89N24111

NASA Technical Memorandum 102183

Numerical Simulation of Flow Through an Artificial Heart

Stuart E. Rogers, Sterling Federal Systems, Palo Alto, California
Paul Kutler and Dochan Kwak, Ames Research Center, Moffett Field, California
Cetin Kiris, Stanford University, Stanford, California

April 1989

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

SUMMARY

A solution procedure has been developed that solves the unsteady, incompressible Navier-Stokes equations, and has been used to numerically simulate viscous incompressible flow through a model of the Pennsylvania State artificial heart. The solution algorithm is based on the artificial compressibility method, and uses flux-difference splitting to upwind the convective terms; a line-relaxation scheme is used to solve the equations. The time-accuracy of the method is obtained by iteratively solving the equations at each physical time step. The artificial heart geometry involves a piston-type action with a moving solid wall. A single H-grid is fit inside the heart chamber. The grid is continuously compressed and expanded with a constant number of grid points to accommodate the moving piston. The computational domain ends at the valve openings where nonreflective boundary conditions based on the method of characteristics are applied. Although a number of simplifing assumptions were made regarding the geometry, the computational results agreed reasonably well with an experimental picture. The computer time requirements for this flow simulation, however, are quite extensive. Computational study of this type of geometry would benefit greatly from improvements in computer hardware speed and algorithm efficiency enhancements.

This work is part of a joint effort with Pennsylvania State University and Stanford University and is partially funded by the NASA Technology Utilization office.

INTRODUCTION

Advances in the field of Computational Fluid Dynamics (CFD) along with advances in supercomputer hardware have made it possible to simulate more complicated problems than ever before. The current work is an example of how this may affect more than just the field of aerodynamics. The goal of this work is to apply a newly developed solution algorithm to the problem of flow through a model of an artificial heart as a demonstration calculation to show the capability of CFD for computing a variety of problems in fluid dynamics. Developing this capability will extend the use of CFD as a design tool into all aspects of fluid dynamic mechanisms. The analysis of blood flow through the heart, blood vessels, and various biomedical prosthetic devices requires detailed knowledge of the flow quantities. Blood may exhibit significant non-Newtonian characteristics locally and the geometry is usually very complicated. Also, the flow is unsteady, possibly periodic, very viscous, and incompressible. In an artificial organ, blood may go through regions of high turbulent stress which may damage the red blood cells. The problems are very much interdisciplinary and an attempt for a complete simulation would be a formidable task. However, an analysis based on a simplified model may provide much needed physical insight into the blood flow analysis. For a more comprehensive study on blood flow, see references 1–5.

Mechanical hearts are in demand as a temporary life support system. Presently, these devices have several problems, many of which are directly attributable to the fluid dynamics of the blood flow. It therefore is of considerable benefit to medical researchers to determine the flow characteristics in these devices by applying state-of-the-art CFD technology. Some ongoing work by the authors has involved the development of viscous, incompressible flow solvers. This research is motivated by needs for realistic three-dimensional simulations of aerospace applications, such as the flow through the Space Shuttle main engine power head.

The current formulation is based on a Newtonian fluid assumption. However, since the governing equations are solved in a generalized coordinate system, viscosity that varies in space and time is allowed; a full simulation of viscoelastic flow is very difficult because of nonlinearities of the fluid. However, as a first step toward full simulations, non-Newtonian effects of the blood flow can be simplified by a constitutive model for viscous stresses. In addition, the code formulation allows the implementation of a moving geometry. Thus, these flow solvers can be applied to analyze mechanical hearts and ventricular assist devices. The primary purpose of the current work is to transfer NASA-developed technology to artificial heart researchers. NASA benefits by advancing the state of the art in CFD for treating unsteady internal flow with moving boundaries, while the artificial heart manufacturers benefit by gaining a better understanding of the fluid flow within their devices and, hopefully, an improved design.

This paper discusses the application of an incompressible Navier-Stokes flow solver recently developed by the authors to the problem of the unsteady flow through a model of the Pennsylvania State (PS) artificial heart (ref. 4). The first section gives the details of the flow solver. Next, the geometry of the PS artificial heart is presented and the grid generation is discussed. The final section details the solution and shows some of the flow physics in the heart.

COMPUTATIONAL SOLUTION METHOD

Recent developments in the numerical solution of the incompressible Navier-Stokes equations include an efficient algorithm for time-dependent flows (refs. 6 and 7). This algorithm uses a flux-difference-split, upwind-differencing scheme for the convective fluxes. The resulting system of numerical equations is more nearly diagonally dominant than that resulting from the use of a central-difference scheme. The system is solved using an unfactored line-relaxation scheme which proves to have very good stability and convergence characteristics. Some of the details of this algorithm are described in this section.

The algorithm requires that the equations be cast in primative variable form, using pressure and the velocity components as the dependent variables. First the equations are transformed into generalized curvilinear coordinates, resulting in the following form for the continuity and momentum equations, respectively

$$\begin{aligned}
&\frac{\partial}{\partial\xi}\left(\frac{U}{J}\right) + \frac{\partial}{\partial\eta}\left(\frac{V}{J}\right) + \frac{\partial}{\partial\zeta}\left(\frac{W}{J}\right) = 0 \\
&\frac{\partial\hat{u}}{\partial t} = -\frac{\partial}{\partial\xi}(\hat{e} - \hat{e}_v) - \frac{\partial}{\partial\eta}(\hat{f} - \hat{f}_v) - \frac{\partial}{\partial\zeta}(\hat{g} - \hat{g}_v) = -\hat{r}
\end{aligned} \tag{1}$$

where $\hat{r}$ represents the right-hand side of the momentum equations. The equations have been written in generalized coordinates using

$$\begin{aligned}
\xi &= \xi(x, y, z, t) \\
\eta &= \eta(x, y, z, t) \\
\zeta &= \zeta(x, y, z, t)
\end{aligned} \tag{2}$$

where J is the Jacobian of the transformation and

$$\begin{aligned}
\hat{u} &= \frac{1}{J}\begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
\hat{e} &= \frac{1}{J}\begin{bmatrix} \xi_x p + uU + \xi_t u \\ \xi_y p + vU + \xi_t v \\ \xi_z p + wU + \xi_t w \end{bmatrix} \\
\hat{f} &= \frac{1}{J}\begin{bmatrix} \eta_x p + uV + \eta_t u \\ \eta_y p + vV + \eta_t v \\ \eta_z p + wV + \eta_t w \end{bmatrix} \\
\hat{g} &= \frac{1}{J}\begin{bmatrix} \zeta_x p + uW + \zeta_t u \\ \zeta_y p + vW + \zeta_t v \\ \zeta_z p + wW + \zeta_t w \end{bmatrix} \\
U &= \xi_x u + \xi_y v + \xi_z w \\
V &= \eta_x u + \eta_y v + \eta_z w \\
W &= \zeta_x u + \zeta_y v + \zeta_z w
\end{aligned} \tag{3}$$

The metrics of the transformation have been represented by

$$\frac{\partial\xi}{\partial x} = \xi_x, \text{ etc.}$$

In deriving the viscous fluxes, constant viscosity was assumed for simplicity and because calculations of a laminar, Newtonian fluid are being performed initially. This simplification is not necessary and will be removed in subsequent calculations where a non-Newtonian fluid will be necessary to model the flow of blood. Thus the viscous fluxes are then given by

$$
\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\xi\cdot\nabla\xi)u_\xi & +(\nabla\xi\cdot\nabla\eta)u_\eta & +(\nabla\xi\cdot\nabla\zeta)u_\zeta \\ (\nabla\xi\cdot\nabla\xi)v_\xi & +(\nabla\xi\cdot\nabla\eta)v_\eta & +(\nabla\xi\cdot\nabla\zeta)v_\zeta \\ (\nabla\xi\cdot\nabla\xi)w_\xi & +(\nabla\xi\cdot\nabla\eta)w_\eta & +(\nabla\xi\cdot\nabla\zeta)w_\zeta \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\eta\cdot\nabla\xi)u_\xi & +(\nabla\eta\cdot\nabla\eta)u_\eta & +(\nabla\eta\cdot\nabla\zeta)u_\zeta \\ (\nabla\eta\cdot\nabla\xi)v_\xi & +(\nabla\eta\cdot\nabla\eta)v_\eta & +(\nabla\eta\cdot\nabla\zeta)v_\zeta \\ (\nabla\eta\cdot\nabla\xi)w_\xi & +(\nabla\eta\cdot\nabla\eta)w_\eta & +(\nabla\eta\cdot\nabla\zeta)w_\zeta \end{bmatrix} \qquad (4) \\
\hat{g}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\zeta\cdot\nabla\xi)u_\xi & +(\nabla\zeta\cdot\nabla\eta)u_\eta & +(\nabla\zeta\cdot\nabla\zeta)u_\zeta \\ (\nabla\zeta\cdot\nabla\xi)v_\xi & +(\nabla\zeta\cdot\nabla\eta)v_\eta & +(\nabla\zeta\cdot\nabla\zeta)v_\zeta \\ (\nabla\zeta\cdot\nabla\xi)w_\xi & +(\nabla\zeta\cdot\nabla\eta)w_\eta & +(\nabla\zeta\cdot\nabla\zeta)w_\zeta \end{bmatrix}
\end{aligned}
$$

where ν is the kinematic viscosity and the velocity gradients in the viscous fluxes were written as

$$\frac{\partial u}{\partial \xi} = u_\xi, \text{ etc.}$$

In the time-accurate formulation the time derivatives in the momentum equations are differenced using a second-order, three-point, backward-difference formula

$$\frac{1.5\,\hat{u}^{n+1} - 2\,\hat{u}^{n} + 0.5\,\hat{u}^{n-1}}{\Delta t} = -\hat{r}^{n+1} \qquad (5)$$

where the superscript n denotes the quantities at time $t = n\Delta t$ and $\hat{r}$ is the residual given in equation (1). To solve equation (5) for a divergence-free velocity at the $n+1$ time level, a pseudo-time level is introduced denoted by a superscript m. The equations are iteratively solved such that $\hat{u}^{n+1,m+1}$ approaches the new velocity $\hat{u}^{n+1}$ as the divergence of $\hat{u}^{n+1,m+1}$ approaches zero. To drive the divergence of this velocity to zero, the following artificial compressibility relation is introduced:

$$\frac{\partial p}{\partial \tau} = -\beta \nabla \cdot \hat{u}^{n+1,m+1} \qquad (6)$$

where τ denotes pseudo-time and where β is the artificial compressibility parameter, which is to be specified by the user. Applying an implicit Euler time differencing to equation (6) and combining it with the momentum equations gives

$$
\begin{aligned}
& I_{t\tau}(\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}) \\
&= -\hat{R}^{n+1,m+1} - \frac{I_m}{\Delta t}(1.5\,\hat{D}^{n+1,m} - 2\,\hat{D}^{n} + 0.5\,\hat{D}^{n-1})
\end{aligned} \qquad (7)
$$

where

$$\begin{aligned}
\hat{D} &= \frac{1}{J}\begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix} \\
\hat{E} &= \begin{bmatrix} \beta U/J \\ \hat{e} \end{bmatrix} \\
\hat{F} &= \begin{bmatrix} \beta V/J \\ \hat{f} \end{bmatrix} \\
\hat{G} &= \begin{bmatrix} \beta W/J \\ \hat{g} \end{bmatrix} \\
\hat{E}_v &= \begin{bmatrix} 0 \\ \hat{e}_v \end{bmatrix} \\
\hat{F}_v &= \begin{bmatrix} 0 \\ \hat{f}_v \end{bmatrix} \\
\hat{G}_v &= \begin{bmatrix} 0 \\ \hat{g}_v \end{bmatrix}
\end{aligned} \tag{8}$$

and where $I_{t\tau}$ is a diagonal matrix and I_m is a modified identity matrix given by

$$\begin{aligned}
I_{t\tau} &= \operatorname{diag}\left[\frac{1}{\Delta\tau}, \frac{1.5}{\Delta t}, \frac{1.5}{\Delta t}, \frac{1.5}{\Delta t}\right] \\
I_m &= \operatorname{diag}[0, 1, 1, 1]
\end{aligned}$$

Finally, the residual term at the $m+1$ pseudo-time level is linearized, giving the following equation in delta form

$$\begin{aligned}
&\left[\frac{I_{t\tau}}{J} + \left(\frac{\partial \hat{R}}{\partial D}\right)^{n+1,m}\right](D^{n+1,m+1} - D^{n+1,m}) \\
&= -\hat{R}^{n+1,m} - \frac{I_m}{J\Delta t}(1.5D^{n+1,m} - 2D^n + 0.5D^{n-1})
\end{aligned} \tag{9}$$

where $D = J\hat{D}$.

The viscous fluxes in the residual vector $\hat{R}$ are formed using second-order central differencing. The convective terms are computed using a flux-difference split form of upwind differencing with a choice of either third- or fifth-order accuracy. All of the results reported in this paper used fifth-order fluxes. The use of upwind differencing makes the equations diagonally dominant, which will lead to fast convergence in the pseudo-time iterations. Approximate Jacobians of the residual were used to form the left-hand side of the equations. To numerically solve the resulting system, the entire numerical approximation to equation (9) was formed at every grid point. A line-relaxation scheme was then used in which one sweep direction was chosen initially. Then the terms on the left-hand side which did not fall on the sweep direction line were shifted over to the right-hand side, resulting in a tridiagonal system of equations. This system was then solved iteratively, using five iterations and one sweep direction. The boundary conditions used with this flow solver are implicit and are based on the method of characteristics (refs. 7 and 8). These have proven to be very robust and non-reflective. For more details on the flow solver, see Rogers and Kwak (refs. 6 and 7).

GEOMETRY AND GRID GENERATION

The actual model of the PS artificial heart poses some very difficult problems from a computational standpoint. A computer-generated, shaded-surface representation of the heart is shown in figure 1. The heart is composed of a cylindrical chamber with two openings on the side for valves. The pumping action is provided by a piston surface which moves up and down inside the chamber. The diameter of the piston is 7.4 cm, with a stroke length of 2.54 cm. The problem was nondimensionalized with a characteristic length of 2.54 cm and a characteristic velocity of 40 cm/sec. The actual heart has a cylindrical tube extending out of each of the valve openings. These tubes contain tilting flat disks which act as the valves. The current computational model will neglect the valves altogether and will use the right and left openings shown in figure 1 for the inflow and outflow boundaries, respectively. In the computations, as the piston reaches its topmost position, the outflow valve closes and the inflow valve will open instantaneously. Similarly, as the piston reaches its bottommost position, the outflow valve will open and the inflow valve will close.

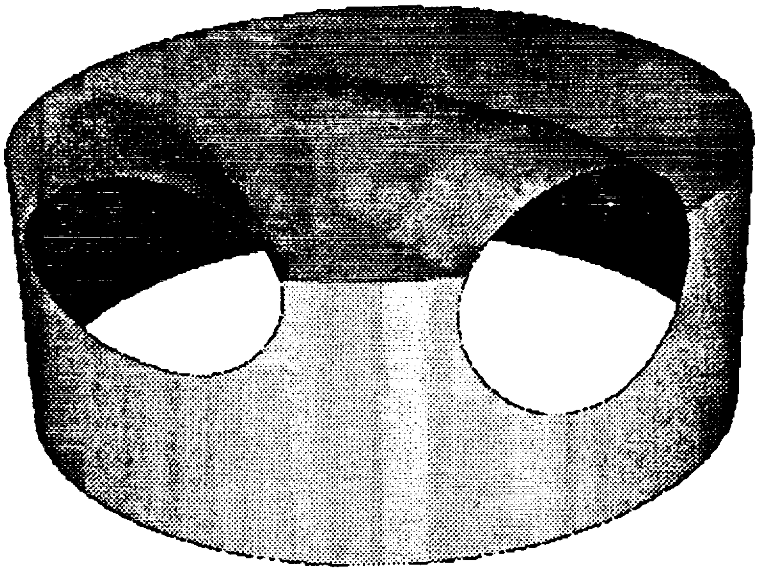
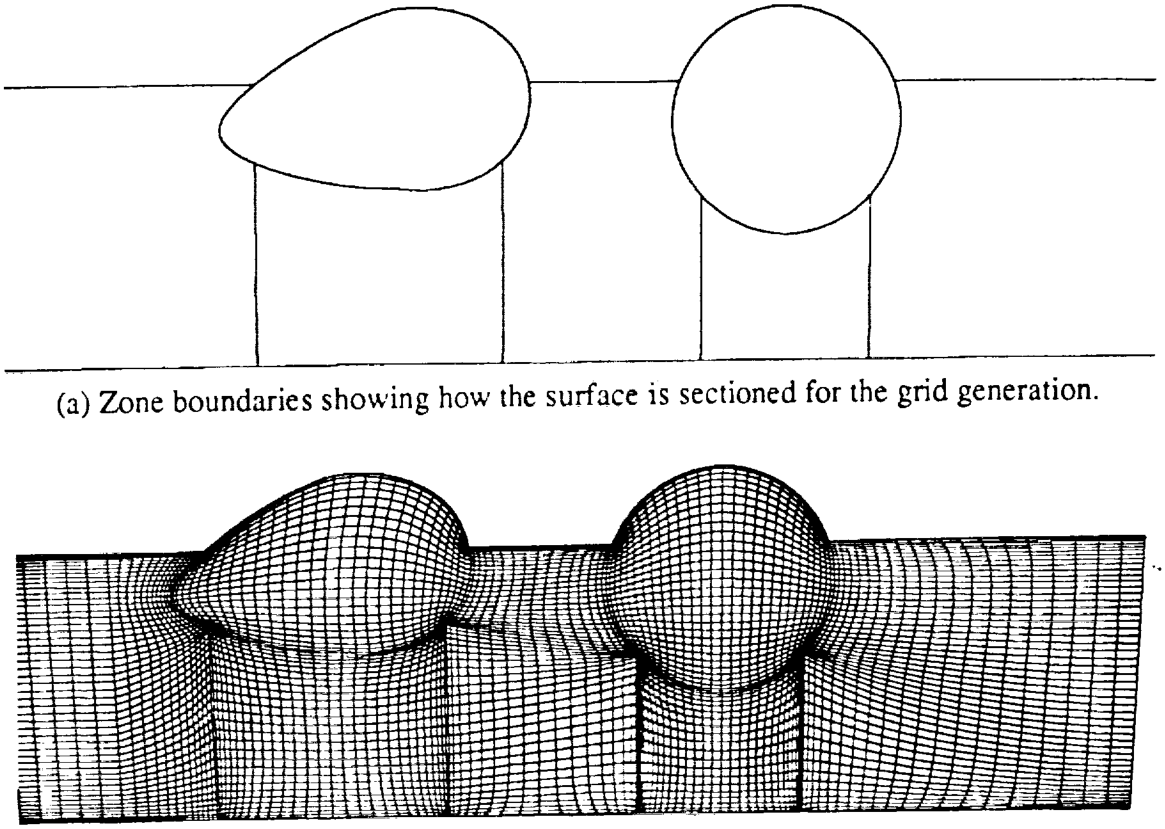


Figure 1.– Artificial heart geometry showing valve openings.

In the actual heart device, the piston moves through the entire chamber volume, including across most of the valve opening. This will cause some severe problems for the computational grid as the piston moves across the valve boundaries. Since the flow solver is designed to use body-fitted coordinates, it is necessary to place the grid lines around the valve to coincide with the valve opening boundaries. Yet since the piston moves past this opening, the grid will have to accommodate both of these surfaces. There are several ways in which a computational grid can handle this motion. One method is with the use of a Chimera scheme (ref. 9) in which two grids are used, one which moves with the piston, and one which is attached to the rest of the body. Information is passed between the two grids by interpolating variables at the grid interfaces. This method can be expensive and complicated to implement. Two methods which are somewhat simpler to implement both involve the use of one grid inside the computational domain. One of these involves a stationary grid through which the piston surface travels. Boundary conditions applied at the piston surface would ensure that mass and momentum are conserved for the partial grid cells as the piston moves through them. However, with the curved surfaces at the valve boundaries, this would involve partial grid cells with an arbitrary number of sides, making this formulation difficult. Another single-grid method uses a grid which varies in time as the piston moves through the boundary. The generalized coordinate transformation given by equation (2) enables the grid motion to be accounted for in the equations with the time-varying metrics. This scheme requires that a separate grid be generated for each discrete position of the piston during the calculations. This may provide the simplest way to accommodate the entire piston motion, yet it still poses a difficult grid-generation problem.

In an effort to eliminate this problem, a simpler piston motion was chosen for the calculations of the present study. The piston motion was restricted so that the piston does not rise above the bottom of the valve openings. The latter of the single-grid approaches was used, so that a constant number of grid points are used and the grid between the piston and the bottom of the valve openings compresses and expands as the piston moves up and down. In order to do this the piston was allowed to travel farther down so that the overall volume of the chamber in the computations was larger than in the actual device. One possible drawback to this approach is that the continuous compression and expansion of the grid will have a varying effect on the accuracy of the flow simulation because the grid density will be changing with time. The effect of this has yet to be studied in detail.

To make the most efficient use of grid points, an H-H grid topology was used to fit the grid to the physical domain. The grid dimensions were chosen to be 39 x 39 x 51 because of memory and computational time limitations of the current flow solver. In order to generate a grid at each time step for this geometry, the surface grid was first generated, and from that an algebraic grid generator and elliptic smoother were used to generate the interior points using the distribution given on the surface grid. To generate the surface grid, the side boundary was divided into seven different zones. The points were distributed along each of the zonal boundaries, and then a biharmonic solver written by Bjorstad (ref. 10) was used to generate the grid interior to each of the zones. The biharmonic solver was also used to generate an H-grid for the the top and bottom surfaces of the heart device. This approach made it relatively simple to repeat the process at each time step for any given position of the piston surface. Figure 2 shows the unwrapped surface of the side of the heart chamber. In figure 2a, the zonal boundaries of the different sections used to generate the surface grid are shown, and in figure 2b, the resulting surface grid is shown.



(a) Zone boundaries showing how the surface is sectioned for the grid generation.

(b) Completed surface grid.

Figure 2.– Unwrapped side surface of the artificial heart.

COMPUTED RESULTS

The calculations were carried out on a Cray 2 supercomputer with a core memory of 256 million words. This large amount of memory has proven to very useful because the implementation of the current implicit scheme requires about 180 times the number of grid points in words of memory. The current modest grid of 77,571 points requires over 14 million words of core memory. This large memory use is due to the fact that the line-relaxation scheme was coded in such a way as to save computational time by storing terms and utilizing more memory whenever possible. Other formulations using less memory are possible and will be coded in the future.

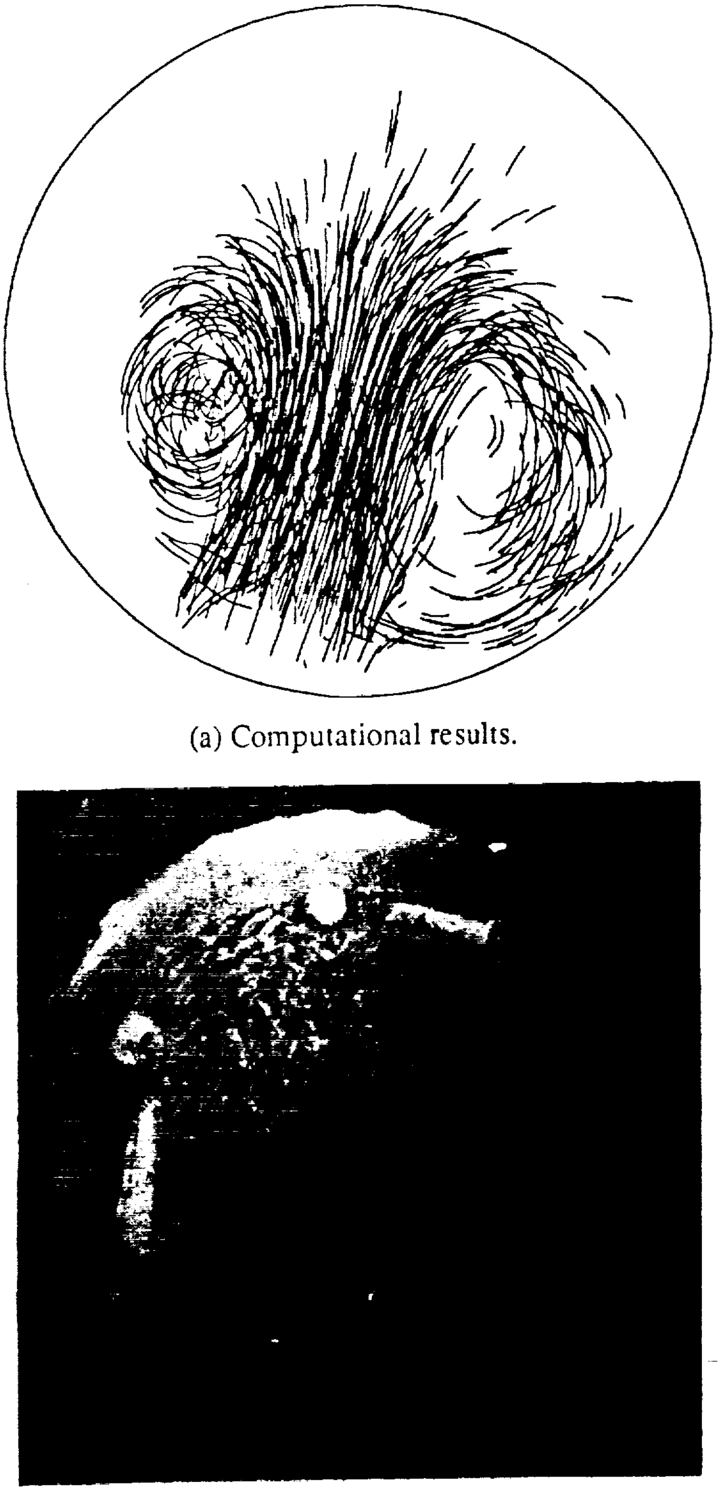
The computations started with the fluid at rest and with the piston at the bottom position and the outflow valve open. The computations were carried out for a Reynolds number of 100 based on unit length and velocity, and the flow was assumed to be laminar. In the actual heart the Reynolds number is about 600, and regions of the flow are turbulent. The laminar assumption is used here because this calculation's main purpose is to test the ability of the flow solver to compute flow through this complicated geometry, separate from the effects of using an inadequate turbulence model. Finally, the fluid is assumed to be Newtonian, which corresponds to the experiment of

Tarbell (ref. 4) who used a water and glycerin fluid whose viscosity is nearly the same as blood, about 3.5 centipoise, but unlike blood, it exhibits a Newtonian fluid behavior.

The use of characteristic relations (refs. 7 and 8) at the inflow and outflow boundaries determines only part of these boundary conditions. At the inflow three variables must held constant, and at the outflow one variable must be held constant. At the inflow valve opening, the total pressure is specified to be constant and the velocity is prescribed to be perpendicular to the open surface. At the outflow valve opening, the static pressure is specified to remain constant. This scheme provides a nonreflective boundary treatment which remains computationally stable. The rest of the boundaries are prescribed to be viscous, no-slip surfaces, where the pressure on the boundary is computed by specifying that the normal pressure gradient be zero.

The artificial compressibility constant β was set to 500. This value was obtained from numerical tests to determine the best convergence during the subiterations. The larger the pseudo-time step $\Delta\tau$ was, the faster the convergence, and therefore it was set to 10^{12}, which effectively set the $1/\Delta\tau$ term to machine zero. The physical time step Δt was set to 0.025. Values of Δt larger than this tended to make the computation slightly unstable. The piston moved with a constant, nondimensionalized velocity of ± 0.2 between its top and bottom positions, requiring 200 physical time steps for one period of the piston's motion. During each time step, the subiterations were carried out until the maximum residual dropped below 10^{-3} or until a maximum of 20 subiterations were used. During most of the piston's cycle only 12-15 subiterations were required, but when the piston was changing directions, it did not completely converge in 20 subiterations. This did not cause any stability problems, yet it remains to be seen what effect this has on the accuracy of the solution. The computing time required for each period of the piston's motion was approximately 4 hr. The computations were run for four periods during which time particle paths were computed after being released near the inflow valve.

Figure 3a shows some of these particle traces as the piston nears its bottom position. Two distinct vortices are seen to have formed from the flow separating as it enters through the inflow valve and encounters the lower pressure regions adjacent to the valve. In figure 3b, an experimental photograph (J. M. Tarbell, private communication, 1988) shows bubbles entering the inflow valve as the piston nears its bottom position. A similar two-vortex system is seen to form here. Figures 4 and 5 show velocity vectors during the inflow phase in planes passing through the center of the inflow valve. The first of these show a top view of vectors in a plane parallel to the piston, while figure 5 shows a side view of vectors in a plane perpendicular to the piston. These figures portray how complicated the vortical structure of this flow is. Figure 4 again shows the presence of two vortices formed as the incoming flow forms a jet. Figure 5 shows also how the flow recirculates underneath the valve opening, but what it does not show is that the flow there is strongly three-dimensional, with the velocity vectors next to the left wall underneath the valve pointing into the paper. The figure also shows the presence of additional vortices or stagnant flow regions against the back wall opposite the valve opening. This is a region which could possibly benefit from a design modification.



(a) Computational results.

(b) Experimental results.

Figure 3.– Incoming particle traces from computations and photograph of experimental results as the piston nears the bottom position.

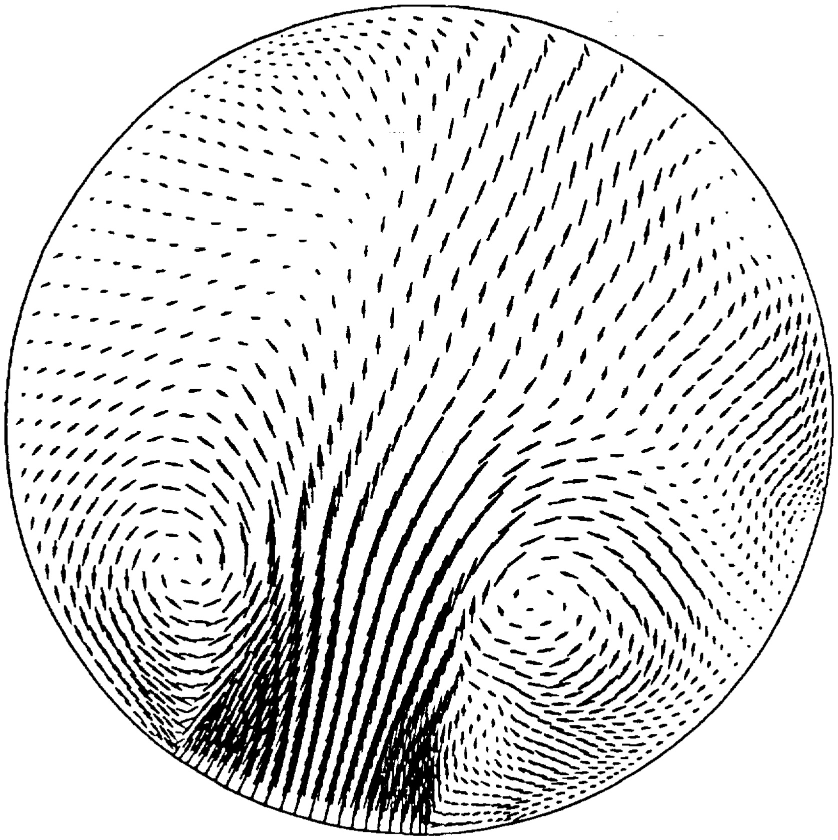


Figure 4.– Top view of velocity vectors in plane through center of inflow valve showing incoming fluid.

CONCLUSION

An algorithm for computing unsteady incompressible Navier-Stokes equations has been extended to simulate the flow through an artificial heart. The present solution shows the capability of the computational procedure for simulating complicated internal flows with moving boundaries. For this initial calculation, the motion of the piston of the actual device was simplified. The fluid was assumed to be laminar and Newtonian. Also neglected were the effects of the valve opening and closing. Even though the computer code, which is based on a nonfactored implicit line-relaxation scheme, converged rapidly, further enhancement in computational efficiency will still be useful. One simple modification would be the use of a multigrid convergence-acceleration scheme. Many different configurations will need to be analyzed in the design of the heart, which will require a faster flow-solver code. One difficulty in performing the present study is that there is currently very little validation which can be done because of the laminar flow assumptions and because the valves were simplified in the computation. It is realized, however, that this work represents a first step toward developing a CFD tool for this type of flow. With the use of more advanced grid techniques, and multiple zones to handle the valves, simulation of the entire artificial heart will be possible in the near future.

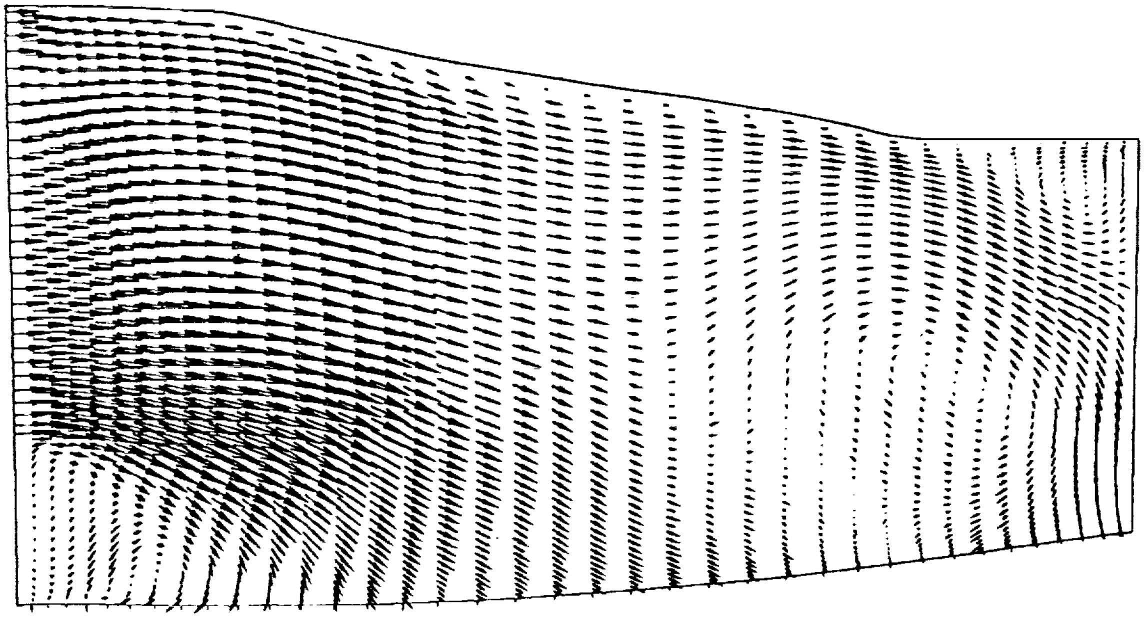


Figure 5.– Side view of velocity vectors in plane through center of inflow valve showing incoming fluid.

REFERENCES

1. Tiderman, W. G., M. J. Steinle, and W. M. Phillips, Two-Component Laser Velocimeter Measurements Downstream of Heart Valve Prostheses in Pulsitile Flow, *J. Biomech. Engr., Trans. ASME*, vol. 108, 1986, pp. 59–64.

2. Liepsch, D., S. Moravec, A. K. Rastogi, and N. S. Vlachos, Measurement and Calculations fo Laminar Flow in a Ninety-Degree Bifurcation, *J. Biomech.*, vol. 15, 1983, pp. 753–766.

3. Liepsch, D., H. J. Steiger, A. Poll, and H.-J. Reulen, Hemodynamics Stress in Lateral Saccular Aeurysms, *J. Biorheology*, vol. 24, 1987, pp. 689–710.

4. Tarbell, J. M., J. P. Gunshinan, D. B. Geselowitz, G. Rosenburg, K. K. Shung, and W. S. Pierce, Pulse Ultrasonic Doppler Velocity Measurements Inside a Left Ventricular Assist Device, *J. Biomech. Engr., Trans. ASME*, vol. 108, 1986, pp. 232–238.

5. Peskin, C. S., The Fluid Dynamics of Heart Valves: Experimental, Theoretical and Computational Methods, *Ann. Rev. Fluid Mech.*, vol. 14, 1982, pp. 235–259.

6. Rogers, S. E., and D. Kwak, An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations, AIAA Paper 88-2583, 1988.

7. Rogers, S. E., and D. Kwak, Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems, AIAA Paper 89-0463, 1989.

8. Merkle, C. L., and P. Y. L. Tsai, Application of Runge-Kutta Schemes to Incompressible Flows, AIAA Paper 86-0553, 1986.

9. Beneck, J. A., P. G. Buning, and J. L. Steger, A 3-D Chimera Grid Embedding Technique, AIAA Paper 85-1523-CP, 1985.

10. Bjorstad, P. E., Numerical Solution of the Biharmonic Equation, Ph.D. Dissertation, Stanford Univ., 1980.

NASA

Report Documentation Page

1. Report No. NASA TM-102183	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

Numerical Simulation of Flow Through an
Artificial Heart

5. Report Date

April 1989

6. Performing Organization Code

7. Author(s)

Stuart E. Rogers (Sterling Federal Systems, Palo Alto, CA),
Paul Kutler and Dochan Kwak (Ames Research Center), and
Cetin Kiris (Stanford University, Stanford, CA)

8. Performing Organization Report No.

A-89099

10. Work Unit No.

505-60

9. Performing Organization Name and Address

Ames Research Center
Moffett Field, CA 94035

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Stuart E. Rogers, Ames Research Center, MS 258-1, Moffett Field, CA 94035
(415) 694-4481 or FTS 464-4481

16. Abstract A solution procedure has been developed that solves the unsteady, incompressible Navier-Stokes equations, and has been used to numerically simulate viscous incompressible flow through a model of the Pennsylvania State artificial heart. The solution algorithm is based on the artificial compressibility method, and uses flux-difference splitting to upwind the convective terms; a line-relaxation scheme is used to solve the equations. The time-accuracy of the method is obtained by iteratively solving the equations at each physical time step. The artificial heart geometry involves a piston-type action with a moving solid wall. A single H-grid is fit inside the heart chamber. The grid is continuously compressed and expanded with a constant number of grid points to accommodate the moving piston. The computational domain ends at the valve openings where nonreflective boundary conditions based on the method of characteristics are applied. Although a number of simplifing assumptions were made regarding the geometry, the computational results agreed reasonably well with an experimental picture. The computer time requirements for this flow simulation, however, are quite extensive. Computational study of this type of geometry would benefit greatly from improvements in computer hardware speed and algorithm efficiency enhancements.

17. Key Words (Suggested by Author(s))

Artificial heart
Incompressible Navier-Stokes
Upwind differencing
Blood flow

18. Distribution Statement

Unclassified – Unlimited

Subject category – 64

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 15	22. Price A–02

For sale by the National Technical Information Service, Springfield, Virginia 22161