INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:467–478 (DOI: 10.1002/fld.306)

Implicit fluid–structure coupling for simulation of cardiovascular problems

J. M. T. Penrose[1,*] and C. J. Staples[2]

[1]*Department of Medical Physics and Clinical Engineering, University of Sheffield, Royal Hallamshire Hospital, Glossop Road, Sheffield S10 2JF, U.K.*
[2]*AEA Technology Engineering Software, Building 8.19 Harwell, Didcot, Oxon. OX11 OQJ, U.K.*

SUMMARY

A three-dimensional (3-D) transient solid–fluid coupling system has been developed in order to study cardiovascular systems and devices. The system utilized two commercial implicit solvers, which exchanged boundary parameters from separate meshes over a common interface. Facility was made for the spatial interpolation of these exchange parameters so that the solid and fluid domain meshes need not have similar density or topology. Stability algorithms were added to the iterative coupling process, as were algorithms to smooth or entirely remesh the fluid domain interior subject to the deformations imposed at the solid–fluid interface. Several application scenarios were undertaken, whereby simulation results could be compared to either analytical or detailed experimental data. It was hoped they would also offer further insight into the operation of a number of clinical devices. The results of these comparisons show that the simulation of complex cardiac systems, with non-linear solid–fluid interactions, can now be achieved with sufficient accuracy to be of significant benefit to manufacturers. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: FEA; CFD; fluid–structure coupling; cardiovascular

1. INTRODUCTION

Over the years, implantable and extra-corporeal devices have significantly improved the life-expectancy and quality of life of patients suffering from cardiovascular disease. However, there are also significant problems and complications associated with many of these devices. These could be partially addressed by better design work arising from a better understanding of the function of these devices, and their interaction with their physiological surroundings [1]. Simulation of these systems can provide valuable information that can be fed back into the design process. Thus, different materials and geometries can be evaluated relatively quickly and cheaply, hopefully leading to better clinical results.

*Correspondence to: J. M. T. Penrose, AEA Technology Engineering Software, The Gemini Building, Fermi Avenue, Harwell International Business Centre, Didcot, Oxfordshire OX11 OQR, U.K.

Contract/grant sponsor: European Commission's IST programme; contract/grant number: BloodSim EP28350

Received May 2001
Revised October 2001

It has long been recognized that the interactions between a fluid phase and a deformable solid phase are of primary importance in a variety of systems, not least in physiological simulation. This is doubly true for cardiovascular simulation where the solid and fluid materials may have complex, non-linear properties; where complex geometries are common, and where even small structural deviations may cause large changes in fluid behaviour (and *vice versa*). However, high-performance computing power and numerical methods have progressed over the last decade to the point where such three-dimensional (3-D) transient coupled simulations can be achieved within realistic time-scales.

This paper describes the development of a transient coupled 3-D finite-element system aimed at the simulation of such cardiovascular problems, backed by a consortium of cardiac device manufacturers.

2. IMPLEMENTATION

Several commercial and research codes exist for the coupling of solid and fluid mechanics. However, the vast majority of these codes have become available only over the last few years, and so algorithmic details, case studies and examples, particularly in physiological areas, are often hard to find. Perhaps, the most well-known cardiovascular research is that of Peskin and McQueen at NYU [2]. They have managed to produce both 2-D and 3-D models of the heart, complete with functioning valves, within which the blood is pumped purely by the constrictive action of the ventricle walls. They use a finite volume code which simultaneously solves for the solid–fluid interaction based on the so-called 'immersed boundary' method. Here, the solid domain is not explicitly represented within the fluid domain, but rather exists only by the additional force 'field' that the solid exerts on the fluid where the two domains would overlap. Such a method can be straightforward to apply, but the researchers' models are very detailed and computationally costly. The method may also be only applicable to 'thin' structures.

An alternative coupling scheme to the simultaneous solution is that whereby two separate solvers are used to solve the two distinct solid and fluid phases. The separate phases are solved sequentially at a given time, with the boundary solution from one used as a boundary constraint for the other. Such a coupling has been exploited by Perktold and colleagues at the University of Graz [3–5]. They have coupled the commercial solid code Abaqus to their own fluid code (an arbitrary Lagrangian/Eularian FE code using the Galerkin method) exploiting the iterative coupling scheme. They have used it to study artery dynamics and anastemoses in 3-D. A similar hybrid iterative scheme, coupling the fluid code CFX and the solids code Abaqus, has been developed by Prof. Collins and his team at City University and Imperial College in London [6–8]. They have produced some impressive 3-D simulations of blood flow through the carotid bifurcation, using initially a coupling over a single cardiac cycle, as oppose to a single timestep of that cycle.

More recent developments have shown coupled simulations performed within a single software package. Kunzelman *et al.* [9] have modified the explicit LS-Dyna package, which now boasts an incompressible flow solver in addition to their well established structures solver, and produced some impressive 3-D simulations of cardiac valves.

It was decided that for the purposes of this research two existing commercial codes, Ansys and CFX, would be coupled, thereby exploiting the particular strengths of each in simulating

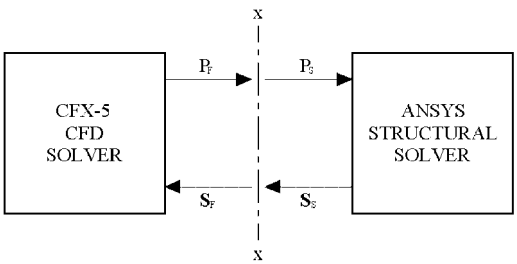

Figure 1. Solution variable exchange between the two solvers.

their particular phase physics. Both codes also offer parallel processing solvers, thus enabling significant reductions in simulation time on multi-processor architectures.

2.1. CFX-5

CFX-5 is a commercial CFD package that uses an implicit finite-element control volume formulation to construct the discrete equations representing the Navier–Stokes equations for the fluid flow. The mixed-element mesh can be unstructured and composed of hexahedra, tetrahedra, wedges and pyramids. A coupled algebraic multi-grid solver is used to give robust solutions for the governing discrete equations.

2.2. ANSYS

ANSYS [10] is a world-leading implicit finite-element software product, for linear and non-linear stress analysis. It has the facility for a wide range of structural elements and formulations, and again meshes can be mixed and unstructured. It is used for calculating the response of any deformable regions within the coupled simulation, based on appropriate loading from the fluid.

2.3. Overview of basic coupling approach

The basic coupling approach was to use each solver to sequentially solve the separate phases, on separate domain meshes that overlap only over a ‘wet’ boundary interface. Ideally, the full-fluid stress tensor is passed from the fluid solution as a boundary load to the solid domain, with the resultant deformation of that boundary returned as a fluid constraint. This procedure is repeated at a given timestep until both solutions consistently produce the same result. Simulation then proceeds to the next timestep. However, it is quite usual for such a system to be unstable due to its inherent non-linearity; small structural deformations can cause quite large changes in the flow and pressure distributions within the fluid, which in turn can have a significantly different effect on the solid deformations. Hence a simple under-relaxation scheme was employed to stabilize the exchange parameters, in both directions:

$$\Phi_n = r\Phi_{\text{calc}} + (1 - r)\Phi_{n-1} \tag{1}$$

Also, to simplify the exchange it was decided that only the dominant static pressure term would be passed from the fluid phase to the solid. Figure 1 is a simplified diagram of the approach taken. x–x represents the fluid–structure interface. P_F is the nodal fluid pressure field,

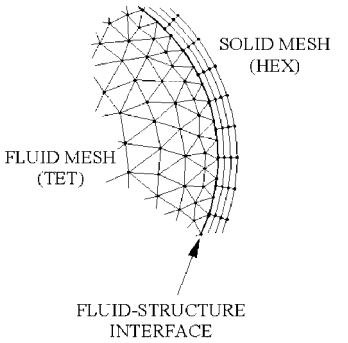

Figure 2. Unmatched solid and fluid meshes.

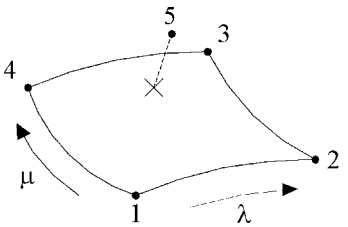

Figure 3. The parametric quadrilateral face interpolation.

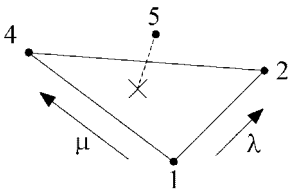

Figure 4. The parametric triangular face interpolation.

$\mathbf{P}_S$ the nodal solid pressure field, $\mathbf{S}_S$ represents the nodal solid displacement vector field and $\mathbf{S}_F$ the nodal fluid displacement field. A *coupling cycle* comprises a single coupled run of the fluid and solid solvers.

2.4. Surface interpolation

The fluid–structure interface is a surface where the fluid and structural meshes conjoin and is defined by a set of element faces and nodes on each mesh. (Figure 2). In many cases, the density or element type of the meshes will differ and the nodes of the two meshes will not coincide spatially. In order to pass solution-variable data across such an interface, surface interpolation must be carried out. A simplified description of this process is as follows:

Consider the fluid mesh only. A mapping is generated for all nodes on the interface surface, as illustrated in Figures 3 (quadrilateral face) and 4 (triangular face). For a given node this

mapping comprises:

- The associated face in the structural mesh defined by nodes 1–4.
- The parametric position defined by λ and μ of the projection of the fluid node (5) onto the above face.

Where the associated face is a triangle, $\lambda + \mu = 1$.

Similar mappings are generated for all interface nodes in the structural mesh. Clearly, some faces may be associated with several nodes; conversely others may be associated with no nodes. If the topology of the surface mesh is to be maintained throughout the simulation, these mapping procedures need only to be carried out once, at the start of the simulation.

The surface interpolation, carried out at each coupling cycle is given in Equation (2) (quad face) and 3 (triangle face)

$$\phi_5 = \mu\lambda(\phi_1 - \phi_2 + \phi_3 - \phi_4) + \mu(\phi_4 - \phi_1) + \lambda(\phi_2 - \phi_1) + \phi_1 \tag{2}$$

$$\phi_4 = \mu(\phi_3 - \phi_2) + \lambda(\phi_2 - \phi_1) + \phi_1 \tag{3}$$

where ϕ is the interpolated variable. Such an interpolation scheme is deemed valid provided that the fluid and structural meshes are of similar density and that the variation of the interpolated variable over a given element face is small [11].

2.5. *Propagation of mesh displacement*

Displacement of the fluid domain is defined by the nodal displacement field at the fluid–structure interface, as calculated by the structural code. In order to maintain a valid fluid mesh, the remaining internal nodes must have their positions adjusted, whilst preserving the mesh topology. The method chosen was based on a simple Laplacian smoothing operation [12]. The mesh is modelled as a spring system where the displacement of each internal node is calculated based on the displacements of its nearest neighbours using a simple iterative procedure

$$S^0 = S^m, \quad S^{(k)} = \frac{\sum_{p=1}^{n} S^{(k-1)}}{n}, \quad k = 1, 2, \ldots, q, \quad S^{(m+1)} = S^q \tag{4}$$

where S is the displacement vector, m is the time-step, q is the number of smoothing iterations and n is the number of neighbouring nodes.

A simple weighting procedure was then added to this scheme based on the distance to each near neighbour, thus better preserving the original mesh topology. In addition, rather that repeating the smoothing procedure for a given number of iterations, a control was introduced whereby the iterations proceeded until the incremental displacement of the internal nodes was small in comparison to the smallest element size.

2.6. *Mesh movement terms*

In order to account for the arbitrary movement of the mesh, it is necessary to modify the governing discrete equations to include the motion of the mesh as well as the fluid in the transient convection terms. If this is not done, a number of artefacts can result, including mesh-generated flows, when a mesh is moved through a stationary flow [13].

The CFX-4 implementation is described approximately in Reference [14]. The implementation presented here follows their approach, although it has been extended to the cell-vertex treatment implemented in CFX-5.

The standard transient and convective terms in the Navier–Stokes equations can be written in the form

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi u_j}{\partial x_j} \tag{5}$$

where ϕ is the conserved quantity and u_j are the cartesian velocity components. On transformation to a moving co-ordinate system, this becomes

$$\frac{1}{\sqrt{g}} \frac{\partial(\sqrt{g}\phi)}{\partial t} + \frac{\partial(\phi(u_j - v_j))}{\partial x_j} = \frac{\partial \phi}{\partial t} + \frac{\phi}{\sqrt{g}} \frac{\partial \sqrt{g}}{\partial t} + \frac{\partial(\phi(u_j - v_j))}{\partial x_j} \tag{6}$$

In these equations g is the determinant of the metric tensor of the transformation and $\sqrt{g}$ therefore represents the volume in physical space corresponding to a unit volume in computational space. v_j is the mesh velocity component. The changes that are required therefore are to add an extra transient term as a volumetric source and to modify the convective velocity subtracting the mesh velocity. However, this must be done in a consistent fashion, so as not to create spurious sources and sinks within the equations.

In a cell-vertex formulation, the control volumes are formed from mesh sectors that surround the nodal points. The convective velocities u_j are required at 'integration' points, in the centre of the sector faces, in order to calculate the convective fluxes across the sector boundaries. See Reference [15] for details. With a moving mesh, the mesh velocity is interpolated from the nodes to the integration points and added to the conservative velocity at these points. The conservation of volume constraint arises from Equation (6), by replacing the variable ϕ with unity.

The equation becomes

$$\frac{1}{\sqrt{g}} \frac{\partial \sqrt{g}}{\partial t} - \frac{\partial v_j}{\partial x_j} = 0 \tag{7}$$

so that

$$\frac{1}{\sqrt{g}} \frac{\partial \sqrt{g}}{\partial t} = \frac{\partial v_j}{\partial x_j} \tag{8}$$

Integrating this over the control volume, we obtain

$$\frac{1}{\sqrt{g}} \int \left(\frac{\partial \sqrt{g}}{\partial t} \right) \mathrm{d}V = \int \left(\frac{\partial v_j}{\partial x_j} \right) \mathrm{d}V = \int (v_j n_j) \, \mathrm{d}S \tag{9}$$

Equation (9) is used to define the volumetric source term in Equation (6), from the mesh velocities, rather than from direct integration of the volume changes. It only requires a knowledge of the mesh velocity, obtained from the movement of the nodal mesh points only and gives a consistent treatment, which eliminates many potential artefacts of the approach. For incompressible, isothermal flows, the only terms requiring modifications are the above convective and transient terms, as the systems considered do not have an enthalpy equation.

This implementation has been tested on a number of different cases; for example, moving meshes through stationary fluid, rigid body fluid motion, to demonstrate that the correct results are obtained.

2.7. *Re-meshing*

Despite the smoothing procedures performed on the internal fluid nodes, if the solid displacements are large there is likely to be a time when one or more fluid elements become so distorted that their volume becomes zero or negative and the mesh no longer remains valid. At such a point, there are two approaches to enable the simulation to continue: First, if the invalid displacement configuration can be guessed *a priori*, then an alternative fluid mesh can be prepared in advance. When the simulation displacement reaches this point, the field variables can be interpolated onto the alternative valid mesh, and the simulation can proceed. Several alternative meshes can be prepared for various displacement configurations. However, there is a significant drawback to this semi-automatic method; unless the solid deformations can be approximated to a single-degree-of-freedom system, it is very difficult to predict when the fluid mesh will become invalid, and also to produce an alternative fluid mesh that will be close enough to the topology of the actual simulation mesh.

The second approach is to actually remesh a section or the entire fluid domain, thus correcting the invalid elements. An automatic scheme such as this was developed, and invoked when a negative volume element was detected after the mesh smoothing procedure. The surface of the fluid domain was retained while the CFX tetrahedral Delaunay mesher 'Meshgen' was invoked to remesh the interior. The field parameters from the last converged timestep were then interpolated onto this new mesh and the simulation could proceed as before. An additional benefit was that since the surface topology was conserved, the interpolation parameters for the exchange parameters at the solid–fluid interface were maintained and did not require re-calculation.

2.8. *Volume interpolation*

As previously described, the fluid mesh may be regenerated, the new mesh having a different topology to the previous. In order that the simulation may proceed, the solution variables, stored on nodes, must be interpolated from the previous mesh to the new mesh. This is accomplished using a second-order tri-linear interpolation procedure. The CFD simulation may then be re-started at the next timestep on the new mesh, using the interpolated solution variables as the initial conditions. However, in using the interpolation with the remeshing schemes it was noticeable that whilst the displacement results were continuous from one timestep to the next, the pressure results often showed an artefact for the single timestep where interpolation took place. Even though there are no constraints on the pressure being continuous, there is no clear reason for this effect.

2.9. *Rigid body simulation*

In some simulation scenarios, a body in contact with, or immersed in, the fluid flow undergoes so little structural deformation that it can be considered rigid. In these situations it was considered computationally inefficient to use the ANSYS solver, and so a separate rigid body solver was developed to compute the response of these bodies to the flow. The familiar

equations of motion that are required to be solved, defined in the fixed global axis system, are

$$m\ddot{U} + C\dot{U} + KU = F(t) \tag{10}$$

$$I\dot{\omega}' + \omega' \times (I.\omega') = M(t) \quad \text{where } I = \Lambda J \Lambda^{\mathrm{T}} \quad \text{and } \omega' = \Lambda\omega \tag{11}$$

Here, $F(t)$ and $M(t)$ are the externally applied force and moment functions derived from the fluid flow, m is the body mass, C and K are damping and stiffness matrices, ω is the angular velocity vector, J is the constant inertia tensor of the body and Λ is a tensor transformation matrix from the body's own axis system to the global axis. The rotational motions can be separated from the translational motion of the centre of mass and so different solution approaches can be separately adopted for each, according to suitability. For the translational motion, the Houbolt implicit time-integration scheme [16] was chosen to discretize Equation (10). For the rotations, an implicit algorithm proposed by Simo and Wong [17] has beenemployed to solve the momentum balance equation in conservation form using a variation on the implicit Newmark integration scheme [16]. Here, a rotation increment is iteratively calculated ensuring that energy and angular momentum are conserved correctly.

Additional facilities were added to this scheme to allow for rigid body degree-of-freedom constraints, initial velocities and rotations, as well as simple linear and torsion springs. This allowed for the simulation of moderately complex rigid body systems such as valves.

3. APPLICATIONS

The consortium were interested in a wide range of cardiovascular devices, employed in both simulated *in vivo* and *in vitro* situations. Thus to ensure that the coupled simulations were providing valid results, a series of test application cases were devised, whereby the results could be directly compared to either analytical solutions or well established experimental data. Some examples of these application cases are presented here.

3.1. Flow in an elastic vessel

The transient progression of a pressure pulse down a tube has been studied by many investigators over the years, and a good review of these can be found in Reference [18]. The initial work on wave propagation in an elastic tube has been attributed to Moens and Korteweg in the latter part of the 19th century, based on Newton's work on the speed of sound in air. Taking t as the thickness of the wall and E as the circumferential Young's Modulus of the wall, and relating the change in radius to the applied pressure, the following can be written:

$$c_0 = \sqrt{\frac{Et}{2R\rho}} \tag{12}$$

This is known as the Moens–Korteweg equation and is based upon several important assumptions: (1) the fluid is 'ideal' incompressible and inviscid, (2) the solid wall is thin and there are no changes in its thickness. The effect of viscosity in the fluid was studied by Womersley [19] in the 1950s, who assumed a sinusoidally variant input pressure, at frequency ω. He showed that in the limiting condition of complete longitudinal constraint

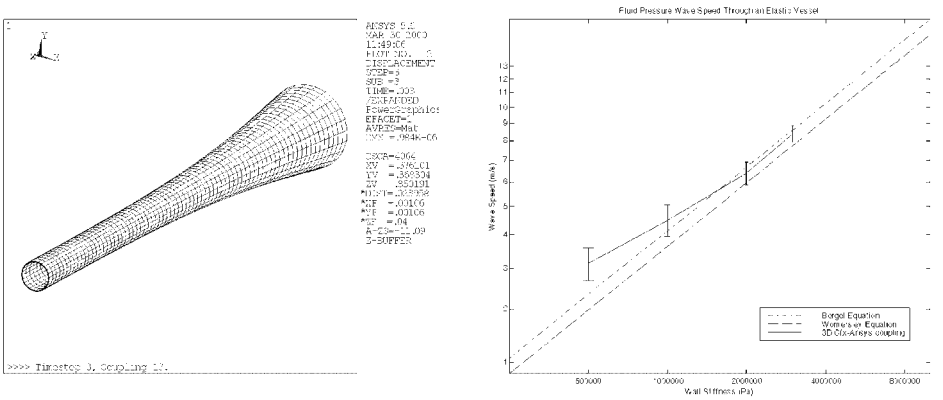

Figure 5. The exaggerated deformation of the vessel wall (left), and the simulation/analytical comparison of wave propagation speed (right). Here, the dotted line is the Moens–Kortweg solution, and the dashed line is the Womersley solution.

(no axial movement of the wall), the following equation for the complex wave speed through an elastic vessel could be written

$$c = c_0\sqrt{\frac{1-F_{10}}{1-\upsilon^2}} \quad \text{where} \quad \left(1-\frac{2J_1(\alpha i^{-3/2})}{\alpha i^{-3/2}J_0(\alpha i^{-3/2})}\right) = [1-F_{10}] = M'_{10}e^{i\varepsilon_{10}} \tag{13}$$

Here, c_0 is the wave speed for an inviscid fluid (the Moens–Korteweg equation), and the real part gives the actual wave speed. R is the vessel radius, ρ the fluid density, α is the dimensionless Womersley parameter where $\alpha^2 = R^2\omega\rho/\mu$ (μ is the fluid viscosity), and J_0, J_1 are Bessel functions of order zero and one, respectively.

Using a simple quarter-cylinder symmetric model for the elastic vessel, it was possible to compare the coupled simulation results with these analytical solutions. A ramped pressure inlet condition was used, together with a timestep of 1 ms, and simulations were conducted over a range of wall stiffness values. In the results, the dilation of the vessel wall could clearly be seen propagating along the vessel at a constant speed (Figure 5) and this speed was plotted alongside the analytical solutions (see Figure 5).

It can be seen that the simulation results are in good agreement with the analytical solutions, particularly considering the assumptions made in the analyses about the wall thickness and fluid behaviour.

3.2. St Jude replacement heart valve

Rigid bi-leaflet mechanical valves, like the St Jude valve, represent over 60% of the annual implants world wide. This particular valve features two symmetrical hinged leaflets made of pyrolytic carbon (see Figure 6), which can be considered rigid under typical physiological loading conditions. Therefore, for this simulation the rigid body solver was used to couple with the fluid solutions, rather than Ansys. The simulation domain and parameters were designed to closely mimic an *in vitro* experimental set-up, data from which provided a validation of the simulation as a whole. A physiological flow rate waveform was applied at the inlet, a quarter-symmetry mesh model was used, and no constraints were placed on the leaflet motion

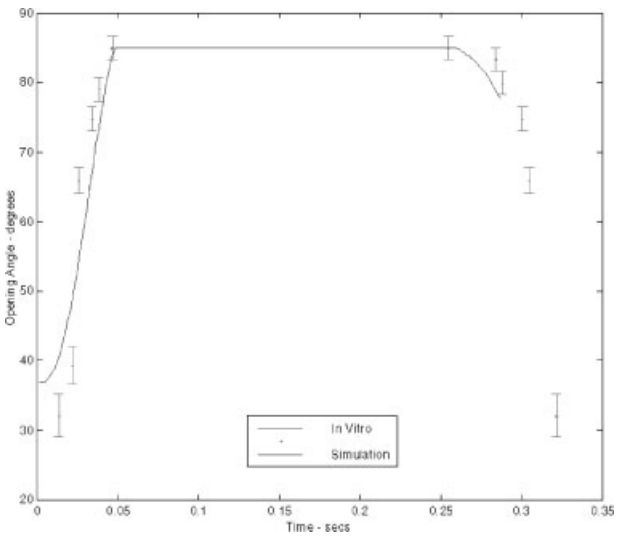

Figure 6. Comparison of the simulated valve opening angle with that measured from high-speed video *in vitro*.

other than the simple hinge. The semi and fully automatic remeshing methods were employed to cope with the large rotational deformation of the leaflets. As the modelling of narrow fluid gaps is difficult, the valve was initially meshed slightly open by 5°, and no effort was made to simulate complete closure.

Solutions were obtained for the entire of systole using timesteps of 1 ms (see Plate 1 for example), and took the order of several days to compute on a multi-processor IBM-SP3 platform. Comparisons of these results were made to data obtained from the experimental set-up. In order to compare the opening and closing characteristics of the valve, a high-speed video camera (with frame rates of up to 600 fps) was employed to record the valve from various positions. From direct measurement on the screen of the recordings, together with the known fully open/closed angles, it was possible to estimate the opening and closing times of the valve, as well as its opening angle during opening/closure with respect to time. Errors were calculated based on an error of ± 2 mm on the measurements.

The graph of the simulated opening angle can be seen in Figure 6, with a 90° opening representing the leaf parallel to the long axis of the valve. It shows that whilst the opening time of the simulated valve is somewhat slower than the experiments indicate (40–50 ms compared to 30–35 ms), the overall behaviour of the simulated valve is remarkably similar to the experimental valve. Part of the difference might also be attributable to the simulated valve starting at a slightly open position.

3.3. Berlin Heart cardiac assist device

The Berlin Heart is an extra-corporeal cardiac assist device, of which there are several varying designs based around the same principle; the predominantly rigid central chamber contains the blood, has an inlet and an outlet, and is bounded by two opposing flexible roll membranes. The pump inlet would normally be connected directly to the ventricle via a cannula, with

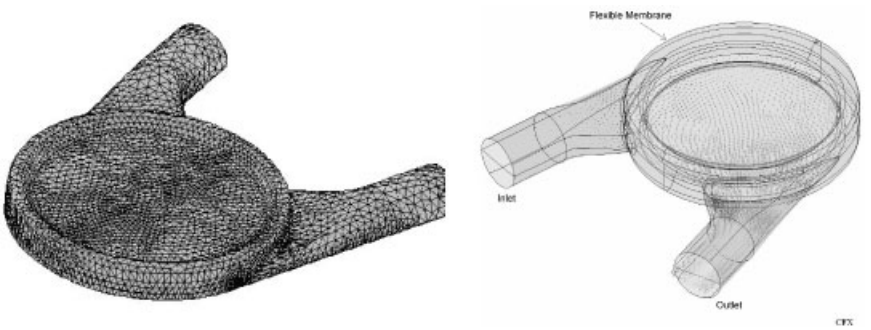

Figure 7. The symmetrical pump fluid mesh (left), and velocity vectors along the symmetry plane during the 'pump' phase (right).

another connecting the outlet to the aorta. Valves are built onto the blood chamber inlet and outlet to ensure the direction of flow, and these valves are usually rigid tilting disc valves (although they were not directly modelled) or a polyeurythane tri-leaflet design. During the pump phase, symmetrically opposing pistons contact the membranes and push them towards each other, thus increasing the pressure in the chamber and forcing the blood through the outlet. During the fill phase, the pistons retreat and contact with the membranes is broken. The movement of the membranes is then constrained only by the action of the blood entering via the inlet, and the pressure within the chamber becomes low thus retarding the movement of the membranes.

Simulation results were obtained over an entire 'cardiac' cycle using timesteps of 10–20 ms (see Figure 7). These were compared to experimental flow field results obtained from laser light sheet and laser Doppler experiments. The simulated flow was clearly seen to pulsate from inlet to outlet and several important features, such as flow separation regions and particular membrane deformations, were identifiable in both the simulation and experiment.

4. CONCLUSIONS

Studies with this coupling system have demonstrated that accurate simulation of cardiovascular systems is now within reach. Indeed, the application of the system could equally extend into any field where fluid–structure interactions are of primary concern. The data from such simulation will provide invaluable benefit to device manufacturers that will optimize the design and prototyping process, reducing cost and ensuring that safer and better understood products reach the market.

ACKNOWLEDGEMENTS

The authors acknowledge the partial support provided for this work by the European Commission's IST programme, under project BloodSim EP28350 and the contribution of all the project partners. All trademarks are acknowledged.

REFERENCES

1. Hose D, Black M. Prosthetic heart valves—the integration of analysis with design. *The Journal of Heart Valve Disease* 1995; **4**(1):50–54.
2. Peskin CS, McQueen DM. A general method for the computer simulation of biological systems interacting with fluids. *Symposium of the Society for Experimental Biology* 1995; **49**:265–276.
3. Perktold K, Rappitsch G. Computer-simulation of local blood-flow and vessel mechanics in a compliant carotid-artery bifurcation model. *Journal of Biomechanics* 1995; **28**(7):845–856.
4. Hofer M, Rappitsch G *et al*. Numerical study of wall mechanics and fluid dynamics in end-to-side anastomoses and correlation to intimal hyperplasia. *Journal of Biomechanics* 1996; **29**(10):1297–1308.
5. Perktold K, Hofer M *et al*. Validated computation of physiologic flow in a realistic coronary artery branch. *Journal of Biomechanics* 1998; **31**(3):217–228.
6. Collins MW, Zhao SZ *et al*. The application of computational fluid dynamics and solid mechanics to haemodynamics in arterial organs and to related problems. *Cardiovascular flow modelling and measurement with application to clinical medicine*. The Institute of Mathematics and its Applications, 1998.
7. Zhao SZ, Xu XY *et al*. The numerical analysis of fluid–solid interactions for blood flow in arterial structures—Part 1: a review of models for arterial wall behaviour. *Proceedings of the Institution of Mechanical Engineers Part H–Journal of Engineering in Medicine* 1998; **212**(H4):229–240.
8. Zhao SZ, Xu XY *et al*. The numerical analysis of fluid–solid interactions for blood flow in arterial structures—Part 2: development of coupled fluid–solid algorithms. *Proceedings of the Institution of Mechanical Engineers Part H-Journal of Engineering in Medicine* 1998; **212**(H4):241–252.
9. Kunzelman K, Einstein D *et al*. Computer modeling for evaluation of cardiac valvular function. *World Congress on Medical Physics and Biomedical Engineering*, Chicago, 2000.
10. *ANSYS*. Swanson Analysis Systems Inc.: Pittsburgh, http://www.ansys.com/
11. Farhat C, Lesoinne M *et al*. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**:95–114.
12. Demirdzic I, Peric M. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids* 1998; **8**:1037–1050.
13. Hassan O, Probert E *et al*. Unstructured mesh procedures for the simulation of three-dimensional transient compressible inviscid flows with moving boundary components. *International Journal for Numerical Methods in Fluids* 1998; **27**:41–55.
14. Hawkins I, Wilkes N. Moving grids in Harwell-FLOW3D, *AEA-InTech*-0608, 1991.
15. *CFX-5.4*. AEA Technology Engineering Software: Harwell, Oxon, http://www.aeat.com/cfx
16. Bathe K-J. *Finite Element Procedures*. Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1996.
17. Simo JC, Wong KK. Unconditionally Stable algorithms for rigid body dynamics that exactly preserve energy and momentum. *International Journal for Numerical Methods in Engineering* 1991; **31**(1):19–52.
18. McDonald. *Blood Flow in Arteries*. Edward Arnold: Paris, 1990.
19. Womersley J. Oscillatory flow in arteries: the constrained elastic tube as a model of arterial flow and pulse transmission. *Physics in Medicine and Biology* 1957; **2**:178–187.

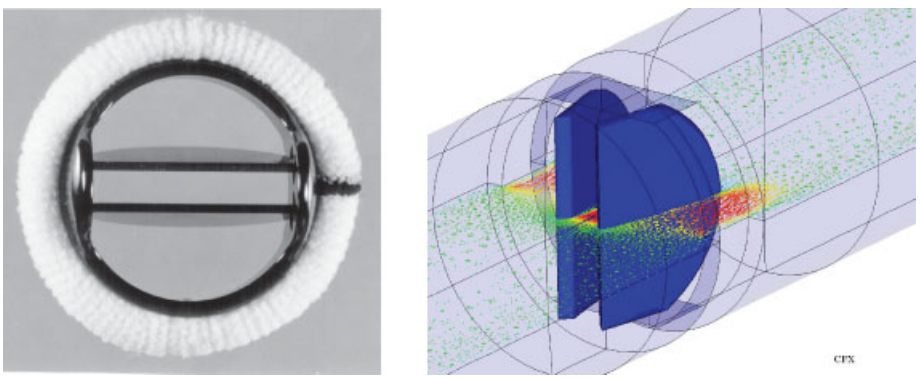

Plate 1. The St Jude valve (left), and the simulation results midway through valve opening (right). Velocity vectors are shown on a horizontal plane through the model.