INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 719–736 (1997)

A THREE-DIMENSIONAL FICTITIOUS DOMAIN METHOD FOR INCOMPRESSIBLE FLUID FLOW PROBLEMS

F. BERTRAND, P. A. TANGUY* AND F. THIBAULT

Department of Chemical Engineering, Ecole Polytechnique, PO Box 6079, Station Centre-ville, Montreal H3C 3A7, Canada

SUMMARY

A new Galerkin finite element method for the solution of the Navier–Stokes equations in enclosures containing internal parts which may be moving is presented. Dubbed the *virtual finite element method*, it is based upon optimization techniques and belongs to the class of fictitious domain methods. Only one volumetric mesh representing the enclosure without its internal parts needs to be generated. These are rather discretized using control points on which kinematic constraints are enforced and introduced into the mathematical formulation by means of Lagrange multipliers. Consequently, the meshing of the computational domain is much easier than with classical finite element approaches.

First, the methodology will be presented in detail. It will then be validated in the case of the two-dimensional Couette cylinder problem for which an analytical solution is available. Finally, the three-dimensional fluid flow inside a mechanically agitated vessel will be investigated. The accuracy of the numerical results will be assessed through a comparison with experimental data and results obtained with a standard finite element method. © 1997 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 719–736 (1997)

No. of Figures: 13. No. of Tables: 3. No. of References: 29.

KEY WORDS: finite element method; fictitious domain method; three-dimensional; internal parts; incompressible fluid flow; constrained optimization

1. INTRODUCTION

Computational fluid dynamics has proven over the years that it is a valuable asset in the prediction and understanding of viscous fluid flow phenomena. Nevertheless, reports on three-dimensional fluid flow simulations are not so numerous despite the fact that many physical phenomena just cannot be tackled adequately in two dimensions; this is quite the case for process engineering problems which in general involve three-dimensional flows in very complex geometries. Two factors come to mind when trying to explain this lack of three-dimensional numerical investigations.

First, the solution of three-dimensional fluid flow problems requires large amounts of both memory and CPU time. These drawbacks can, however, be alleviated by selecting suitable numerical techniques such as the preconditioned Uzawa algorithm[2,3] and multigrid methods.[4] As a result, the solution of huge systems of equations (10^5–10^6) can today be performed on desktop workstations with relative ease.

* Correspondence to: P. A. Tanguy, Department of Chemical Engineering, Ecole Polytechnique, PO Box 6079, Station Centre-ville, Montreal H3C 3A7, Canada.

CCC 0271–2091/97/060719–18 $17.50

Received 26 February 1996
Revised 25 October 1996

Next, it is known to anyone who has ever dealt with three-dimensional simulations that the meshing of the computational domain is in general cumbersome and time-consuming. To illustrate this fact, we have displayed in Figure 1 the mesh of a helical ribbon mixer typical of those used in polymerization reactors and fermenters. This mesh alone has necessitated many hours of work using I-DEAS, the mesh generator from SDRC. Fortunately, for this kind of problem a steady solution can be obtained if the computation is carried out using the impeller viewpoint.[5] As a result, a single mesh is required.

In the more general context of three-dimensional geometries containing arbitrarily shaped internal parts or bodies, such as in the example above, a few interesting strategies can be found in the literature, all of which aim at simplifying the mesh generation and the treatment of particular problems.

First, Thompson *et al.*[6] describe a method, based upon body-fitted co-ordinates, whereby an automatic mesh generation can be achieved for regions of rather general shapes containing any number of internal parts. Although originally designed for two-dimensional geometries, this method has since been extended to the three-dimensional case[7] and, because it may provide orthogonal or quasi-orthogonal meshes, is mostly used in conjunction with the finite difference and finite volume methods.

Also of interest is the work by Demirdzic and Peric,[8] who developed moving mesh techniques in which a single fixed mesh spanning both the stationary and moving parts of a geometry is allowed to move over time. Remeshing then becomes necessary when the grid gets too distorted.

To obviate the need for repetitive remeshings, Perng and Murthy[9] used an idea put forward by Rai[10] for the simulation of flow in mixing tanks. Called the sliding mesh technique, it makes use of two complementary grids, one which is fixed and which corresponds to the vessel itself, and another which is allowed to move over time and which characterizes the impeller. As a consequence, the grid

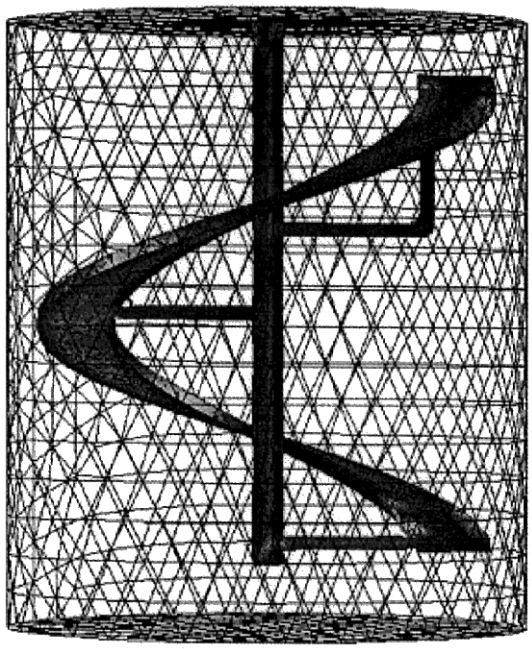


Figure 1. Standard finite element mesh for helical ribbon mixer

nodes do not coincide at the interface so that conservative interpolation is needed at each time step, which, from a practical point of view, is rather delicate. Moreover, extension of this method to complex cases such as those involving intermeshing impellers or multibody systems, for example, is not trivial.

Another approach, often used in practice, consists of ignoring some internal parts of the geometry when building the mesh. These are rather taken into account *a posteriori* as body forces in the Navier–Stokes equations. Along this line, Pelletier and Schetz[11] modelled the fluid flow behind a submarine propeller by mimicking the effect of the latter by equivalent forces inside a disc. These were calculated as a function of radius-varying thrust and torque values using empiricisms. Another example of such an approach is given by Hutchings *et al.*,[12] who modelled the impact of baffles in a mixing vessel upon fluid flow. Once again the baffles were not meshed as such but rather accounted for by introducing into the governing equations of change axisymmetric sinks of momentum at the tank walls. Consequently, steady state solutions could be obtained easily. Unfortunately, their technique proved valid only for simple geometries yielding negligible interactions between the impeller and the baffles.

To carry out simulations for blood flow in the heart, Peskin and McQueen[13] devised a technique called the immersed boundary method. In this method a fixed mesh is used and the moving bodies (the muscular heart walls) are approximated by a series of control points on which tension forces are imposed pointwise and distributed in the neighbouring elements. Here again these forces are not known *a priori* and are calculated using theoretical models. A variant of this technique, known as the immersed interface method, has been recently developed by Leveque and Li.[14]

All the above approaches apply to very specific types of problems and are more or less limited in scope. In other words, they all lack the generality and flexibility needed for widespread applicability.

In the early 1990s there have been a few reports on the use of fictitious domain methods for the solution of partial differential equations[15,16] in complex geometries. With these methods, domain boundaries are imbedded in some auxiliary domain of simple shape. As a consequence, the meshing tank is simplified significantly. However, this apparent simplification is accompanied by some complications such as the need to manage data structures pertaining to the actual geometry, an essential operation which for three-dimensional problems is in general far from being trivial.

Recently a new class of fictitious domain methods has been developed. It is based on the explicit use of Lagrange multipliers for the treatment of the internal parts of a geometry. These parts, which may be moving, are not meshed as such. Instead they are taken into account by means of a set of pointwise kinematic constraints that are coupled with the equations of change using a Lagrangian method.[17] Although similar to Peskin and McQueen's approach in as much as it is based upon control points for the characterization of the internal parts, this type of fictitious domain method does not resort to empirical forces. Velocity constraints are imposed on the control points through Lagrange multipliers so that only kinematics of the internal parts must be known *a priori*. Sheehy *et al.*[18] considered this approach for the numerical solution of an extrusion profile cooling problem by modelling the cooling channels as two-dimensional fictitious domains. Glowinski *et al.*[19] also demonstrated the efficiency of this type of method for predicting the three-dimensional fluid flow around a sphere imbedded in a rectangular domain and for solving an optimal shape problem involving two-dimensional Navier–Stokes flows.[20]

The objective of this paper is to introduce a new three-dimensional fictitious domain method for the solution of the Navier–Stokes equations in enclosures containing internal parts which may be moving.

First, the methodology will be presented in detail. It will then be validated in the case of the two-dimensional Couette cylinder problem for which an analytical solution is available. Finally, the strong potential of this method, dubbed the *virtual finite element method*, will be shown by means of

the solution of a typical industrial problem, that of the fluid flow inside a helical ribbon mixer. The accuracy of the numerical results will be assessed through a comparison with experimental data and results obtained with a standard finite element method.

2. EQUATIONS OF CHANGE

The flow of an incompressible fluid in a given geometry Ω with boundary Γ is governed by the classical Navier–Stokes equations

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \operatorname{grad} \mathbf{v}\right) + \operatorname{div} \tau + \operatorname{grad} p = \mathbf{f} \quad \text{in } \Omega, \tag{1}$$

$$\operatorname{div} \mathbf{v} = 0 \quad \text{in } \Omega, \tag{2}$$

where $\mathbf{v}$ is the velocity, p is the pressure, $\mathbf{f}$ is a body force and ρ is the density. The stress tensor τ is a function of the velocity field as expressed by a rheological equation of state

$$\tau = -2\eta(|\dot{\gamma}|)\dot{\gamma}, \tag{3}$$

where $\dot{\gamma} = \frac{1}{2}[\operatorname{grad} \mathbf{v} + (\operatorname{grad} \mathbf{v})^{\mathrm{T}}]$ is the rate-of-strain tensor. In this work the Newtonian model

$$\eta(|\dot{\gamma}|) = \mu \tag{4}$$

and the power-law model

$$\eta(|\dot{\gamma}|) = m|\dot{\gamma}|^{n-1} \tag{5}$$

are considered. The consistency and shear-thinning indices m and n characterize the fluid at hand and can be obtained by fitting viscosity versus shear rate experimental data.

Finally, equations (1) and (2) must be provided with appropriate initial and boundary conditions for mathematical well-posedness.

3. MATHEMATICAL FORMULATION

Let us consider for the moment the case of a steady state, inertialess Newtonian fluid flow problem so that equations (1) and (2) simplify to the Stokes equations

$$-\mu\Delta\mathbf{v} + \operatorname{grad} p = \mathbf{f} \quad \text{in } \Omega, \tag{6}$$

$$\operatorname{div} \mathbf{v} = 0 \quad \text{in } \Omega. \tag{7}$$

If we assume, without loss of generality, that Dirichlet boundary conditions in velocity are imposed on Γ, then it is well known[17] that for $\mathbf{v} \in [H_0^1(\Omega)]^3$ the solution of this problem is equivalent to that of the constrained minimization problem

$$\inf_{\substack{\mathbf{v} \in [H_0^1(\Omega)]^3 \\ \operatorname{div} \mathbf{v} = 0}} \frac{\mu}{2} \int_\Omega |\operatorname{grad} \mathbf{v}|^2 \, \mathrm{d}\Omega - \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}\Omega, \tag{8}$$

which can be further transformed into the saddle-point problem

$$\inf_{\mathbf{v} \in [H_0^1(\Omega)]^3} \sup_{p \in L^2(\Omega)} L(\mathbf{v}, p), \tag{9}$$

where

$$L(\mathbf{v}, p) = \frac{\mu}{2}\int_\Omega |\text{grad } \mathbf{v}|^2 \, d\Omega - \int_\Omega p\text{div } \mathbf{v} \, d\Omega - \int_\Omega \mathbf{f} \cdot \mathbf{v} \, d\Omega \tag{10}$$

is called a Lagrangian. In this last expression the pressure p is nothing but a Lagrange multiplier that enforces the incompressibility constraint. Moreover, the Euler equations characterizing the solution of this optimization problem constitute the variational form of the Stokes equations (6) and (7):

$$a(\mathbf{v}, \psi) - b(\psi, p) = (\mathbf{f}, \psi) \quad \forall \psi \in [H_0^1(\Omega)]^3, \tag{11}$$

$$b(\mathbf{v}, \phi) = 0 \quad \forall \phi \in L^2(\Omega), \tag{12}$$

where

$$a(\mathbf{v}, \psi) = \mu \int_\Omega \text{grad } \mathbf{v} \cdot \text{grad } \psi \, d\Omega, \tag{13}$$

$$b(\mathbf{v}, \phi) = \int_\Omega \phi \text{ div } \mathbf{v} \, d\Omega \tag{14}$$

and $(\cdot,\cdot)$ stands for the standard scalar product in $L^2(\Omega)$, i.e.

$$(u, v) = \int_\Omega uv \, d\Omega \quad \forall u, v \in L^2(\Omega). \tag{15}$$

A variant of the above method, often used for its efficiency, consists of adding to the standard Lagrangian $L(\mathbf{v}, p)$ a penalty term to enhance the imposition of the incompressibility constraint. The following so-called augmented Lagrangian results:

$$L_r(\mathbf{v}, p) = L(\mathbf{v}, p) + \frac{r}{2}\int_\Omega |\text{div } \mathbf{v}|^2 \, d\Omega, \tag{16}$$

where r is a penalty parameter.

One classical way of solving the saddle-point problem (10) is through the Uzawa algorithm.

0. Given $p^{(0)}$.
1. For $n = 0, 1, 2, \ldots$ until convergence:
   1.1. Solve for $\mathbf{v}^{(n+1)}$

$$a(\mathbf{v}^{(n+1)}, \psi) + r(\text{div } \mathbf{v}^{(n+1)}, \text{div } \psi) = (\mathbf{f}, \psi) + b(\psi, p^{(n)}) \quad \forall \psi \in [H_0^1(\Omega)]^3. \tag{17}$$

   1.2. Solve for $p^{(n+1)}$

$$(p^{(n+1)}, \phi) = (p^{(n)}, \phi) + \alpha b(\mathbf{v}^{(n+1)}, \phi) \quad \forall \phi \in L^2(\Omega). \tag{18}$$

The Uzawa algorithm can therefore be viewed as a means whereby a constrained problem is split up into a series of unconstrained problems. In practice, choosing $\alpha = r$ ensures convergence.

There exists no variational principle for the Navier–Stokes equations.[21] Consequently, a weak formulation for this problem cannot be obtained by minimizing a quadratic functional. Nevertheless, the Uzawa algorithm can be used formally in such a case after adding to equation (17) the advective and unsteady terms. In a similar fashion this equation can be modified to allow for non-linear viscosities. The reader is referred to the work by Tanguy *et al.*[22] for more details on this topic.

4. FICTITIOUS DOMAIN METHODS

Let us consider the steady laminar flow of a viscous incompressible fluid inside a bounded domain Ω containing a part Ω^* (Figure 2). Such a flow is governed by the Stokes equations to which Dirichlet boundary conditions are added without loss of generality:

$$-\mu\Delta\mathbf{v} + \operatorname{grad} p = \mathbf{f} \quad \text{in } \Omega a\backslash\Omega^*, \tag{19}$$

$$\operatorname{div} \mathbf{v} = 0 \quad \text{in } \Omega\backslash\Omega^*, \tag{20}$$

$$\mathbf{v} = 0 \quad \text{on } \Gamma, \tag{21}$$

$$\mathbf{v} = \mathbf{v}^* \quad \text{on } \Gamma^*. \tag{22}$$

Quite clearly, mesh generation may be cumbersome if the computational domain $\Omega\backslash\Omega^*$ is complex. One way to handle such a complexity consists of looking upon the part Ω^* as an obstacle to the flow by means of a kinematic constraint imposed on its boundary Γ^*. By analogy with the treatment of incompressibility, the Stokes problem (19)–(22) then becomes equivalent to the optimization problem

$$\inf_{\substack{\mathbf{v}\in[H_0^1(\Omega)]^3 \\ \mathbf{v}|_{\Gamma^*}=\mathbf{v}^*}} \sup_{p\in L^2(\Omega)} L_r(\mathbf{v}, p), \tag{23}$$

where $L_r(\mathbf{v}, p)$ is the augmented Lagrangian defined in (16). This is the essence of what are referred to as fictitious domain methods.

As before, this problem can be transformed into a saddle-point problem by introducing a Lagrange multiplier $\lambda \in [L^2(\Gamma^*)]^3$ to enforce the kinematic condition on Γ^*:

$$\inf_{\mathbf{v}\in[H_0^1(\Omega)]^3} \sup_{p\in L^2(\Omega)} \sup_{\lambda\in[L^2(\Gamma^*)]^3} L_{rs}^*(\mathbf{v}, p, \lambda), \tag{24}$$

where

$$L_{rs}^*(\mathbf{v}, p, \lambda) = L_r(\mathbf{v}, p) - \int_{\Gamma^*} \lambda \cdot (\mathbf{v} - \mathbf{v}^*)\, d\Gamma + \frac{s}{2}\int_{\Gamma^*} |\mathbf{v} - \mathbf{v}^*|^2\, d\Gamma \tag{25}$$

is the corresponding augmented Lagrangian and s is a penalty parameter.

This saddle-point problem can readily be solved by the Uzawa algorithm.

0. Given $p^{(0)}$ and $\lambda^{(0)}$.
1. For $n = 0, 1, 2, \ldots$ until convergence:

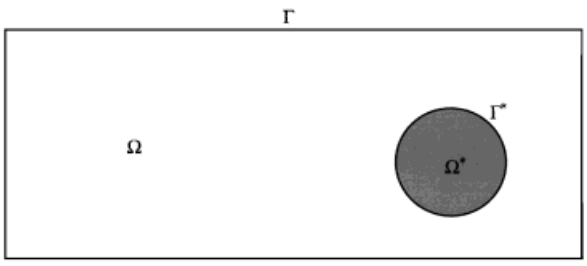


Figure 2. Two-dimensional schematic diagram representing bounded domain Ω and its internal part Ω^*

1.1. Solve for $\mathbf{v}^{(n+1)}$

$$a(\mathbf{v}^{(n+1)}, \psi) + r(\operatorname{div} \mathbf{v}^{(n+1)}, \operatorname{div} \psi) + s(\mathbf{v}^{(n+1)}, \psi)_{\Gamma^*} = (\mathbf{f}, \psi) + b(\psi, p^{(n)}) + (\lambda^{(n)}, \psi)_{\Gamma^*} + s(\mathbf{v}^*, \psi)_{\Gamma^*} \quad \forall \psi \in [H_0^1(\Omega)]^3. \tag{26}$$

1.2. Solve for $p^{(n+1)}$

$$(p^{(n+1)}, \phi) = (p^{(n)}, \phi) + \alpha b(\mathbf{v}^{(n+1)}, \phi) \quad \forall \phi \in L^2(\Omega). \tag{27}$$

1.3. Solve for $\lambda^{(n+1)}$

$$(\lambda^{(n+1)}, \xi)_{\Gamma^*} = (\lambda^{(n)}, \xi)_{\Gamma^*} + \beta((\mathbf{v}^{(n+1)} - \mathbf{v}^*), \xi)_{\Gamma^*} \quad \forall \xi \in [L^2(\Gamma^*)]^3. \tag{28}$$

In equations (26) and (28) the scalar product in $L^2(\Gamma^*)$ is defined as

$$(\lambda_1, \lambda_2)_{\Gamma^*} = \int_{\Gamma^*} \lambda_1 \lambda_2 \, \mathrm{d}\Gamma. \tag{29}$$

By analogy with the choice of α and r, we set $\beta = s$ in practice. We will come back to the selection of parameters r and s later.

The Euler equations characterizing the saddle-point problem (24) are given by

$$a(\mathbf{v}, \psi) = (\mathbf{f}, \psi) + b(\psi, p) + (\lambda, \psi)_{\Gamma^*} \quad \forall \psi \in [H_0^1(\Omega)]^3, \tag{30}$$

$$b(\mathbf{v}, \phi) = 0 \quad \forall \phi \in L^2(\Omega), \tag{31}$$

$$((\mathbf{v} - \mathbf{v}^*), \xi)_{\Gamma^*} = 0 \quad \forall \xi \in [L^2(\Gamma^*)]^3 \tag{32}$$

and constitute a mixed problem in $\mathbf{v}$, p and λ.

5. DISCRETIZATION

The discretization of equations (26)–(28) in the Uzawa algorithm is carried out using the Galerkin finite element method. Suitable approximation spaces must be selected for discrete variables $\mathbf{v}_h$, p_h and λ_h.

In this work, velocity and pressure are approximated by enriched tetrahedral elements P_1^+–P_0 (Figure 3) defined on the whole domain Ω and characterized by finite-dimensional subspaces

$$V_h = \{\mathbf{v}_h \in [C^0(\Omega)]^3;\ \mathbf{v}_h|_T \in [P_1^+(T)]^3,\ \forall T \in \mathscr{T}_h\}, \tag{33}$$

$$Q_h = \{q_h \in L^2(\Omega);\ q_h|_T \in P_0(T),\ \forall T \in \mathscr{T}_h\}, \tag{34}$$

where $\mathscr{T}_h$ is a regular tetrahedrization of Ω. The finite element P_1^+–P_0 is based on the classical linear element P_1–P_0. Extra degrees of freedom are added at the middle of each face to satisfy the Brezzi–Babuska condition in the case of the Stokes problem (6), (7).[23]

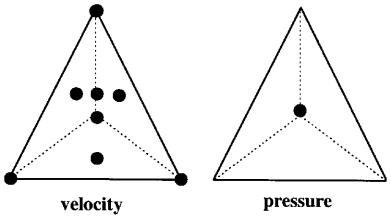


Figure 3. P_1^+–P_0 element

The finite-dimensional subspace $\Lambda_h \subset L^2(\Gamma^*)$ is related to the boundary Γ^* of the fictitious domain Ω^*. If $\mathscr{T}_h{}^*$ stands for a rectangular triangulation of Γ^*, then a natural choice for Λ_h is

$$\Lambda_h = \{\lambda_h \in [L^2(\Gamma^*)]^3;\ \lambda_h|_T \in [P_0(T)]^3,\ \forall T \in \mathscr{T}_h\}, \tag{35}$$

so that each of the three components of the Lagrange multiplier λ_h is approximated by piecewise constants.

At this stage one must realize that a stability condition which guarantees uniform convergence as $h \to 0$ is provided by an inf-sup condition which rules the selection of approximation spaces for the primitive variables. Unfortunately, little is known about the fulfilment of this condition in the context of fictitious domain methods, except for some indications given by Glowinski *et al.*[24] They state that, for the inf-sup condition to be satisfied, it is necessary that the distance between two adjacent nodes of $\mathscr{T}_h{}^*$ be larger than the mesh size of $\mathscr{T}_h$.

As clearly mentioned by Glowinski *et al.*,[19] an important issue related to the fictitious domain method lies in the ability to compute the boundary integrals appearing in equations (26) and (28). In practice these integrals can be computed numerically using Gaussian quadrature. Although this is not really a problem even for three-dimensional geometries, it requires that the velocity be interpolated at each Gaussian node. To carry out such an interpolation, the corresponding finite element of $\mathscr{T}_h$ must first be localized, which adds to the cost of the method. In this work we propose as an alternative a variant which we refer to as the *virtual finite element method*.

Let us consider a set of points $\{\mathbf{x}_i\}_{i=1}^N$, hereafter called control points, which discretize the fictitious boundary Γ^*. One possibility is then to enforce the kinematic constraint $\mathbf{v}|_{\Gamma^*} = \mathbf{v}^*$ pointwise:

$$\mathbf{v}(\mathbf{x}_i) = \mathbf{v}^*(\mathbf{x}_i) \quad \forall i = 1, 2, \ldots, N. \tag{36}$$

This way of imposing the velocity on Γ^* in a strong fashion is reminiscent of collocation methods. Constraints in (36) are equivalent, in the sense of distributions, to equation (32) if the components of ξ are Dirac functions defined as

$$\delta(\mathbf{x} - \mathbf{x}_i) = \begin{cases} \infty & \text{if } \mathbf{x} = \mathbf{x}_i, \\ 0 & \text{if } \mathbf{x} \neq \mathbf{x}_i. \end{cases} \tag{37}$$

Quite obviously this approach requires neither the computation of boundary integrals nor the generation of a surface mesh $\mathscr{T}_h^*$ for Γ^*. The fictitious domain Ω^* is delimited by a collection of control points. In this work each component of the Lagrange multiplier $\lambda \in [L^2(\Gamma^*)]^3$ is approximated using one constant per control point.

The matrix version of the Uzawa algorithm (26)–(28) follows.

0. Given $\mathbf{P}^{(0)}$ and $\mathbf{\Lambda}^{(0)}$.
1. For $n = 0, 1, 2, \ldots$ until convergence:
   1.1. Solve for $\mathbf{V}^{(n+1)}$

$$(\mathbf{A} + r\mathbf{B}^{\mathrm{T}}\mathbf{B} + s\mathbf{D})\mathbf{V}^{(n+1)} = \mathbf{F} - \mathbf{B}^{\mathrm{T}}\mathbf{P}^{(n)} + \mathbf{E}\mathbf{\Lambda}^{(n)} + \mathbf{G}. \tag{38}$$

   1.2. Solve for $\mathbf{P}^{(n+1)}$

$$\mathbf{P}^{(n+1)} = \mathbf{P}^{(n)} + \alpha\mathbf{B}\mathbf{V}^{(n+1)}. \tag{39}$$

   1.3. Solve for $\mathbf{\Lambda}^{(n+1)}$

$$\mathbf{\Lambda}^{(n+1)} = \mathbf{\Lambda}^{(n)} + \beta\mathbf{H}\mathbf{V}^{(n+1)}. \tag{40}$$

In these equations, **V**, **P** and $\mathbf{\Lambda}$ are vectors that represent the velocity, the pressure and the Lagrange multiplier respectively. **A** stands for the diffusion matrix, **B** stands for the divergence matrix and **F** accounts for the body force. As for **D**, **E**, **G** and **H**, they relate to the virtual finite element method and their respective meanings can be readily deduced from equations (26)–(28).

In practice the set of control points $\{\mathbf{x}_i\}_{i=1}^{N}$ can be obtained through a mesh generator or standard techniques such as octree-based or random shooting methods.[25]

The virtual finite element method can be extended to unsteady problems involving moving parts. The use of a time scheme allows the decomposition of such a problem into a series of steady state problems. Control points are time-dependent so that their location must be updated at each time iteration to account for the kinematics of the moving parts. This strategy has been used by the authors for the simulation of the flow inside a planetary mixer.[26]

The solution of the matrix systems (38)–(40) can be performed using either direct or iterative solvers. In the context of three-dimensional flow simulations the advantages of the latter methods are obvious in terms of both CPU time and memory requirements.[27]

The development of fast and reliable solvers to be used with the Uzawa algorithm is not trivial. For instance, it is well known that the convergence speed of the Uzawa algorithm increases with the value of penalty parameters such as r and s. In other words, these penalty terms originating from the augmented Lagrangian can be viewed as a preconditioner. Unfortunately, it can be proven[17] that the convergence speed of conjugate gradient solvers when solving equation (38) decreases with increasing values of these parameters. This property has led to the development of alternative preconditioning techniques.[2] Others have been successful in putting together an iterative solver based on the augmented Lagrangian method. Called the *incomplete Uzawa algorithm*, it is based on an optimal descent method for the solution of the linear system (38) and a conjugate gradient method for updating the pressure, as opposed to the descent method (39) in the classical Uzawa algorithm. Consequently, the parameter α is not fixed, its value being optimized at each iteration. The reader is referred to the article by Robichaud *et al.*[3] for more details. In this work the incomplete Uzawa algorithm was used for the computation of the velocity and pressure unknowns, while the Lagrange multiplier $\mathbf{\Lambda}$ was updated by means of a classical Uzawa iteration.

0. Given $\mathbf{P}^{(0)}$ and $\mathbf{\Lambda}^{(0)}$.
1. For $n = 0, 1, 2, \ldots$ until convergence:
   1.1. Get $\mathbf{P}^{(n+1)}$ and $\mathbf{V}^{(n+1)}$ using the incomplete Uzawa algorithm.
   1.2. Solve for $\mathbf{\Lambda}^{(n+1)}$

$$\mathbf{\Lambda}^{(n+1)} = \mathbf{\Lambda}^{(n)} + \beta \mathbf{H}\mathbf{V}^{(n+1)}. \tag{41}$$

As for the values of parameters r, s and β, one may take $r = 1$ and $\beta = s = 1$. With such a choice the linear system (38) remains well conditioned and the overall algorithm converges within a few iterations.

One will have noticed that step 1.2 in the previous algorithm is nothing but a descent method for the determination of the Lagrange multiplier $\mathbf{\Lambda}$. Alternatively, we could have developed an algorithm based on the conjugate gradient method as in Reference 24 and sought an efficient preconditioner as a replacement for the penalty parameter s. Such a procedure was not deemed essential within the scope of this paper.

6. VALIDATION OF THE VIRTUAL FINITE ELEMENT METHOD

In this section we assess the accuracy of the virtual finite element method and show its strong potential for solving three-dimensional fluid flow problems involving complex geometries. Two test

problems will be considered: the two-dimensional Couette cylinder problem with its analytical solution and the problem of three-dimensional fluid flow in a helical ribbon mixer for which experimental data are available.

Two-dimensional Couette cylinder problem

Let us consider the laminar flow of a viscous incompressible fluid between two coaxial cylinders of radii R and κR, the outer one of which is rotating with an angular velocity $\Omega_0 = 10$ rev min^{-1} (Figure 4). For this problem we set $R = 1$ m and $\kappa = 0{\cdot}5$.

The analytical solution can be obtained by integrating the Stokes equations in cylindrical co-ordinates:

$$v_\theta = \Omega_0 R \frac{\kappa R/r - r/\kappa R}{\kappa - 1/\kappa}. \tag{42}$$

This problem was solved with the virtual finite element method. The inner cylinder was considered as a fictitious domain and its boundary was accounted for through sets of control points. Velocity and pressure were discretized using the quadratic element P_2^+–P_1 of Crouzeix and Raviart[28] (Figure 5). Each component of the Lagrange multiplier λ was approximated with one constant per control point. Moreover, a maximum of two control points, i.e. two kinematic constraints, were allowed in each triangle of $\mathscr{T}_h$. Such a limitation is necessary for stability reasons and guarantees that the problem does not become overconstrained.

Simulations were carried out with four different meshes, the characteristics of which are summarized in Table I. The mesh and control points corresponding to the case $h = 0{\cdot}025$ are depicted in Figure 6.

Velocity vectors and streamlines obtained for $h = 0{\cdot}025$ are presented in Figure 7. These results comply with the analytical solution. The velocities inside the inner cylinder are negligible and the

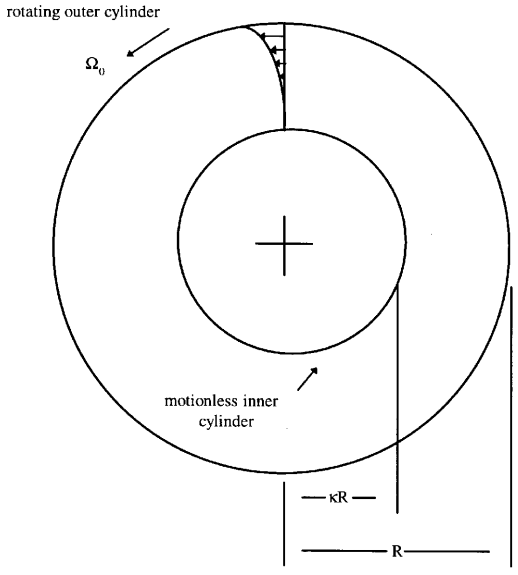


Figure 4. Couette cylinder problem

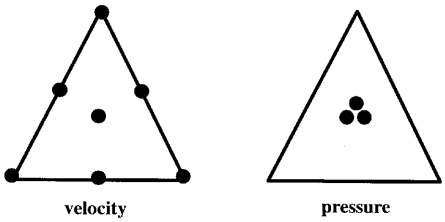


Figure 5. P_2^+–P_1 element

Table I. Characteristics of simulations for Couette cylinder problem

Mesh size (h)	Number of elements	Number of nodes	Number of control points	Number of velocity equations
0·1	697	1458	71	2664
0·05	2834	5795	137	11086
0·033	6230	12649	201	24546
0·025	11221	22694	264	44384

streamlines are concentric circles. Moreover, the distance between adjacent streamlines gets smaller when approaching the boundary of the outer cylinder, which means that the velocity gradients become larger, as expected.

The behaviour of the solution in the vicinity of the fictitious boundary can be appreciated in Figure 8. It clearly shows the strong ability of the proposed fictitious domain method even when just a few control points are used.

Finally, a graph of the velocity $L^\infty(\Omega)$-error with respect to the mesh size is presented in Figure 9. It shows that for this problem the virtual finite element method is stable and linear. It is, however, less

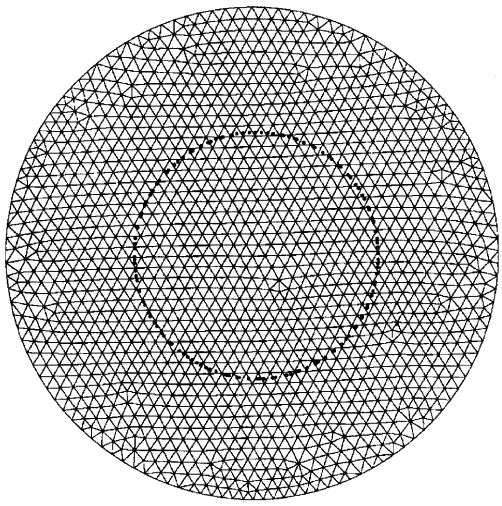


Figure 6. Mesh and control points for Couette cylinder problem ($h = 0{\cdot}025$)

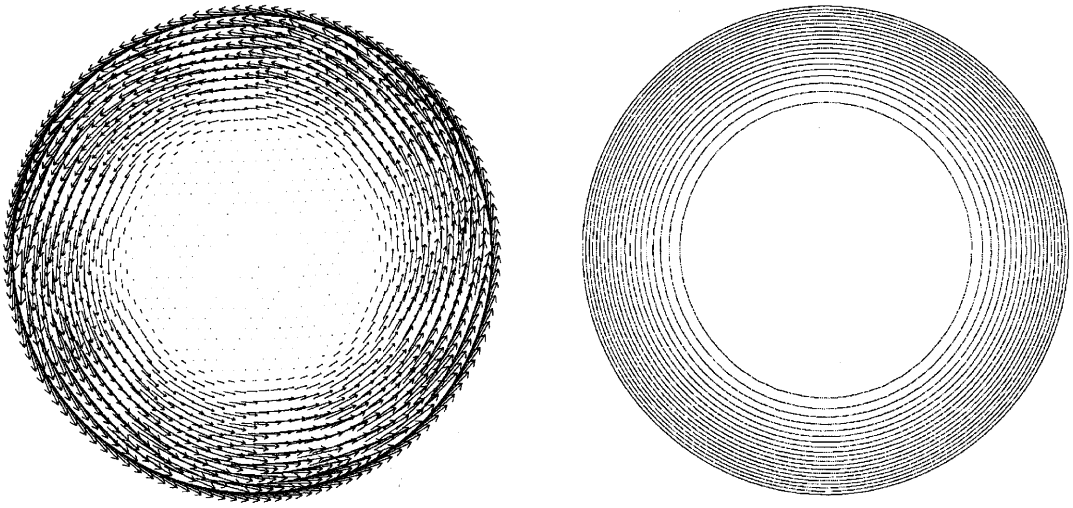


Figure 7. Velocity vectors and streamlines for Couette cylinder problem ($h = 0{\cdot}025$)

accurate, though easier to implement, than the fictitious domain method proposed by Glowinski *et al.*[24] For instance, choosing (35) in combination with Crouzeix–Raviart element would lead in all likelihood to a third-order-accurate method. A comparison between the two methods will be the subject of a forthcoming article.

Three-dimensional helical ribbon mixer problem

Let us consider the mixing system described in Reference 29 and shown in Figure 10. It consists of a cylindrical vessel (8 l) provided with a helical ribbon (HR) impeller set in motion at a rotating speed of 20 rev min^{-1}. The vessel is filled with a highly non- linear power-law fluid with parameters $m = 3{\cdot}82$ Pa s^n and $n = 0{\cdot}16$ in equation (5). Its density is 1000 kg m^{-3}.

For the solution of this problem it was decided to consider the viewpoint of an observer located on the moving impeller. As explained by Tanguy *et al.*,[5] the use of this Lagrangian frame of reference makes the imposition of the boundary conditions much easier on a standard finite element mesh. In a

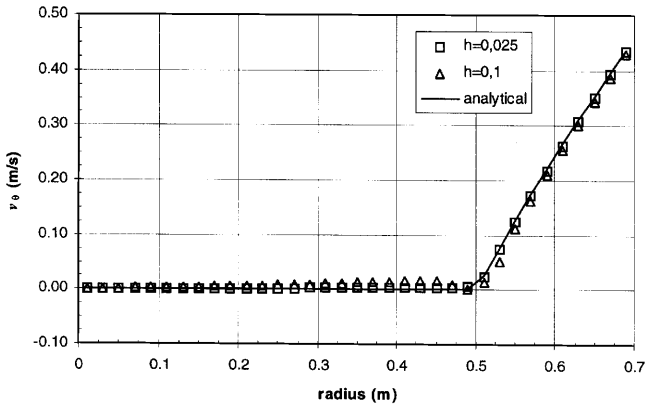


Figure 8. Graph of angular velocity versus radius

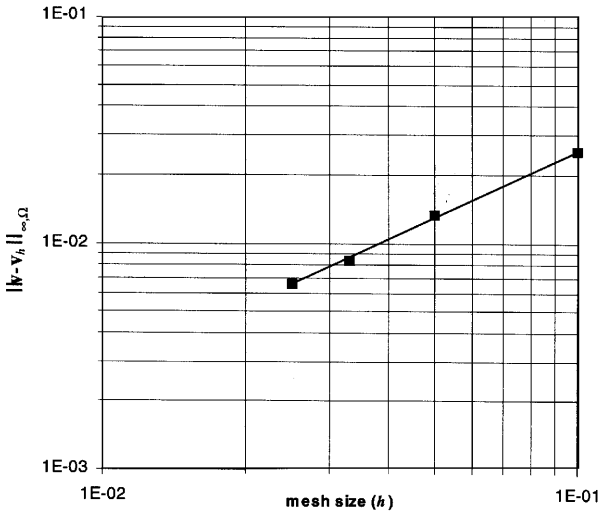


Figure 9. Graph of velocity $L^{\infty}(\Omega)$-error versus mesh size h

Eulerian frame of reference one would have to deal with moving boundary conditions so that the problem would become unsteady in nature, calling for a costly remeshing of the computational domain at each time step. However, the situation would not be so intricate with the proposed fictitious domain method. As said before, moving boundaries could be tackled quite naturally by considering time-dependent Lagrange multipliers and by allowing the control points to evolve over time according to the impeller kinematics.

In the Lagrangian viewpoint the boundary conditions are as follows:

(a) a no-slip condition on the impeller $\mathbf{v}^*(\mathbf{x}_i) = 0$, $\forall i = 1, 2, \ldots, N$

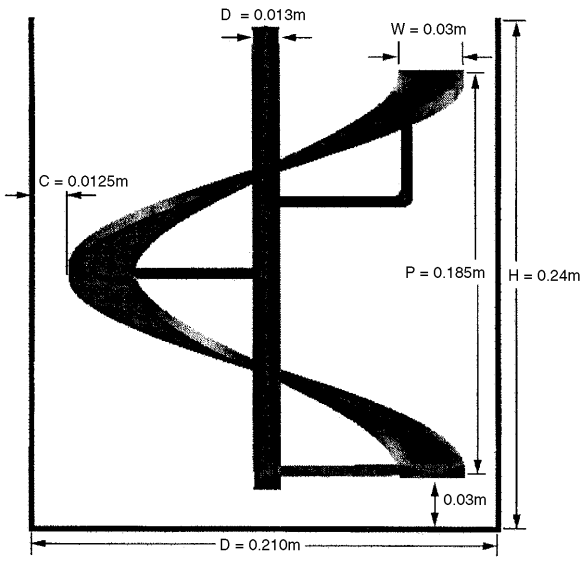


Figure 10. Geometric description of helical ribbon mixer

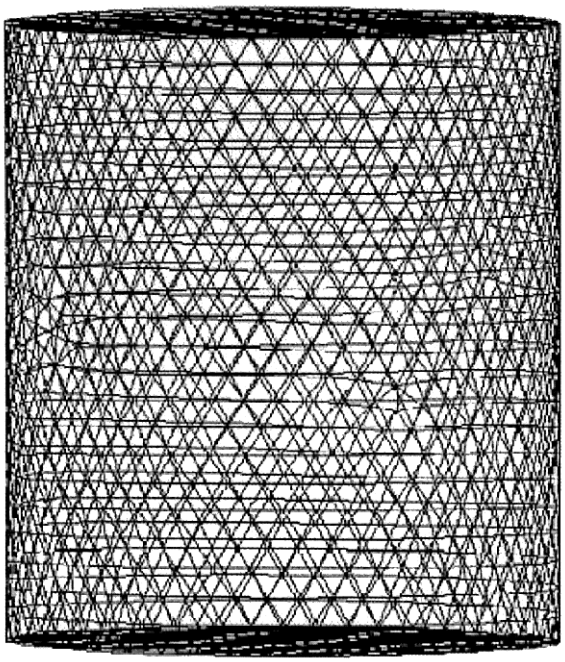


Figure 11. Finite element mesh of vessel

(b) the rotational speed (20 rev min^{-1}) on the vessel walls
(c) free surface conditions (the surface is considered as flat).

As the frame of reference is non-Galilean (the observer is in rotation), the Navier–Stokes equations must be complemented by the centrifugal and Coriolis forces. Finally, using this frame of reference, the fluid flow can be treated at steady state.

From a numerical standpoint, two meshes were considered: one for the standard finite element method and one for the virtual finite element method, the latter being a mesh of the vessel only (Figures 1 and 11 respectively). In this case two different surface representations of the HR impeller corresponding to two sets of control points were used for the simulations. Moreover, for reasons mentioned previously, a maximum of two control points were permitted in each tetrahedron of $\mathscr{T}_h$. Relevant numbers relating to these meshes are summarized in Table II.

The numerical simulations were performed on an IBM RISC/6000 590 workstation. A loading strategy was used for the treatment of the non- linearities inherent to the power-law model. It consists of carrying out a first simulation at $n = 1{\cdot}00$ and then using the result obtained as an initial guess for a

Table II. Characteristics of simulations for helical ribbon mixer problem

Mesh	Number of elements	Number of nodes	Number of velocity equations	Number of control points
Standard	47195	105383	299146	—
Fictitious 1	37481	84000	237487	1186
Fictitious 2	37481	84000	237487	1817

Table III. Results for circulation time

Type of results	Circulation time (s)
Standard	77
Fictitious 1	152
Fictitious 2	86
Experimental	87 (± 8)

second simulation at a lower value of n. This procedure is repeated until the target value ($n = 0{\cdot}16$) is reached. In this work each simulation required 5 h of CPU time with the standard finite element method and 10 h with the virtual finite element method. In the latter case about 50 Uzawa iterations were needed for convergence.

The results obtained with the fictitious domain method will now be compared with those obtained experimentally and through a standard Galerkin finite element method.

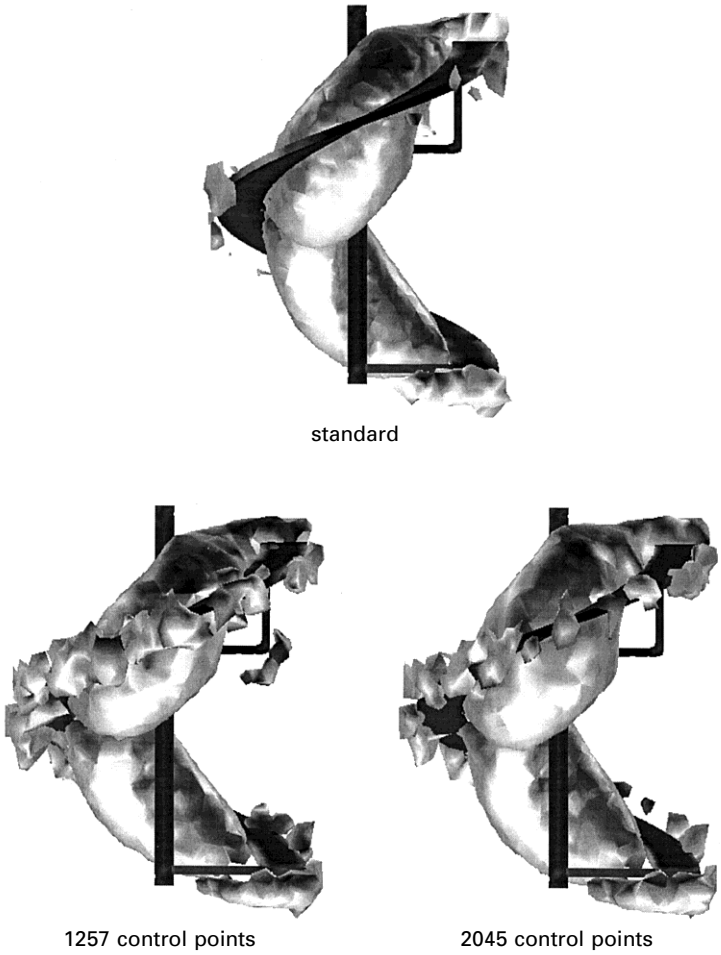
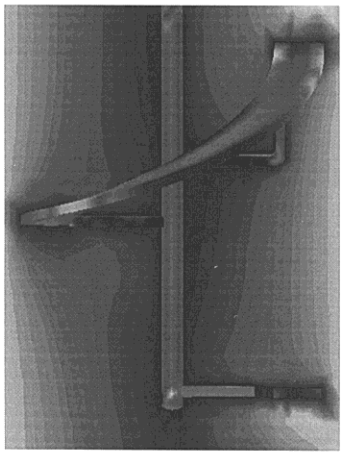


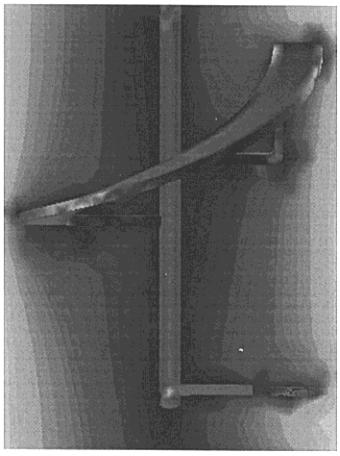
Figure 12. Pumping tubes in helical ribbon mixer

First, values of the circulation time are presented. Experimental measurements of circulation times were obtained by using a thermal technique, as explained by Brito-De La Fuente.[29] From a numerical standpoint, values of the circulation time were obtained as in a previous paper,[5] i.e. by integrating the velocity of a neutrally buoyant particle so as to get a graph of its position with respect to time. In the present case, particles were injected at the fluid surface. Results for the circulation time are presented in Table III.

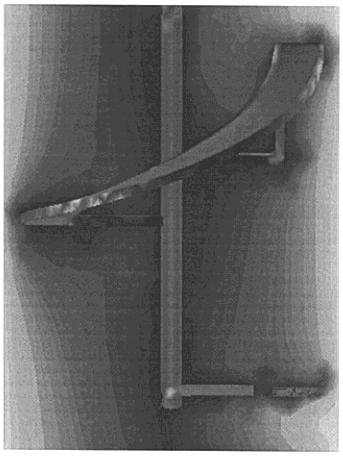
One may readily notice that the number of control points has a strong impact on the values of the circulation time. While the values obtained with the standard mesh and the finer fictitious domain mesh (fictitious 2) are right on target, the same cannot be said about the coarser fictitious domain mesh (fictitious 1), whose computed value of the circulation time is way above what is measured experimentally. Although this may seem rather surprising at first sight, an explanation of this phenomenon can be found in Figure 12, which characterizes the so-called pumping tubes for the three meshes and within which the axial component of the velocity is negative. It appears that the volume



standard



1257 control points



2045 control points

Figure 13. Contour plots of magnitude of velocity field for helical ribbon mixer problem

of the tube obtained with the coarser fictitious domain mesh (fictitious 1) is underestimated (especially in the lower part) with respect to that predicted with the standard mesh. The relatively small number of control points for this mesh means that the impeller is probably not as well approximated and that it is somewhat porous. Consequently, it is not surprising that this particle takes more time to come back to its starting point in this case.

Finally, Figure 13 exhibits contour plots of the magnitude of the velocity field on a cross-section. It clearly evidences the quality of the solutions obtained with the virtual finite element method.

7. CONCLUSIONS

The main objective of this paper was to present a new finite element method for the analysis of incompressible flow problems in enclosures containing internal (moving) parts. Belonging to the class of fictitious domain methods, its main advantage lies in the fact that only a mesh that represents the enclosure needs to be generated. The internal parts are rather accounted for by a set of kinematic constraints that are formulated in the classical Navier–Stokes equations using an augmented Lagrangian method. Simulations for the two-dimensional Couette cylinder problem and the three-dimensional helical ribbon mixer problem showed that, provided that a sufficient number of control points are used, the virtual finite element method can be used as an advantageous alternative to the standard Galerkin finite element method.

ACKNOWLEDGEMENT

This paper is based on a presentation at the SIAM 1994 Annual Meeting.[1]

REFERENCES

1. F. Bertrand, P. A. Tanguy and F. Thibault, 'A new finite element method for viscous flow problems in enclosures containing moving parts', *Proc. SIAM Ann. Meet.*, San Diego, CA, SIAM, Philadelphia, PA, 1994.
2. J. Cahouet and J. P. Chabard, 'Some fast 3D finite element solvers for the generalized Stokes problem', *Int. j. numer. methods fluids*, **8**, 869–895 (1988).
3. M. P. Robichaud, P. A. Tanguy and M. Fortin, 'An iterative implementation of the Uzawa algorithm for 3D fluid flow problems', *Int. j. numer. methods fluids*, **10**, 429–442 (1990).
4. R. Verfürth, 'A combined conjugate gradient–multigrid algorithm for the numerical solution of the Stokes problem', *IMA J. Numer. Anal.*, **4**, 441–455 (1984).
5. P. A. Tanguy, R. Lacroix, F. Bertrand, L. Choplin and B. Brito-De La Fuente, 'Finite element analysis of viscous mixing with an helical ribbon-screw impeller', *AIChE J.*, **38**, 939–944 (1992).
6. J. F. Thompson, F. C. Thames and C. W. Mastin, 'Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **15**, 299–319 (1974).
7. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation*, *Formulation and Applications*, North-Holland, Amsterdam, 1985.
8. I. Demirdzic and M. Peric, 'Finite volume method for the prediction of fluid flow in arbitrarily shaped domains with moving boundaries', *Int. j. numer. methods fluids*, **10**, 771–790 (1990).
9. C. Y. Perng and J. Murthy, 'A moving mesh technique for the simulation of flow in mixing tanks', *Proc. AIChE Ann. Meet.*, Miami Beach, FL, AIChE, 1992.
10. M. M. Rai, 'Navier–Stokes simulations of rotor–stator interaction using patched and overlaid grids', *AIAA Paper 85-1519*, 1985.
11. D. H. Pelletier and J. A. Schetz, 'Finite element Navier–Stokes calculation of three-dimensional turbulent flow near a propeller', *AIAA J.*, **24**, 1049–1416 (1986).
12. B. J. Hutchings, B. R. Patel and R. J. Weetman, 'Computational prediction of flow fields in mixing tanks with experimental verification', *Proc. ASME Ann. Winter Meet.*, San Francisco, CA, ASME, 1989, New York, 1989.
13. C. S. Peskin and D. M. McQueen, 'A three-dimensional computational method for blood flow in the heart: I. Immersed elastic fibers in a viscous incompressible fluid', *J. Comput. Phys.*, **81**, 372–405 (1989).
14. R. J. Leveque and Z. Li, 'The immersed interface method for elliptic equations with discontinuous coefficients and singular sources', *SIAM J. Numer. Anal.*, **31**, 1019–1044 (1994).

15. D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samanth and J. E. Bussoletti, 'A locally refined finite rectangular grid finite element method. Application to computational physics', *J. Comput. Phys.*, **92**, 1–66 (1991).
16. J. E. Bussoletti, F. T. Johnson, S. S. Samanth, D. P. Young and R. H. Burkhart, 'EM-TRANAIR: steps toward solution of general 3D Maxwell's equations', in R. Glowinski (ed.), *Computer Methods in Applied Sciences and Engineering*, Nova Science, New York, 1991, pp. 49–72.
17. M. Fortin and R. Glowinski, *The Augmented Lagrangian Method*, North-Holland, Amsterdam, 1983.
18. P. Sheehy, P. A. Tanguy and D. Blouin, 'Finite element model for complex profile calibration', *Polym. Eng. Sci.*, **34**, 650–655 (1994).
19. R. Glowinski, T. Pan and J. Périaux, 'A fictitious domain method for Dirichlet problem and applications', *Comput. Methods Appl. Mech. Eng.*, **111**, 283–303 (1994).
20. R. Glowinski, T. W. Pan, A. J. Kearsley and J. Périaux, 'Numerical simulation and optimal shape for viscous flow by a fictitious domain method', *Int. j. numer. methods fluids*, **20**, 695–711 (1995).
21. J. Leray, 'Essai sur les mouvements plans d'un liquide visqueux que limitent des parois', *J. Math. Pures Appl.*, **13**, 331–418 (1934).
22. P. A. Tanguy, M. Fortin and L. Choplin, 'Finite element simulation of dip coating. II: Non-Newtonian fluids', *Int. j. numer. methods fluids*, **4**, 441–458 (1984).
23. F. Bertrand, M. Gadbois and P. A. Tanguy, 'Tetrahedral elements for fluid flow problems', *Int. j. numer. methods eng.*, **33**, 1251–1267 (1992).
24. R. Glowinski, T. Pan and J. Périaux, 'A fictitious domain method for external incompressible viscous flow problems modelled by Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **112**, 133–148 (1994).
25. P. L. George, *Génération Automatique des Maillages*, Masson, Paris, 1990.
26. P. A. Tanguy, F. Bertrand, R. Labrie and E. Brito-De La Fuente, 'Numerical modeling of viscoplastic slurries in a twin-blade planetary mixer', *Trans. IChemE*, **74A**, 499–504 (1996).
27. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems*, Academic, New York, 1994.
28. M. Crouzeix and P. A. Raviart, 'Conforming and non-conforming finite element methods for solving the stationary Stokes equations', *RAIRO Anal. Numér.*, **7**, 33–76 (1973).
29. E. Brito-De La Fuente, *Ph.D Thesis*, Laval University, 1992.

