (and hence that they *are not very sensitive to the volume error of the old* method). The agreement between the old and new results is particularly good in the left-hand panels of Fig. 4, perhaps because the sources and sinks tend to regulate the ventricular volume independent of the numerical leakiness of the old method. Both in Fig. 2 and in Fig. 4, the upper left panel plots the flow through the mitral valve and the lower left panel plots the atrial and ventricular pressures. In Fig. 4, there is an additional pressure trace, the one with a flat top during ventricular systole. It records the pressure in the outflow tract of the left ventricle and is clamped near the aortic pressure while the aortic valve is open.

SUMMARY AND CONCLUSIONS

This paper describes a general procedure that can be used to construct a finite-difference divergence operator corresponding to a given interpolation scheme. The construction is such that the discrete divergence at a mesh point is the same as the average over a box surrounding that mesh point of the continuous divergence of the interpolated velocity field. Thus the interpolation recipe dictates the form of the discrete divergence operator.

The use of a divergence operator constructed in this way requires only a modest change in the immersed-boundary method. Indeed, all of the changes are confined to the projection step of the fluid dynamics subroutine. Since the new divergence operator still has constant coefficients, and since the domain of the computation is periodic, the projection step can be accomplished by Fourier methods in much the same way as before. This modest change in the method yields a dramatic improvement in the results. In the two examples reported here, volume errors are reduced by factors of 67 and 225.

The improvement in methodology described in this paper is particularly important for computations in which an elastic boundary separates chambers which are at substantially different pressures. The situation arises, for example, in the case of cardiac fluid dynamics during the contraction of the ventricles.

APPENDIX: ANALYSIS OF THE VOLUME ERROR

Given a function $\mathbf{u}(\mathbf{x}, h)$ which maps $R^2 \times R^+ \to R^2$, we define another such function $\mathbf{U}(\mathbf{X}, h)$ by the interpolation formula

$$\mathbf{U}(\mathbf{X}, h) = \sum_{x \in Z_h^2} \mathbf{u}(\mathbf{x}, h)\, \delta_h^2(\mathbf{X} - \mathbf{x})\, h^2, \tag{A1}$$

where

$$Z_h^2 = Z_h \times Z_h \tag{A2}$$

$$Z_h = \{x: x = ih, i \text{ an integer}\} \tag{A3}$$

$$\delta_h^2(\mathbf{x}) = \delta_h(x_1)\, \delta_h(x_2), \quad \mathbf{x} = (x_1, x_2), \tag{A4}$$

and where $\delta_h(x)$ is an approximation to the Dirac delta-function.

In this Appendix, we assume that δ_h has the following properties:

$$\delta_h(x) = 0, \quad \text{for} \quad |x| > O(h); \tag{A5}$$

$$\delta_h(x) = O(h^{-1}); \tag{A6}$$

$$\delta_h'(x) = O(h^{-2}); \tag{A7}$$

$$\sum_{x \in Z_h} \delta_h(X - x)h = 1, \quad \text{all} \quad X \in R; \tag{A8}$$

$$\sum_{x \in Z_h} (X - x)\, \delta_h(X - x)h = 0, \quad \text{all} \quad X \in R; \tag{A9}$$

$$\sum_{x \in Z_h} (X - x)^2\, \delta_h(X - x)h = Ch^2, \quad \text{all} \quad X \in R; \tag{A10}$$

where C is independent of X and h. The function δ_h used in the body of the paper satisfies Eqs. (A5)–(A8) exactly but Eqs. (A9)–(A10) only approximately, see Fig. A1.

Differentiating Eqs. (A8)–(A10), we obtain the relations:

$$\sum_{x \in Z_h} \delta_h'(X - x)h = 0, \quad \text{all} \quad X \in R; \tag{A11}$$

$$\sum_{x \in Z_h} (X - x)\, \delta_h'(X - x)h = -1, \quad \text{all} \quad X \in R; \tag{A12}$$

$$\sum_{x \in Z_h} (X - x)^2\, \delta_h'(X - x)h = 0, \quad \text{all} \quad X \in R; \tag{A13}$$

where we have used Eq. (A8) to simplify Eq. (A12), and Eq. (A9) to simplify Eq. (A13).

Let $\langle\ \rangle_{\mathbf{x},h}$ denote the average of the enclosed quantity over a square of side h centered on the point $\mathbf{x}$ and alligned with the coordinate axes. We call this square $B(\mathbf{x}, h)$. Thus

$$B(\mathbf{x}, h) = \{\mathbf{x}': \|\mathbf{x} - \mathbf{x}'\|_{\max} \leqslant h/2\} \tag{A14}$$

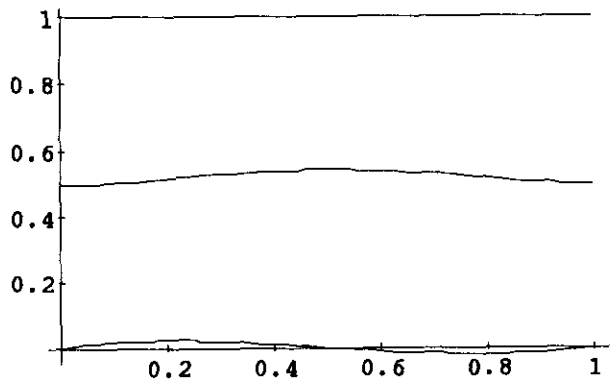

FIG. A1. Some properties of the function δ_h. This Appendix makes certain hypotheses which are only approximately true for the function δ_h used in the body of the paper. Equation (A8) is exactly satisfied (top line) but Eqs. (A.9)–(A.10) are only approximately satisfied (bottom and middle curves, respectively). For each of these equations, the left-hand side is plotted as a function of X/h. In all cases, the result is periodic and one period is shown.

and

$$\langle \phi \rangle_{\mathbf{x},h} = h^{-2} \int_{B(\mathbf{x},h)} \phi(\mathbf{x}')\, d^2\mathbf{x}'. \tag{A15}$$

We impose the following restrictions on the function $\mathbf{u}(\mathbf{x}, h)$ which appears in Eq. (A1). First, $\mathbf{u}(\mathbf{x}, h)$ has an asymptotic expansion of the form

$$\mathbf{u}(\mathbf{x}, h) = \mathbf{u}^{(0)}(\mathbf{x}) + h\mathbf{u}^{(1)}(\mathbf{x}) + h^2\mathbf{u}^{(2)}(\mathbf{x}) + \cdots, \tag{A16}$$

where $\mathbf{u}^{(0)}$, $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, etc., are smooth functions (independent of h). Second, we assume $\mathbf{u}(\mathbf{x}, h)$ is such that

$$\langle \nabla \cdot \mathbf{U} \rangle_{\mathbf{x},h} = 0 \qquad \text{for all} \quad h > 0 \text{ and all } \mathbf{x} \in Z_h^2. \tag{A17}$$

(Recall that $\mathbf{u}$ determines $\mathbf{U}$, see Eq. (A1).)

With these assumptions, we shall prove: (1) that $\nabla \cdot \mathbf{u}^{(0)} = \nabla \cdot \mathbf{u}^{(1)} = 0$, so that $\nabla \cdot \mathbf{u} = O(h^2)$; (2) that $\nabla \cdot \mathbf{U} = O(h^2)$; and (3) that $\int_\Omega \nabla \cdot \mathbf{U}\, d^2\mathbf{X} = O(h^3)$ for any fixed region Ω whose boundary has dimension 1. Note the gain of one power of h in the integral of $\nabla \cdot \mathbf{U}$ as opposed to $\nabla \cdot \mathbf{U}$ itself.

The first step in the proof is to show that $(\nabla \cdot \mathbf{U})(\mathbf{X}) = (\nabla \cdot \mathbf{u})(\mathbf{X}) + O(h^2)$ for all $\mathbf{X} \in R^2$. This is done by Taylor series, and it makes extensive use of the properties of the function δ_h that were postulated above. Let $\mathbf{u}(\mathbf{x}, h) = (u_1(x_1, x_2, h), u_2(x_1, x_2, h))$, and similarly for $\mathbf{U}(\mathbf{X}, h)$. For convenience we shall drop the h argument in this part of the proof. We have

$$\frac{\partial U_1}{\partial X_1}(X_1, X_2) = \sum_{\substack{x_1 \in Z_h \\ x_2 \in Z_h}} u_1(x_1, x_2)\, \delta_h'(X_1 - x_1)\, h \delta_h(X_2 - x_2) h. \tag{A18}$$

But

$$\begin{aligned} u_1(x_1, x_2) = {} & u_1(X_1, X_2) + (x_1 - X_1)\frac{\partial u_1}{\partial x_1}(X_1, X_2) \\ & + (x_2 - X_2)\frac{\partial u_1}{\partial x_2}(X_1, X_2) \\ & + \frac{1}{2}(x_1 - X_1)^2 \frac{\partial^2 u_1}{\partial x_1^2}(X_1, X_2) \\ & + (x_1 - X_1)(x_2 - X_2)\frac{\partial^2 u_1}{\partial x_1\, \partial x_2}(X_1, X_2) \\ & + \frac{1}{2}(x_2 - X_2)^2 \frac{\partial^2 u_1}{\partial x_2^2}(X_1, X_2) + O(h^3), \end{aligned} \tag{A19}$$

where we have assumed that $x_1 - X_1 = O(h)$ and $x_2 - X_2 = O(h)$. These are the only values of x_1 and x_2 that we need to consider because δ_h has support of width $O(h)$, see Eq. (A5). In Eq. (A19), expressions like $\partial u_1/\partial x_1$ are shorthand for $\partial u_1^{(0)}/\partial x_1 + h\, \partial u_1^{(1)}/\partial x_1 + h^2\, \partial u_1^{(2)}/\partial x_1 + \cdots$.

Substituting Eq. (A19) in Eq. (A18), we find

$$\begin{aligned} \frac{\partial U_1}{\partial X_1}(X_1, X_2) = {} & u_1(X_1, X_2) \sum_{x_1 \in Z_h} \delta_h'(X_1 - x_1) h \sum_{x_2 \in Z_h} \delta_h(X_2 - x_2) h \\ & + \frac{\partial u_1}{\partial x_1}(X_1, X_2) \sum_{x_1 \in Z_h} (x_1 - X_1) \\ & \times \delta_h'(X_1 - x_1) h \sum_{x_2 \in Z_h} \delta_h(X_2 - x_2) h \\ & + \frac{\partial u_1}{\partial x_2}(X_1, X_2) \sum_{x_1 \in Z_h} \delta_h'(X_1 - x_1) h \\ & \times \sum_{x_2 \in Z_h} (x_2 - X_2)\, \delta_h(X_2 - x_2) h \\ & + \frac{1}{2}\frac{\partial^2 u_1}{\partial x_1^2}(X_1, X_2) \sum_{x_1 \in Z_h} (x_1 - X_1)^2 \\ & \times \delta_h'(X_1 - x_1) h \sum_{x_2 \in Z_h} \delta_h(X_2 - x_2) h \\ & + \frac{\partial^2 u_1}{\partial x_1\, \partial x_2}(X_1, X_2) \sum_{x_1 \in Z_h} (x_1 - X_1) \\ & \times \delta_h'(X_1 - x_1) h \sum_{x_2 \in Z_h} (x_2 - X_2)\, \delta_h(X_2 - x_2) h \\ & + \frac{1}{2}\frac{\partial^2 u_1}{\partial x_2^2}(X_1, X_2) \sum_{x_1 \in Z_h} \delta_h'(X_1 - x_1) h \\ & \times \sum_{x_2 \in Z_h} (x_2 - X_2)^2\, \delta_h(X_2 - x_2) h \\ & + O(h^3) \cdot O(h^{-2}) \cdot O(h^{-1}) \cdot O(h^2). \end{aligned} \tag{A20}$$

Making use of the δ_h-function identities, Eqs. (A8)–(A13), we see that Eq. (A20) simplifies as follows:

$$\frac{\partial U_1}{\partial X_1}(X_1, X_2) = \frac{\partial u_1}{\partial x_1}(X_1, X_2) + O(h^2). \tag{A21}$$

Similarly

$$\frac{\partial U_2}{\partial X_2}(X_1, X_2) = \frac{\partial u_2}{\partial x_2}(X_1, X_2) + O(h^2). \tag{A22}$$

It follows that

$$(\nabla \cdot \mathbf{U})(\mathbf{X}) = (\nabla \cdot \mathbf{u})(\mathbf{X}) + O(h^2), \qquad \text{all} \quad \mathbf{X} \in R^2 \tag{A23}$$

as claimed above.

Next we average Eq. (A23) over each of the boxes $B(\mathbf{x}, h)$, $\mathbf{x} \in Z_h^2$. The result is

$$\langle \nabla \cdot \mathbf{U} \rangle_{\mathbf{x},h} = \langle \nabla \cdot \mathbf{u} \rangle_{\mathbf{x},h} + O(h^2), \quad \mathbf{x} \in Z_h^2. \tag{A24}$$

Now recall the asymptotic expansion of $\mathbf{u}$ (Eq. (A16)) and the fact that $\langle \nabla \cdot \mathbf{U} \rangle_{\mathbf{x},h} = 0$. Then Eq. (A24) becomes

$$0 = \langle \nabla \cdot \mathbf{u}^{(0)} \rangle_{\mathbf{x},h} + h \langle \nabla \cdot \mathbf{u}^{(1)} \rangle_{\mathbf{x},h} + O(h^2). \tag{A25}$$

Since $\nabla \cdot \mathbf{u}^{(0)}$ and $\nabla \cdot \mathbf{u}^{(1)}$ are smooth functions, we have

$$\langle \nabla \cdot \mathbf{u}^{(0)} \rangle_{\mathbf{x},h} = (\nabla \cdot \mathbf{u}^{(0)})(\mathbf{x}) + O(h^2) \tag{A26}$$

$$\langle \nabla \cdot \mathbf{u}^{(1)} \rangle_{\mathbf{x},h} = (\nabla \cdot \mathbf{u}^{(1)})(\mathbf{x}) + O(h^2). \tag{A27}$$

Hence

$$0 = \nabla \cdot \mathbf{u}^{(0)} + h \nabla \cdot \mathbf{u}^{(1)} + O(h^2). \tag{A28}$$

Since this holds for all $h > 0$, we conclude that

$$\nabla \cdot \mathbf{u}^{(0)} = 0 \tag{A29}$$

$$\nabla \cdot \mathbf{u}^{(1)} = 0 \tag{A30}$$

and hence that

$$\nabla \cdot \mathbf{u} = O(h^2). \tag{A31}$$

From Eq. (A23), it then follows that

$$\nabla \cdot \mathbf{U} = O(h^2). \tag{A32}$$

It remains only to consider $\int_\Omega \nabla \cdot \mathbf{U}\, d^2\mathbf{X}$. (The argument that we use here was suggested by R. J. LeVeque.) Let

$$I_h = \{\mathbf{x} : \mathbf{x} \in Z_h^2 \text{ and } B(\mathbf{x}, h) \subset \Omega\} \tag{A33}$$

$$\Omega_h = \bigcup_{\mathbf{x} \in I_h} B(\mathbf{x}, h) \tag{A34}$$

$$\partial\Omega_h = \Omega - \Omega_h. \tag{A35}$$

Thus Ω_h consists of all *interior* boxes of the form $B(\mathbf{x}, h)$ where $\mathbf{x} \in Z_h^2$, and $\partial\Omega_h$ is the part of Ω that is not in Ω_h. Since Ω is a fixed region independent of h, and since the boundary of Ω has dimension 1, we have

$$\text{area}(\partial\Omega_h) = O(h). \tag{A36}$$

Now

$$\begin{aligned} \int_\Omega (\nabla \cdot \mathbf{U})\, d^2\mathbf{X} &= \int_{\Omega_h} (\nabla \cdot \mathbf{U})\, d^2\mathbf{X} + \int_{\Omega - \Omega_h} (\nabla \cdot \mathbf{U})\, d^2\mathbf{X} \\ &= \sum_{\mathbf{x} \in I_h} h^2 \langle \nabla \cdot \mathbf{U} \rangle_{\mathbf{x},h} + \int_{\Omega - \Omega_h} O(h^2)\, d^2\mathbf{X} \\ &= 0 + O(h^2) \cdot O(h) = O(h^3). \end{aligned} \tag{A37}$$

This completes the proof of all claims.

While this Appendix gives some insight into the utility of imposing the condition $\langle \nabla \cdot \mathbf{U} \rangle_{\mathbf{x},h} = 0$, the theory developed here cannot be directly applied to the computational results reported in the body of the paper. Several of the assumptions made here are too strong to be satisfied in practice. It has already been mentioned that the function δ_h used in our computations does not satisfy all of the conditions laid down here, and it is also true that the smoothness assumptions made here are unrealistic, since some velocity derivatives jump as one crosses an immersed boundary. Thus, further theoretical work is needed to estimate the volume errors that are to be found in practice.

ACKNOWLEDGMENTS

The authors are indebted to D. M. McQueen and R. J. LeVeque for helpful discussions during the course of this work. In particular, the concluding argument of the Appendix is due to LeVeque. The research described here was supported by the National Institutes of Health under research Grant *HL17859, Computational Methods in Cardiac Fluid Dynamics*. B.F.P. is also supported by the Medical Scientist Training Program at the Mt. Sinai School of Medicine. Computation was performed at the Minnesota Supercomputer Center and at the Pittsburgh Supercomputing Center.

REFERENCES

1. C. S. Peskin, *Flow Patterns Around Heart Valves: A Digital Computer Method for Solving the Equations of Motion*, Thesis (Physiology), Albert Einstein College of Medicine, July, 1972. (University Microfilms #72-30, 378.211 pp.)

2. C. S. Peskin, *J. Comput. Phys.* **25**, 220 (1977).
3. C. S. Peskin and D. M. McQueen, *J. Comput. Phys.* **37**, 113 (1980).
4. C. S. Peskin and D. M. McQueen, *J. Comput. Phys.* **81**, 372 (1989).
5. D. M. McQueen and C. S. Peskin, *J. Comput. Phys.* **82**, 289 (1989).
6. D. M. McQueen, C. S. Peskin, and E. L. Yellin, *Am. J. Physiol.* **242**, H1095 (1982).
7. J. S. Meisner, D. M. McQueen, Y. Ishida, H. O. Vetter, U. Bortolotti, J. A. Strom, R. W. M. Frater, C. S. Peskin, and E. L. Yellin, *Am. J. Physiol.* **249**, H604 (1985).
8. D. M. McQueen and C. S. Peskin, *J. Thorac. Cardiovasc. Surg.* **86**, 126 (1983).
9. D. M. McQueen and C. S. Peskin, *Scand. J. Thorac. Cardiovasc. Surg.* **19**, 139 (1985).
10. A. L. Fogelson, *J. Comput. Phys.* **56**, 111 (1984).
11. L. J. Fauci and C. S. Peskin, *J. Comput. Phys.* **77**, 85 (1988).
12. R. P. Beyer, *J. Comput. Phys.* **98**, 145 (1992).
13. A. L. Fogelson and C. S. Peskin, *J. Comput. Phys.* **79**, 50 (1988).
14. C. Tu and C. S. Peskin, *SIAM J. Sci. Statist. Comput.*, in press.
15. A. J. Chorin, *Math. Comp.* **22**, 745 (1968).
16. A. J. Chorin, *Math. Comp.* **23**, 341 (1969).