INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:471–489
Published online 22 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.826

Multipoint constraint methods for moving body and non-contiguous mesh simulations[‡]

David K. Gartling[*,†]

Sandia National Laboratories, Engineering Sciences Center, Albuquerque, NM 87185, U.S.A.

SUMMARY

A sliding mesh method is demonstrated for moving body simulations involving thermal and fluid problems. Static problems with non-contiguous mesh constructions are also solved using the same methodology. The proposed algorithm employs a parallel implementation of multipoint constraints with an efficient node-in-mesh search procedure. A simple method for treating the problem of partially covered/exposed element surfaces is also summarized. Several examples of static and dynamic thermal conduction and viscous flow problems illustrate the algorithms. Published in 2004 by John Wiley & Sons, Ltd.

KEY WORDS: finite element; sliding mesh; multipoint constraint

1. INTRODUCTION

Bodies with prescribed relative motions or deforming bodies responding to induced forces are often difficult to accommodate in a computational analysis, in part, because of the required mesh motions and mesh update strategies. Fluid dynamics problems, described in an Eulerian reference frame accompanied by the Lagrangian motion of a body, are typical of this type of computational difficulty. Problems of this type are usually denoted as fluid–structure interaction (FSI) problems. Another common example of this type of problem is the relative motion of electrically conducting bodies within a magnetic field, with or without an accompanying fluid motion. Finally, contact mechanics for thermal, structural and/or electrical problems often involve the relative motion of different regions and the resulting surface mechanics. In all

[*] Correspondence to: D. K. Gartling, Sandia National Laboratories, Engineering Sciences Center, Albuquerque, NM 87185, U.S.A.

[†] E-mail: dkgartl@sandia.gov

[‡] This article is a U.S. Government work and is in the public domain in the U.S.A.

Contract/grant sponsor: U.S. Department of Energy; contract/grant number: DE-AC04-94AL85000

Received 20 February 2004
Revised 6 May 2004

of these cases, at least part of the computational difficulty is due to the need to resolve surface interactions between regions that are most naturally described with independent, non-contiguous meshes.

Because of the wide variety of interaction problems, a broad spectrum of computational methods have been developed and demonstrated. Since the primary interest here is in fluid and thermal problems, the following descriptions will be focused on those areas; the more complex solid mechanical contact and deformation problem is not considered. Of the work in fluid dynamics, the interface tracking and interface capturing strategies outlined and demonstrated by Tezduyar *et al.* [1–3] are the most noteworthy for their wide applicability and robustness; an excellent summary of this work can be found in Reference [4]. In general, these front tracking methods use an Arbitrary Lagrangian–Eulerian (ALE) formulation, in a space–time framework. Stabilizing procedures, such as streamline upwind Petrov Galerkin (SUPG) [5] and pressure stabilizing Petrov Galerkin (PSPG) [6], are an essential part of the algorithms. For regions involving limited motions, mesh moving schemes are used for the fluid domain. The requisite mesh motion is determined by solution of the equations for an elastic body. When deformations become large, mesh motion is combined with the judicious use of remeshing to achieve a useful overall method. Automated mesh generation is an important ingredient in this approach and therefore suggests the use of tetrahedral finite elements or volumes in large parts of a complex domain. The interface capturing methods outlined in Reference [4] use volume of fluid [7] and level set [8] ideas as a base with numerous enhancements for accuracy and robustness; the implementation is in the space–time framework. A variety of similar techniques have been developed for more specialized applications, including those employed by Lynch [9], Cairncross *et al.* [10], Löhner [11], Cruchaga [12] and others. Another type of fluid–structure interaction method is the immersed boundary method originated by Peskin [13, 14] with its later implementation in a finite element method due to Zhang *et al.* [15]. The fictitious domain method due to Glowinski *et al.* [16] has also been used by Bertrand *et al.* [17] for relative motion problems. Each of these methods have advantages and disadvantages, with none being universally applicable. Continuous remeshing would be the most general but also the least affordable for large, complex flows. The work described here is focused on the motion of rigid bodies; deforming bodies could be processed in the current scheme but are not the primary concern in this work. The approach to be evaluated employs ‘sliding’ meshes and multipoint constraints to continually reattach the moving mesh regions to the nonmoving mesh regions. This is similar in spirit to mortar methods [18] and the shear-slip mesh update method of References [21, 22], though it is simpler to implement and has utility beyond relative motion problems. Multipoint constraints, as described here, also have use in simplifying mesh construction through non-contiguous meshing and meshes with varying element geometries and interpolation orders. Components of the present approach also have utility in the implementation of physical boundary conditions such as contact and periodic boundaries.

The plan of the paper is as follows. The next section outlines the continuum application areas and the components of the algorithm and is followed by a description of the basic search procedure that is central to the efficient implementation of the technique. A short section then discusses the constraint equations needed and their implementation in several types of mechanics applications. A section of example simulations demonstrates the use of the method for thermal and fluid problems in both static and dynamic situations. Some conclusions and observations are at the end of the paper.

2. NUMERICAL METHOD

2.1. Continuum problems

The sliding and non-contiguous mesh procedures have great generality and have been applied to thermal conduction problems, non-isothermal, incompressible, viscous flow problems, and quasi-static electromagnetics. As the continuum description of these problems is standard, the field equations for all of these applications will not be reiterated here. A summary description of each formulation will be sufficient. The time-dependent, thermal conduction problem was described in a Lagrangian frame and used a Galerkin finite element approximation for spatial discretization. Standard linear and quadratic temperature interpolation on two- and three-dimensional elements was employed. Prescribed rigid material motions are easily accommodated in the simulation software and material deformation and motion can be included through coupling with a solid mechanics simulation. For the non-isothermal, viscous flow application, the moving mesh region was described by an Arbitrary Lagrangian Eulerian (ALE) formulation; the stationary mesh regions used a standard Eulerian formulation. The field equations of mass, momentum and energy were discretized in primitive variable form with a Galerkin-based finite element method. A mixed interpolation method was used with quadratic interpolation for the velocity and temperature on triangles, quadrilaterals, hexahedrons and tetrahedrons. The pressure was interpolated on each element with either continuous or discontinuous linear functions. The fluid description in an ALE formulation is given by the continuity and momentum equations as

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\rho \frac{\partial u_i}{\partial t} + \rho(u_j - u_j^m)\frac{\partial u_i}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \tag{2}$$

where ρ is the fluid density, u_i is the velocity, τ_{ij} is the stress tensor and f_i is the body force vector. The mesh velocity is u_j^m and is required in the advection term whenever mesh motion is employed. A similar advection term is required in the fluid energy equation when the flow is non-isothermal.

Though not detailed or demonstrated in the present work, the relative motion algorithms are also applicable to problems in quasi-static electromagnetics. For this type of application a nodal-based finite element method for the electric scalar potential and the magnetic vector potential was used for the development. The equations were discretized in both Lagrangian and Eulerian formats; the Eulerian description again required an ALE type advection term similar to the one in (2) for describing mesh motion in the magnetic equation. A gauge condition, when needed, was enforced with a penalty method. Linear and quadratic interpolation for the potentials was used over a variety of two and three dimensional elements.

The computational implementation for each of these applications was designed for use on both single processor and massively parallel, distributed memory computers. The parallel implementation was based on domain decomposition and the use of MPI [19].

2.2. Sliding mesh method

Sliding mesh methods are not new but have been used previously in electromagnetics for rotary motion simulations (e.g. Reference [20]) and in fluids for both translational and rotational motions (e.g. References [21, 22]). Many of these implementations employ a mortar finite element method, though commercial software implementations seem to favour finite volume methods with a flux matching algorithm at the moving interfaces. The implementation proposed here is designed to force continuity of the dependent variables across the interface or a specified jump in the dependent variable. The method is intended for large scale applications with numerous constraint surfaces; efficient implementation in both single processor and parallel environments is also a requirement. Though the description presented here is for object–object interactions with an intervening fluid region, the method generalizes to other types of interactions. A fluid region around the body or bodies in relative motion is not required but provides the most general and most complex situation.

A schematic of a typical case is shown in Figure 1 where a solid object is prescribed to move relative to another solid region. The region between the solids is filled with an incompressible, viscous fluid. A simple geometric surface is defined between the solids along which the meshes attached to the two solids will slide. For translational motions the intervening surface will be a plane and for rotational motions the surface will be a cylinder or sphere. Typically, the one mesh region is labelled the master and the other mesh region is labelled the slave. Along the common mesh interface the meshes move such that the nodes of the slave mesh are always in contact with the master mesh elements. At each time in the simulation,

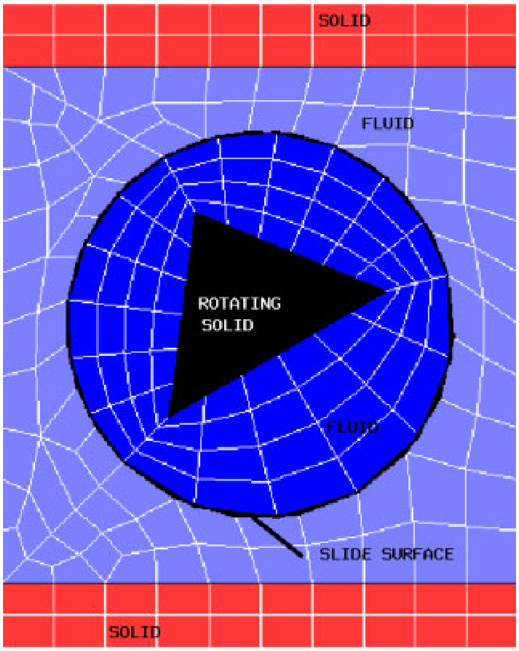

Figure 1. Schematic of a rigid body moving in a fluid relative to a stationary body. The slide surface is circular.

constraint conditions are developed that constrain some or all of the dependent variables from the slave mesh to be the interpolants of the variables in the master mesh. The kinematics of the mesh surfaces are assumed to be known and physically valid, i.e. interpenetration of the meshes and the development of mesh overlap and voids are not considered. Experience has shown that the occurrence of small overlaps, gaps and the use of non-zero capture distances can be tolerated with the energy and electromagnetic equations.

As outlined above, this construction is geometrically very limiting since only simple, prescribed translational and rotational motions can be accommodated. However, a surprising number of complex simulations can make use of this approach. In addition, this method can be combined with other algorithms to allow more complicated simulation options. Some of these possibilities are listed here.

1. Within both the slave and master regions, additional mesh motion can be accommodated independent of the sliding motions. Mesh motion governed by the equations of elasticity [4, 10] are particularly useful in this context. With this type of mesh motion, small deformations of the object can be considered. Remeshing within each region would allow large deformations of the object.
2. Multiple sliding surfaces can be defined to allow more general motions. Rotational regions inside of sliding regions are easily defined as are multiple rotational regions.
3. The kinematics of the slave region do not always have to be predefined. The response of the solid region to fluid forces can be defined from Newton's law of motion and mesh motion developed from the resulting integration.
4. When the intervening fluid layer is absent, this approach is virtually the same as the mechanical contact algorithm without deformation. Object–object surface interactions are easily implemented as either continuity in the dependent variables or enforcement of a flux condition at the interface. In this case, object motions are arbitrary with contact, sliding and/or release being easily modelled.
5. In static mesh situations, the multipoint constraint method provides an effective method for joining non-contiguous meshes and thus simplifying the mesh generation process.
6. Also, in static situations, the multipoint constraint algorithm allows easy implementation of periodic boundary conditions.

Two primary developments are required for the general sliding mesh implementation: an efficient node-in-mesh search method and the construction and data handling of constraint equations. The search method was of primary importance since it is generally required to find the new location of the slave (sliding) nodes on the master (non-sliding) elements at every time step. For the sliding mesh case, the slave nodes and master elements are known lists that are generated from the problem input in the same way as surface boundary conditions. When object to object contact is considered, the kinematics will dictate which nodes and surfaces must be included in the search list and in some cases this may include the entire surface of each object. Because of the possible distribution of node and element data across many processors, this type of search requires careful consideration to ensure that a scalable algorithm is produced. The procedure used here to distribute the data and perform the node-in-element search was originally developed for the mechanical contact problem by Plimpton *et al.* [23]. The basic method was later adopted for use in parallel data transfer between multiple grids; this version is the basis for the present implementation. The method uses a recursive bisection method to organize the data and localize the search; Newton's method is used to find the

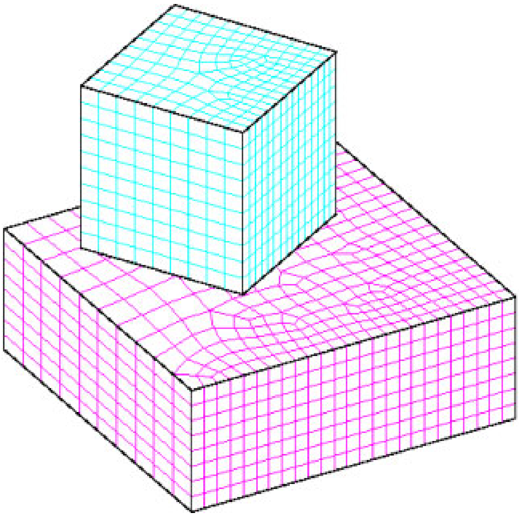

Figure 2. Schematic of non-contiguous meshing of contacting solids with partially covered/exposed element faces.

local co-ordinates of the slave node within an element. A rendezvous decomposition algorithm gathers slave nodes and master elements that are geometrically ‘close’ using a recursive co-ordinate bisection method due to Berger and Bokhari [24]. The original data transfer search and interpolation algorithm [23] was altered to provide additional data to each slave node owning processor. For each slave node, the master element number, element type, material identifier, nodal connectivity and local element co-ordinates for the slave node position in the element, were returned to the processor. As an option, the interpolated dependent variables at the slave node location could also be transferred; this was for use in implementing flux boundary conditions, such as thermal or electrical contact and fluid slip.

For applications involving object to object contact a second search may be required. This situation occurs when non-contiguous meshes are coupled, resulting in partially exposed element surfaces that have flux type boundary conditions applied. Figure 2 shows a heat conduction example where the horizontal top surface of the bottom block has partially exposed (covered) element faces. It is assumed that the exposed surfaces have a flux type boundary condition applied, such as a convection coefficient or black body radiation. A simple but effective method for boundary condition application involves determining which surface quadrature points are uncovered and will therefore participate in the integration of the surface boundary condition. By knowing the quadrature points for each element surface in the interface, the node (point)-in-element search can be used again to determine which quadrature points are exposed/covered. A more accurate, but more complex method for treating partially covered element surfaces was developed by Rashid [25]. The performance of the simple method proposed here is demonstrated in a later section. Note that when the partially exposed faces are part of a radiation enclosure, the problem is somewhat more difficult. An accurate view factor computation requires the irregular shaped face to be precisely described and an integration performed over the area.

The second part of the multipoint constraint/contact algorithm involves the development of the constraint equations and their implementation in the equation solver. For each degree of freedom to be constrained (interpolated) at a slave node, a constraint relation is constructed. Using the temperature as an example, a constraint condition may be expressed as

$$\mathbf{CT} = \mathbf{IT}_c + \mathbf{C}_k\mathbf{T}_k = \mathbf{F}_c \tag{3}$$

where $\mathbf{T}_c$ are constrained (slave) temperatures, $\mathbf{T}_k$ are 'known' nodal point temperatures from the master element, $\mathbf{I}$ is the identity matrix and $\mathbf{C}_k$ is the constraint coefficient matrix. The entries in $\mathbf{C}_k$ are the shape function interpolation values for the slave node location in the master element. The constraint relation in (3) can be implemented using either a Lagrange multiplier method or a penalty method. The approach used here is a penalty method that is written as

$$\mathbf{K}_c\mathbf{T} = \mathbf{C}^T\boldsymbol{\alpha}\mathbf{CT} = \mathbf{C}^T\boldsymbol{\alpha}\mathbf{F}_c = \mathbf{G}_c \tag{4}$$

where $\boldsymbol{\alpha}$ is a matrix of penalty coefficients and $\mathbf{K}_c, \mathbf{G}_c$ are the constraint arrays added to the global matrix system.

The penalty method was selected because it introduces no additional degrees of freedom and is very simple to implement in a general application code. Penalty constraints may increase the bandwidth of the global matrix but this is only significant for direct methods for solving the matrix problem. The entries in the $\mathbf{K}_c$ (or $\mathbf{C}$) matrix are known when the locations of the slave nodes in the master elements are determined. This data comes from the search procedure. The connectivity for the $\mathbf{K}_c$ matrix also comes from the search algorithm and forces the communication paths to be established dynamically at the solver level. A general interface for solver libraries [26] has been developed that accommodates the off-processor communications required by the application code when using constraints.

The remaining parameters to be specified in (4) are the penalty coefficients in the α matrix. The $\mathbf{K}_c$ matrix is added to the global matrix and should be dimensionally consistent with the overall equations. For the heat conduction case, the penalty coefficient should be defined according to $\boldsymbol{\alpha} \sim kh$ where k is the thermal conductivity for the slave material and h is a length scale for the mesh at the constraint location. The penalty coefficient for slave node i is most conveniently set for computations by defining $\alpha_i = SK_{ii}$ where S is a user defined scale factor and K_{ii} is the diagonal matrix entry for the global slave equation. Experience has shown that for single precision computations a scale factor of 10^1 to 10^3 is adequate to ensure satisfaction of the constraint without adversely affecting the matrix solver. Scaling of the diagonal maintains dimensional consistency even when the diffusion operator is augmented with advection terms.

For general use in multiphysics simulations, a few other issues must be addressed by the application code. In some cases there may be differences in mechanics phenomena across the multipoint constraint or contact interface. For example, if the constraint occurs at a fluid/solid boundary, the temperature will normally be constrained to be continuous while the velocity components will have a boundary condition applied. The search procedure must be able to recognize and record differences in materials or mechanics across the interface. A similar situation occurs when dependent variables are interpolated differently in different element types or different material regions. A node based continuous pressure approximation would imply the need for a constraint condition; a discontinuous pressure approximation would not require constraints on the pressure. Finally, for periodic boundary conditions a geometric mapping from the slave surface to the master surface must be specified so that the node-in-element search can be used without modification. In the present applications, only simple translations and rotations of planar periodic surfaces have been implemented. Note that the meshing for the master and slave surfaces in the periodic case need not be contiguous.

3. EXAMPLE SIMULATIONS

The multipoint constraint methods and sliding mesh procedures outlined above have been installed in several mechanics codes. These include a thermal analysis code [27], a viscous, incompressible fluid dynamics code [28] and a quasi-static electromagnetics code [29]. Example simulations have been completed on several demonstration problems using both single and multiple processor machines. Only thermal and fluid applications are illustrated here with the electromagnetic cases to be presented elsewhere.

3.1. Multiblock thermal conduction

The first example is a static thermal conduction problem involving multiple blocks attached to a thick plate as shown in Figure 3. The blocks of various geometries and thermal conductivities are all meshed independently and are then attached to the plate via the previously described multipoint constraint; trilinear brick elements are used throughout this example. The bottom of the plate is subjected to a specified heat flux ($q = 10.0$) or temperature ($T = 1000.0$) boundary condition. The top of each block loses energy to the surroundings through a convective boundary condition with specified heat transfer coefficient and reference temperature. The heat transfer coefficients are different for each block (see Figure 3) though the reference temperature is uniform ($T_{\text{ref}} = 400.0$). The exposed portion of the top of the plate is also

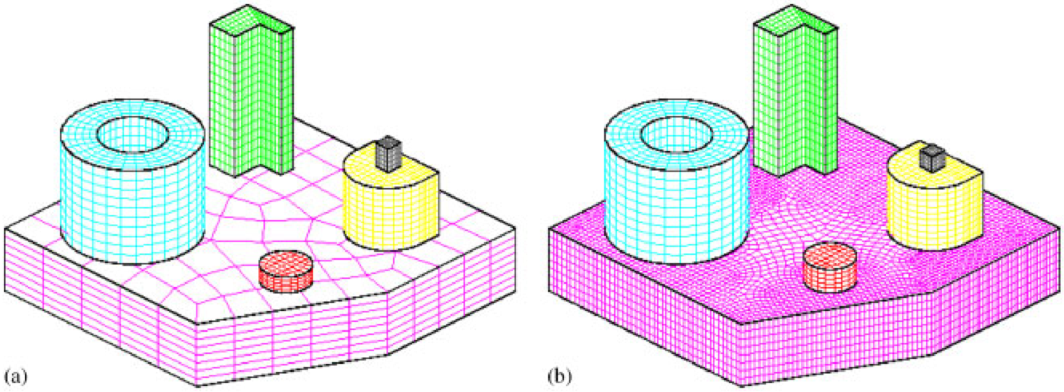

Figure 3. Mesh geometries for the thermal conduction problem with multiple blocks attached to a thick plate. Thermal properties for plate: $k = 0.06, h = 0.01$, annulus: $k = 0.12, h = 0.04$, L block: $k = 0.10, h = 0.01$, semicircular block: $k = 0.06, h = 0.004$, square block: $k = 0.01, h = 0.008$, disc: $k = 0.10, h = 0.008$: (a) non-contiguous mesh, coarse plate mesh; and (b) non-contiguous mesh, refined plate mesh.

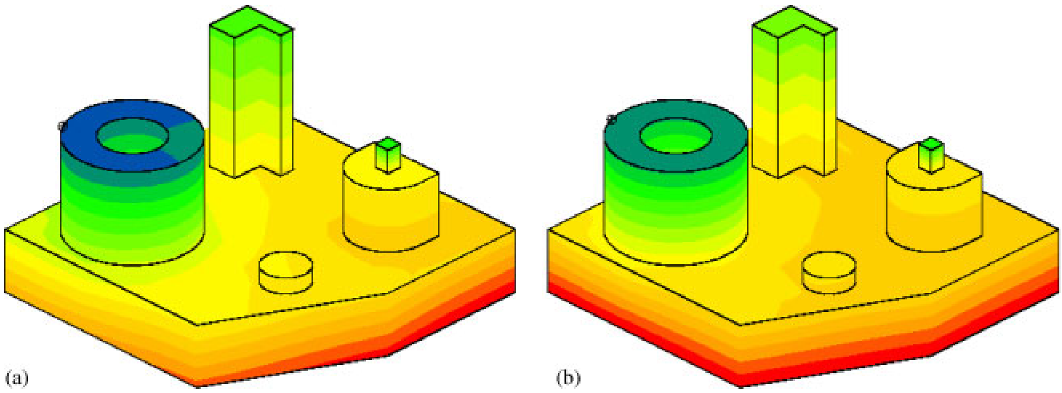

Figure 4. (a) Isotherms for non-contiguous mesh, applied flux to plate, $T_{\max} = 1787.0$, 20 contours; and (b) isotherms for non-contiguous mesh, fixed temperature on plate, $T_{\max} = 1000.0$, 20 contours.

subjected to a convection boundary condition. Because no boundary conditions are specified on the lateral surfaces of the components and plate, these surfaces are insulated. Several increasingly refined horizontal meshes for the plate were developed while holding the block meshes and vertical plate discretization constant, see Figure 3(b). A simple integration of the surface heat fluxes through the tops of the blocks and horizontal plate surfaces allowed the overall energy balance to be verified and the accuracy of the partially covered/exposed element algorithm to be evaluated. A refined, contiguously meshed geometry (not shown) was also produced to help verify the solution.

Figure 4 shows isotherm plots for the least refined of the plate meshes; the two plots correspond to the two different boundary condition cases of an applied flux and a fixed temperature. The continuity of the temperature between the blocks and the plates is easily seen. Continuity was also verified by comparing temperature profiles on the joined surfaces. The penalty form of the multipoint constraint provides an accurate implementation. With the plate and block conductivities as shown in Figure 3, penalty parameter scaling factors of $10 \leqslant \alpha \leqslant 1000$ could be used with no significant change in the solution. Shown in Table I is a compilation of the flux results with mesh refinement in the plate. The improved resolution of the exposed plate area leads to a better energy balance for the model, though even at the coarsest plate mesh the solution is quite accurate. Note that the primary benefit to using the multipoint constraint approach for meshing is the ease with which the geometry can be changed. In the present simulation, the locations of the blocks can be altered without remeshing and blocks added or subtracted. Such geometry changes using a contiguous mesh, especially with brick elements, is a significantly more complex process. Also, note that the finite element approximation and topology within each region could be varied independently without the need to employ transition elements.

3.2. Driven cavity

The second static example demonstrates the multipoint constraint algorithm with incompressible, viscous flow in the standard lid-driven cavity configuration. Shown in Figure 5 are three meshes used to solve the isothermal flow in a unit cavity at a Reynolds number of $Re = 100$.

Table I. Parameter variations with mesh refinement for the thermal conduction in blocks stacked on a plate.

Case	Mesh parameters for plate	Exposed surface area for plate (computed)	Heat flux— top of plate	Total energy transfer (% error)	$T_{\max}$ for the assembly
1	Non-contiguous— 67 surface elements	76.832	9.989	965.49 (2.98%)	1786.77
2	Non-contiguous— 258 surface elements	76.223	9.871	951.484 (1.49%)	1781.05
3	Non-contiguous— 1004 surface elements	76.401	9.865	951.630 (1.51%)	1779.38
4	Non-contiguous— 3645 surface elements	76.483	9.884	952.651 (1.62%)	1782.39
5	Contiguous— 3081 surface elements	76.447	9.841	949.507 (1.28%)	1780.61
Exact	—	76.465	—	937.50	—

Results are for the case of an applied heat flux.

For this simulation, each mesh region contains nine node, quadratic velocity elements from a mixed, finite element formulation. Both continuous, bilinear pressure (Q_2Q_1) elements and discontinuous, linear pressure (Q_2P_{-1}) elements were tested. Clearly, the Q_2P_{-1} element only requires constraints on the velocity components, while the Q_2Q_1 element has constraints on velocity and pressure. For each of the meshes shown, the two upper corner elements at the sliding lid were altered by moving the midside node on the vertical wall to the quarter point of the element edge. With a unit velocity specified on the cavity lid, this alteration results in 'closed' cavity. That is, the integral of the horizontal velocity over the vertical element surfaces is zero.

Typical stream function and velocity magnitude contour plots for each of the meshes are shown in Figures 6 and 7; these results are for the Q_2P_{-1} element. For the non-contiguous mesh cases, small offsets in the streamlines are visible since there is no constraint on the stream function; the stream function is computed by line integration of the velocity field around each element. The maximum values of the stream function are noted in Figure 6 and show excellent agreement among the various mesh configurations. The maximum error in the element mass balance (as determined from the stream function computation) is $\sim 10^{-9}$ and is insensitive to the mesh and penalty parameters used in the constraint. The velocity magnitude plots in Figure 7 show the expected continuity of the velocity components as enforced by the multipoint constraint. Figure 8 shows the vertical velocity component plotted along both sides of the mesh interface (two region mesh) for the case where the finer (slave) mesh is tied to the coarser (master) mesh region. Figure 9 shows the same profiles when the coarser (slave) mesh is tied to the finer (master) mesh. Though the profile is decidely less smooth in the second case, the overall field solution and element mass balances are very comparable to

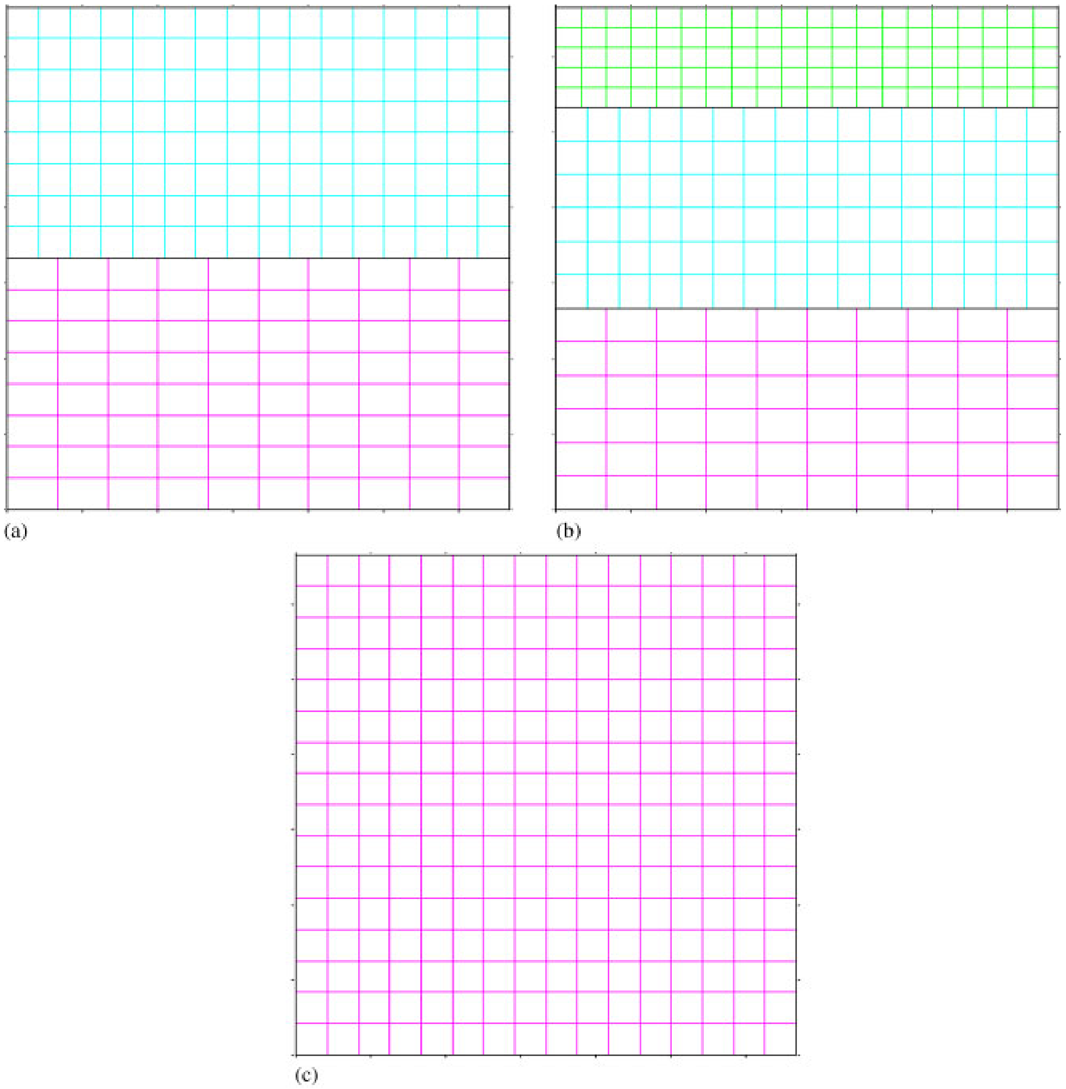

Figure 5. Meshes used for the lid-driven cavity problem: (a) two region, non-contiguous mesh; (b) three region non-contiguous mesh; and (c) contiguous mesh.

the smoother case from Figure 8 and the contiguous mesh result. The practice of tying the finer mesh to the coarser mesh is clearly recommended.

3.3. *Cylinder on a plate*

The simple problem of a end-heated cylinder sliding over a plate is used to demonstrate the repeated use of the node-in-mesh search and dynamics in the partially exposed element surface computation. A mesh for the initial state is shown in Figure 10 where the cylinder is stationary

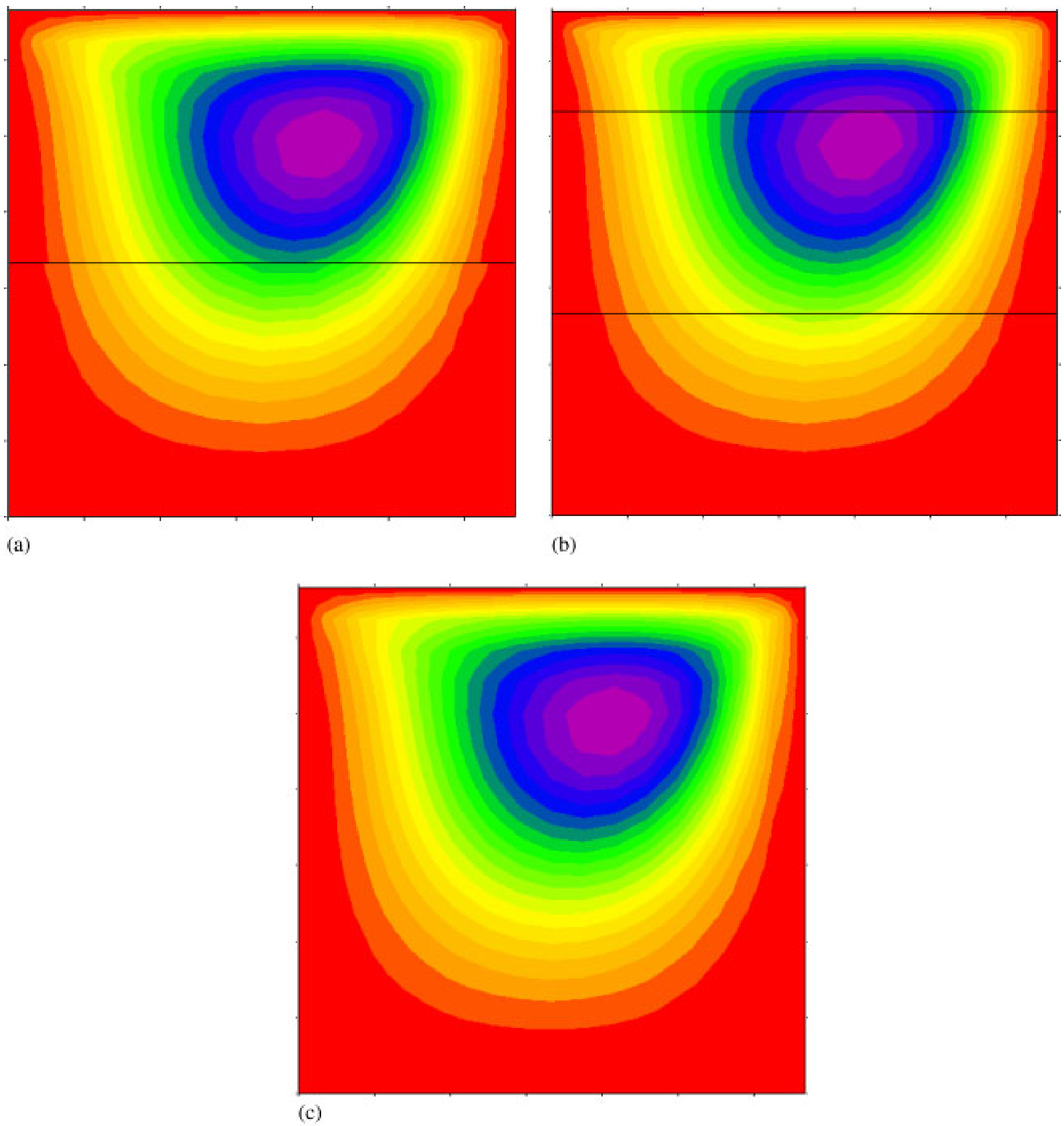

Figure 6. Stream function contours for lid-driven cavity at $Re = 100$: (a) two region mesh, $\Psi_{\max} = 0.10333$; (b) three region mesh, $\Psi_{\max} = 0.10326$; and (c) contiguous mesh, $\Psi_{\max} = 0.10331$.

and in contact with the plate. The top end of the cylinder is heated with a constant, spatially uniform surface flux. After a short time delay, the heated cylinder slides along the plate (no frictional heating) with a constant velocity and describes an L-shaped trajectory; the cylinder is lifted from the plate at the end of the horizontal motion and becomes motionless. The bottom of the plate is cooled by convection. The cylinder is 'tied' to the plate through a heat transfer or contact resistance coefficient. During each time step, each node of the contacting cylinder

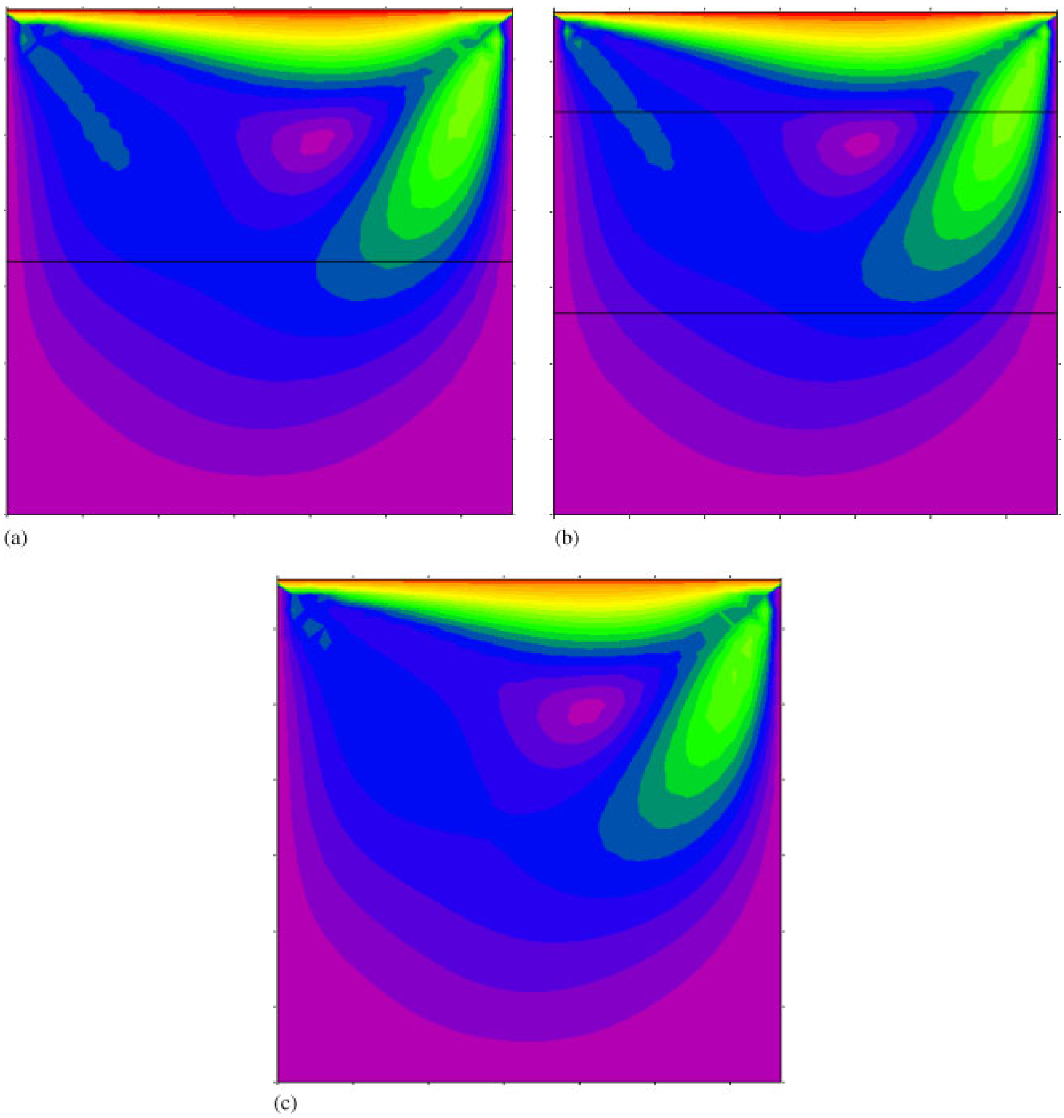

Figure 7. Velocity magnitude contours for lid-driven cavity at $Re = 100$: (a) two region mesh; (b) three region mesh; and (c) contiguous mesh.

face is located in the plate mesh and a plate or reference temperature found by interpolation for use in the cylinder boundary condition. Likewise, each node in the plate, covered by the cylinder footprint, is located in the cylinder; a cylinder or reference temperature is interpolated for the convection boundary condition applied to the plate. Quadrature points in the plate elements are also searched and flagged as covered or exposed.

The time-dependent thermal conduction problem is solved with the well-known implicit, predictor–corrector, integration algorithm due to Gresho *et al.* [30]. Multiple corrections within

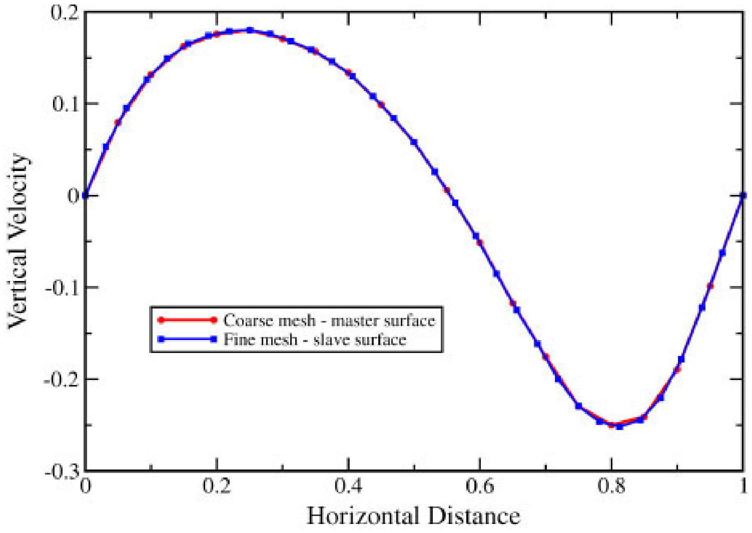

Figure 8. Vertical velocity profile along mesh interface for two region cavity. Coarse mesh is tied to fine mesh.

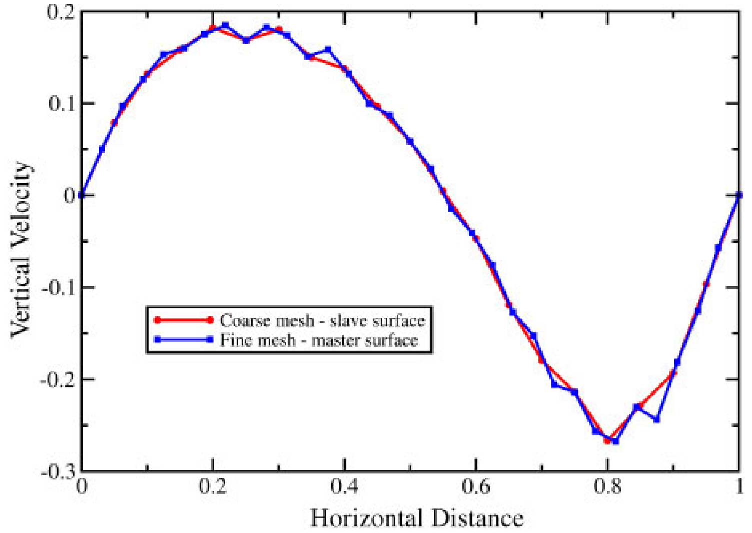

Figure 9. Vertical velocity profile along mesh interface for two region cavity. Fine mesh is tied to coarse mesh.

a time step are required to iterate the explicitly formed thermal contact condition and ensure equilibrium. An implicit formulation of the contact boundary condition could be developed but is more complex with a need for connectivity and matrix structure updating.

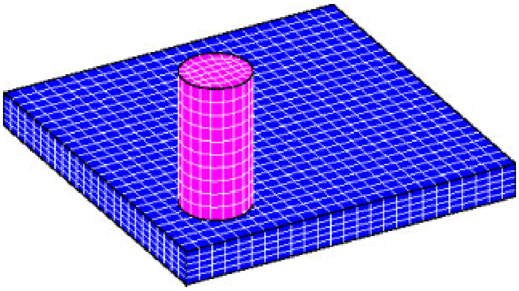

Figure 10. Mesh for heated cylinder sliding along a plate with a contact resistance. Cylinder is at the initial position.

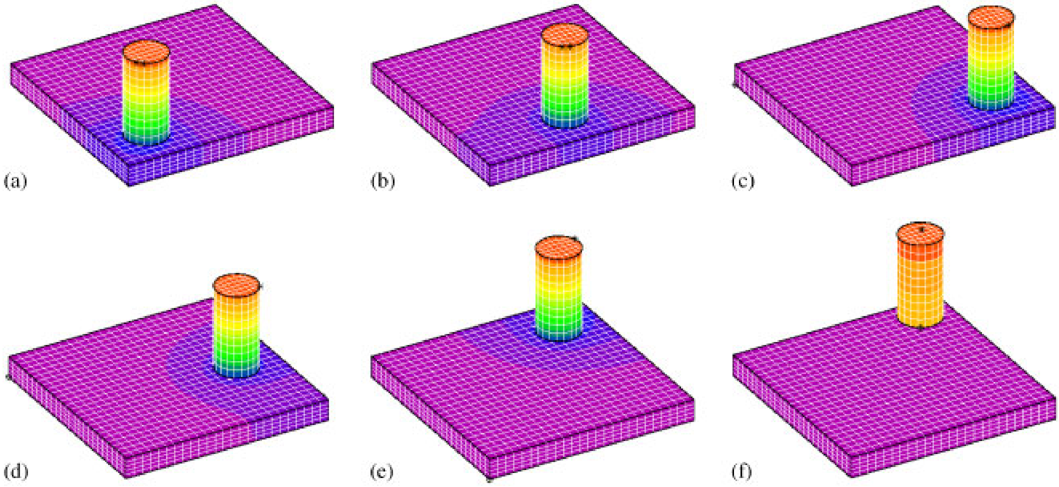

Figure 11. Isotherms at various times during the motion of a cylinder sliding on a plate. The cylinder has lifted from the plate in the last frame: (a) time = 4 s; (b) time = 11 s; (c) time = 19 s; (d) time = 25 s; (e) time = 34 s; and (f) time = 37 s.

A series of isotherm plots is shown in Figure 11 for various times during the motion of the cylinder. The isotherm levels are the same for each plot. The cylinder has lifted from the plate in the last frame, at which point the plate cools and the cylinder heats due to the now insulated condition on the bottom surface of the cylinder.

3.4. Rotating impeller

A moving mesh, viscous flow problem is the last example and simulates the flow in a simplified, two-dimensional impeller geometry as shown in Figure 12. The three-bladed impeller rotates at a constant angular velocity after a short, constant acceleration from rest. The upper inlet channel has a specified inflow velocity and the outlet boundary condition is a parallel flow with a uniform normal stress distribution. The problem is designed to illustrate the utility of the sliding mesh implementation and is not the analysis of a realistic flow device.

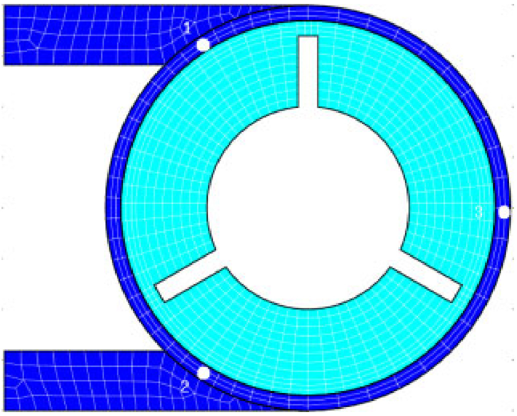

Figure 12. Mesh for simplified impeller. Fixed mesh region is dark grey; rotating mesh region is light grey.

A sliding surface is located at the mid-radius of the gap between the blade tip and the outer wall. In Figure 12, the mesh region with the darker shading is fixed and the inner, more lightly shaded mesh rotates with the blades. Note that multipoint constraint surfaces are also used to attach the channel regions to the fixed cylindrical mesh, thus reducing the complexity of mesh generation.

As a precursor to the more complex problem outlined above, a smooth annular gap problem was simulated to check the sliding mesh algorithm against a problem that could be solved with simple boundary conditions. The annular gap flow was solved on a contiguous mesh with a time-dependent tangential velocity specified on the outer cylinder wall; the inner cylinder was stationary. This solution was compared to a sliding mesh, ALE solution where the outer half of the gap mesh was attached to the outer wall and rotated with respect to the stationary inner mesh. The solutions for the two cases were essentially identical, though the ALE method required more computational effort. Both cases were integrated with an adaptive time step, predictor–corrector method [30] with the same integration tolerance and an allowance for multiple (Newton) corrector iterations. The contiguous mesh problem generally required one Newton correction per time step while the sliding mesh, ALE solution needed two Newton corrections per step and ran at a generally smaller time step. This is not unexpected, as the predictor solution is from the geometry of the previous step, which differs significantly from the corrected solution, and results in a smaller time step being selected for a given tolerance. A better method uses a reasonable fixed time step (after the start-up transient) that requires only two Newton corrections per step to ensure convergence. However, this requires some experimentation to select the appropriate time step.

The transient impeller flow was computed using the adaptive time-step integration algorithm with a one-step Newton corrector. The Q_2P_{-1} element was used throughout the mesh. The simulation was carried out for two revolutions of the impeller. Shown in Figure 13 are a series of instantaneous streamline plots superimposed on the updated mesh. The streamlines shown were initiated at the inlet plane and on the impeller blades; the colour map for the streamlines indicates the fluid speed. The first three timeplanes shown are just after the rotation begins

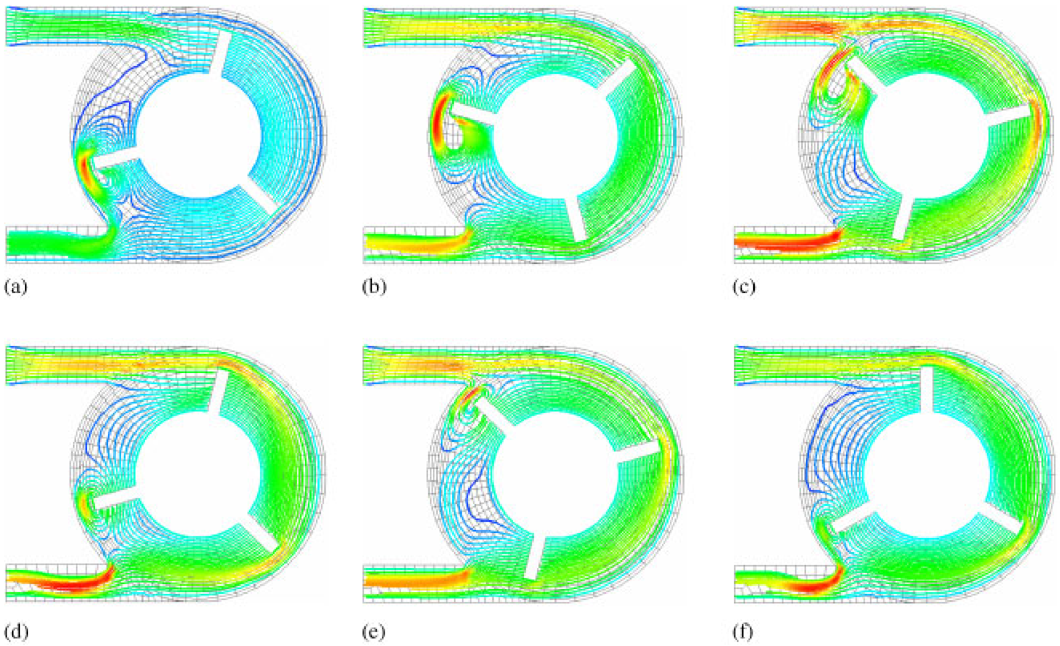

Figure 13. Streamlines at various times during the rotation of the impeller on the updated mesh: (a) time = 1.019 s; (b) time = 2.105 s; (c) time = 3.134 s; (d) time = 5.175 s; (e) time = 7.273 s; and (f) time = 8.888 s.

and a blade has not yet passed the inlet section. The remaining three frames are for times when an impeller blade is approaching and passing the inlet. At these times, the flow has already reached a periodic behaviour. The time-dependent nature of the flow can be better seen from time history plots of the velocity at various points in the field. Shown in Figure 14 are velocity magnitudes as a function of time for three locations; the selected points are labelled in Figure 12 and are at the mid-channel height where the inlet and outlet meet the cylindrical part of the mesh and at a mid-gap radius on the cylinder horizontal diameter. Data points are sampled every 25 time steps, which leads to some roughness in the time histories. The graph confirms the rapid establishment of a time periodic process.

4. CONCLUSIONS

The present work has demonstrated the feasability and utility of multipoint constraints for problems involving rigid body motions in fluid and thermal analysis. The implementation was also shown to be useful as a general meshing method through the use of non-contiguous mesh patches. The essential parts of the algorithm are an efficient, node-in-mesh search method and the use of penalty type constraints to enforce continuity of the primary variables. A simple method for considering partially covered/exposed element faces involved in boundary condi-

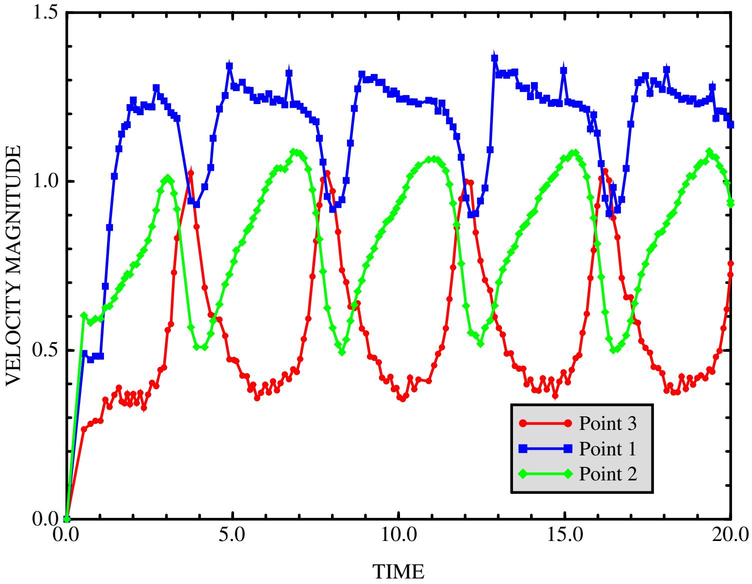

Figure 14. Time histories for three points within the impeller; point locations correspond to the labelled points shown in Figure 12.

tions was also outlined and demonstrated. Future reports will demonstrate the effectiveness of these methods for more complex non-isothermal analyses.

ACKNOWLEDGEMENTS

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin company, for the U.S. Department of Energy under contract DE-AC04-94AL85000.

REFERENCES

1. Tezduyar TE, Behr M, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure. I. The concept and preliminary tests. *Computer Methods in Applied Mechanics and Engineering* 1992; **94**:339–351.
2. Tezduyar TE, Aliabadi S, Behr M. Enhanced-discretization interface-capturing technique. In *Proceedings ISAC '97 High Performance Computing on Multiphase Flows*, Matsumoto Y, Prosperetti A (eds). Japan Society of Mechanical Engineers, 1997; 1–6.
3. Tezduyar TE, Aliabadi S, Behr M. Enhanced-discretization interface-capturing technique (EDICT) for computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:235–248.
4. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**:83–130.
5. Hughes TJR, Brooks A. A multi-dimensional upwind scheme with no crosswind diffusion. In *Finite Element Methods for Convection Dominated Flows*, Hughes TJR (ed.), vol. AMD-34. ASME: New York, 1979; 19–35.
6. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics* 1991; **28**:1–44.
7. Hirt CW, Nicholls BD. Volume of fluid (VOF) method for dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.

8. Sethian JA. *Level Set Methods and Fast Marching Methods*. Cambridge University Press: Cambridge, UK, 1999.
9. Lynch DR. Unified approach to simulation on deforming elements with application to phase change problems. *Journal of Computational Physics* 1982; **47**:387–411.
10. Cairncross RA, Schunk PR, Baer TA, Rao RR, Sackinger PA. A finite element method for free surface flows of incompressible fluids in three dimensions, Part I. Boundary fitted mesh motion. *International Journal for Numerical Methods in Fluids* 2000; **33**:375–403.
11. Löhner R, Yang C, Baum JD, Luo H, Pelessone D, Charman C. The numerical simulation of strongly unsteady flows with hundreds of moving bodies. *International Journal for Numerical Methods in Fluids* 1999; **31**: 113–120.
12. Cruchaga M, Celentano D, Tezduyar T. A moving Lagrangian interface technique for flow computation over fixed meshes. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:525–543.
13. Peskin CS, McQueen DM. A three-dimensional computational method for blood flow in the heart: I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics* 1989; **81**:372–405.
14. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
15. Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2004, in press.
16. Glowinski R, Pan T, Periaux J. A fictitious domain method for Dirichlet problem and application. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:283–303.
17. Bertrand F, Tanguy PA, Thibault F. A three-dimensional fictitious domain method for incompressible fluid flow problems. *International Journal for Numerical Methods in Fluids* 1997; **25**:719–736.
18. Bernardi C, Maday Y, Patera AT. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear Partial Differential Equations and Their Applications*, Brezis H, Lions J-L (eds). Longman Scientific & Technical: Harlow, UK, 1994; 13–51.
19. Gropp W, Lusk E, Skjellum A. *Using MPI*. MIT Press: Cambridge, MA, 1995.
20. Buffa A, Maday Y, Rapetti F. Calculation of eddy currents in moving structures by a sliding mesh-finite element method. *IEEE Transactions on Magnetics* 2000; **36**:1356–1359.
21. Tezduyar TE, Aliabadi S, Behr M, Johnson A, Kalro V, Litke M. Flow simulation and high performance computing. *Computational Mechanics* 1996; **18**:397–412.
22. Behr M, Tezduyar TE. Shear-slip mesh update method. *Computer Methods in Applied Mechanics and Engineering* 1999; **174**:261–274.
23. Plimpton S, Hendrickson B, Stewart J. A parallel rendezvous algorithm for interpolation between multiple grids. *Proceedings of ACM/IEEE SuperComputing98 Conference*, Orlando, FL, 1998.
24. Berger MJ, Bokhari SH. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers* 1987; **36**:570–580.
25. Rashid M. The arbitrary local mesh replacement method: an alternative to remeshing for crack propagation analysis. *Computer Methods in Applied Mechanics and Engineering* 1998; **154**:133–150.
26. Clay RL, Mish KD, Otero IJ, Taylor LM, Williams AB. An annotated reference guide to the finite element interface (FEI) specification, Version 1.0. *Sandia Report SAND99-8229*, Sandia National Laboratories, Albuquerque, NM, 1999.
27. Gartling DK, Hogan RE, Glass MW. COYOTE—A finite element computer program for nonlinear heat conduction problems. *Sandia Report SAND94-1173, SAND94-1179*, Sandia National Laboratories, Albuquerque, NM, 2003.
28. Gartling DK. KACHINA—A finite element computer program for viscous, incompressible flow problems. *Sandia Report*, Sandia National Laboratories, Albuquerque, NM, 2004, in preparation.
29. Gartling DK. TORO—A finite element computer program for nonlinear quasi-static problems in elctromagnetics. *Sandia Report SAND95-2472, SAND96-0903*, Sandia National Laboratories, Albuquerque, NM, 2003.
30. Gresho PM, Lee RL, Chan S, Sani RL. Solution of the time-dependent, incompressible Navier–Stokes and Boussinesq equations using the Galerkin finite element method. *Proceedings of IUTAM Symposium on Approximation Methods for Navier–Stokes Problems*, Paderborn, Germany, 1979.