

THREE-DIMENSIONAL ADAPTIVE FINITE ELEMENT COMPUTATIONS AND APPLICATIONS TO NON-NEWTONIAN FLUIDS

H. H. DANNELONGUE AND P. A. TANGUY*

Department of Chemical Engineering and CERSIM, Université Laval, Québec G1K 7P4, Canada

SUMMARY

An adaptive strategy for the finite element solution of three-dimensional viscous flow problems is defined and implemented. The solution strategy is based on an advancing front mesh generator making use of binary data structures for fast geometrical data handling. The error is estimated *a posteriori* with a residual-type bound. The error estimate is shown to exhibit proper convergence for tetrahedral elements. Its combination with the mesh generator and an interpolation scheme for unstructured meshes is shown to generate adaptive meshes and to reduce the solution cost for a given error level, as illustrated by the isothermal flow of a shear-thinning fluid.

KEY WORDS Finite elements Meshing Adaptivity Navier–Stokes Power law Error estimation

1. INTRODUCTION

Computational fluid dynamics is gaining acceptance as a valuable tool for engineers in a wide range of fields. In aerodynamics, the field which has successfully pioneered such numerical methods, the complexity of mathematical models has grown with the increase in computer speed and memory size¹ and now allows the 3D modelling of the flow past an entire aircraft. As the discipline matures, additional topics besides the eternal quest to minimize computational time have come to the forefront of development:² the estimation of the solution accuracy and the reduction of the engineer's involvement in the drudgery of the meshing process.

The development in the field of material forming—to which the results contained in this paper are applied—has been similar, engineers relying nowadays on computer simulation during design. The emphasis is now put on the areas identified above: reduction of computational cost, knowledge of solution accuracy and better use of the engineer's time by integration of the finite element code with a solid modeller.

In order to answer these requirements globally, adaptive techniques were developed for 2D elasticity problems. In these methods a solution is computed on a user-defined mesh; the error is then estimated from that solution³ and the mesh refined accordingly in areas of high error. This leads to a minimum number of degrees of freedom for a chosen accuracy and reduces the engineer's work load if the refinement is done automatically. These methods are implemented in available codes using structured meshes (PLTMG,⁴ FEARS⁵), but no 3D extension has been published.

* Author to whom correspondence should be addressed.

More recently, an adaptive method based on triangular unstructured grids has been developed for a 2D aerodynamics problem, the solution of Euler's equation.⁶ It has been applied since to linear elasticity problems⁷ as well as to forming processes.⁸ Moreover, its use has been extended to 3D Euler computations,⁹ showing the potential of unstructured grids for complex geometries.

The objective of the present paper is to show how a 3D unstructured mesh generator combined with a rigorous *a posteriori* error estimate¹⁰⁻¹² allows the creation of an adaptive solution scheme for the diffusion-dominated Navier-Stokes equations. This adaptive scheme is an answer to the challenges stated above. Section 2 of the paper details the features of the mesh generator. The relationship between an object and its discretization is dealt with¹³ as well as other issues having a strong bearing on the quality of the final mesh.^{14, 15} Section 3 details the equations to be solved and their finite element approximation and solution.^{16, 17} This is followed by a study of the error estimate and the adaptive cycle itself (Section 4). Finally, in Section 5, the solution process is applied to a classical polymeric fluid flow problem, the low-Reynolds-number flow of a non-Newtonian fluid in a contraction.

2. MESHING ALGORITHM

2.1. Advancing front mesh generator

Unstructured mesh generators have proved very versatile and simple to implement in 2D and their use has grown rapidly in commercial software. Their success is based on the simplicial rule, i.e. that triangles can mesh any polygon and that tetrahedra can mesh any polyhedron. The general principles of the advancing front method of unstructured mesh generation, pioneered by Lo,¹⁸ are briefly described with the help of Figure 1. The boundary of the domain is first

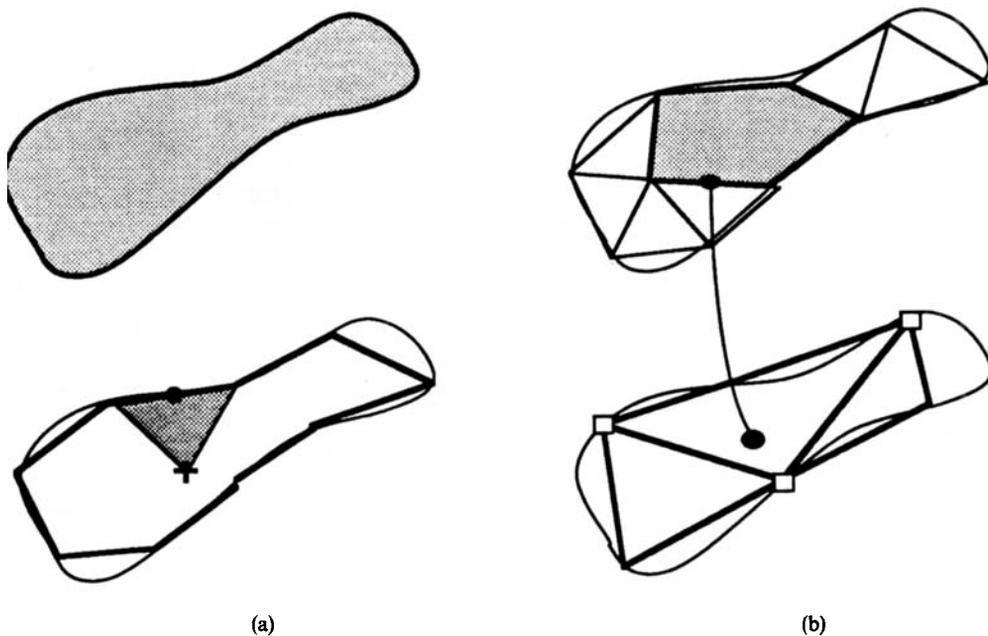


Figure 1. Advancing front mesh generation in 2D: (a) discretization of the edges, creation of a point and a triangle; (b) in a partly meshed domain, interpolation of the mesh size information necessary to build the next element from a background mesh before creation of a new point

discretized and the points and elements are then created simultaneously (Figure 1(a)), thus bypassing the point generation step required in other meshing techniques such as Delaunay triangulation.¹⁹ For graded meshes the mesh size information is recovered by interpolation (Figure 1(b)) from a background mesh of the same object. The efficient implementation of these concepts requires that proper data structures be used for the geometrical searches encountered during meshing. The emphasis in this section is on additional topics of special interest for 3D implementation; basic results are not reviewed.²⁰

2.2. Internal representation of 3D objects

The interaction between the computational domain—called the object here—and the user is done with a two-dimensional representation of the object, either a sheet of paper or a CRT. This is natural for a plane computational domain, for which it is simple to devise an unambiguous representation. For three-dimensional domains, however, the integrity of the visual representation is harder to achieve. This increase in complexity is also true for the internal representation of the object, from which the mesh is built, and its careful design requires more thought than in 2D.

Early developments in computer-aided design led to the wireframe representation of 3D objects, then to the parametric representation of free-form surfaces (e.g. Reference 21). These surface patches are still widely used for the description of objects in commercial databases, usually integrated in what is called a boundary representation, or B-rep, of the object (Figure 2). The 3D region to be meshed is bound by an assembly of faces called a shell. Each face, a plane or curved surface, is bound by a loop. Each loop is itself the assembly of edges connecting at vertices. The B-rep can lead to inaccuracy or inconsistency¹⁴ when used as the sole description of an object, since no track is kept of the topological origin of each feature.



Figure 2. Boundary representation (B-rep) terminology

An object can also be represented by the union or intersection of elementary volumes, e.g. boxes, cylinders and cones, and such a framework is called constructive solid geometry or CSG. A purely CSG description of an object, such as the one used in octree-based mesh generators, does not allow for easy meshing of the boundaries.²²

Recognizing the fact that both types of representations are truly complementary, an international effort is under way to incorporate both CSG and B-rep in a hierarchical database built around an object-oriented language called STEP, allowing the unique definition of any solid and all its features.¹³ The database written for the present work is also hybrid in nature. It allows handling of cylinders, cones and cubes. The boundary representation of these primitives is stored along with their implicit polynomial representation. The intersection of surfaces can then be computed using this second representation rather than the first one, and the transformation to develop a surface into a plane is easily computed.

2.3. Shell discretization

Before the 2D discretization of a plane domain, its edges have to be discretized. Similarly, in 3D the initial step is the discretization or triangulation of the shell. Each face is first mapped onto a plane using an invertible area-preserving transformation. This transformation is straightforward²³ for plane or ruled surfaces provided the surface is recognized as such using the CSG representation of the solid. A large proportion of surfaces encountered in polymer forming are of these types. In the case of free-form surfaces, for which no exact area-preserving transformation exists, a transformation similar to that of Reference 24 can be used: it maps the unit parametric square into a distorted parametric quadrilateral in order to approximatively conserve angles. In this latter case the transformation has to remain invertible, therefore linear, and it may distort the mesh, hence the necessity of using the CSG representation when possible. Each plane equivalent of a face is then meshed using the 2D mesh generator described in Reference 20. In addition, for the consistency of the final shell mesh, each edge is only discretized when encountered for the first time. The computed points are then duplicated to mesh adjacent faces. The meshed shell forms the initial advancing front for the 3D mesh generation.

2.4. Volume discretization

The discretization of the region or volume itself starts by the storage of the initial advancing front in a binary tree structure.²⁰ The triangle from which to build the first tetrahedron is then extracted from the root of the structure. Following the 2D procedure, the apex of the tetrahedron is determined by locating the centroid of the base triangle in the background 3D mesh, which spans the object to be meshed, and then interpolating h from the background mesh vertices. For convex objects, however, this may cause problems: if the new mesh is finer than the background mesh, new centroids on a face are not located inside the polyhedron defined by the background mesh. It is then necessary to perturb iteratively the co-ordinates of the point until they are inside the previous mesh to interpolate the mesh size from it.

The new tetrahedral finite element is built using the computed apex or an existing point if one is found close enough, a notion which will be explained later. In order to obtain a valid mesh, the generation of this new element should not create any intersection with existing geometrical features in its neighbourhood. Namely, the edges of the element should not intersect existing faces of the advancing front and vice versa. The proper programming of this intersection check is the key to the success of the advancing front generator. The two key points for doing so are detailed below.

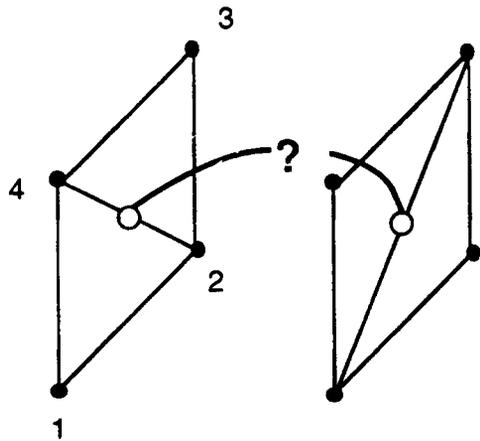


Figure 3. Topological identification allows the distinction between edges 1-3 and 2-4, while geometrical identification using the co-ordinates of the middle point does not

First of all, it is essential to deal with the inaccuracy introduced by finite precision floating point arithmetic, since it may lead to multiple answers to a given test. To this end, topology is used rather than geometry whenever possible: checks are performed on node numbers rather than co-ordinates, even if it involves a larger number of checks. The advantage of this approach is illustrated by the simple example of the diagonal of a quadrilateral (Figure 3). A geometrical check identifying each edge by the co-ordinates of its midpoint will conclude on the equivalence between the two edges; a topological check identifying each edge by its end vertices will yield the correct answer.

When geometrical checks cannot be avoided, a perturbation approach is used,¹⁴ the test on the distance $d < 0$ being replaced by $d < \varepsilon$, ε being a user-defined tolerance. Furthermore, for the mesh generator to yield consistent results regardless of the overall size of the object, ε is multiplied by the local mesh size for checks on distances and by the square of the mesh size for area checks. The perturbation ε is therefore a true percentage perturbation. The perturbation method may prohibit the construction of certain tetrahedra and thus invalidate the simplicial rule. If this leads the algorithm to come to a halt, it is necessary to temporarily set ε to zero to create a new element. This implicit use of machine zero tolerance may lead to a degeneracy that has to be dealt with later, and the number of this element is stored for this purpose. It should be noted that such occurrences have been encountered very rarely.

The second key point regarding the programming of the intersection check is the reduction of its cost. The most frequent computation is that of the distance between a point and a plane face. This requires computation of the unit normal vector to the face. The exact computation of the unit normal involves costly arithmetic, and it should be stored instead as long as the face remains in the front. For uniform meshes, both storage and computation of the length of the normal can be avoided by replacing its exact length by a constant, since all elements are similar.

While this intersection check is the major bottleneck for the meshing of small domains, a study of the distribution of total meshing time reveals that the recovery of the neighbourhood of the potential apex from the structure takes over for large domains (Figure 4). The neighbour search time plotted includes the cost of updating the tree structure. This can be explained by the following argument. The intersection checks are only performed on the faces located in the small neighbourhood recovered from the tree. While the cost of this search within the tree structure

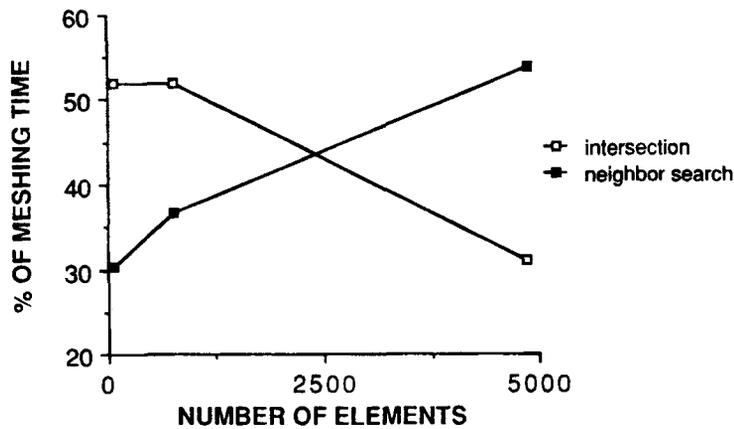


Figure 4. Percentage of total meshing time spent in the intersection and search functions, measured on a scalar computer

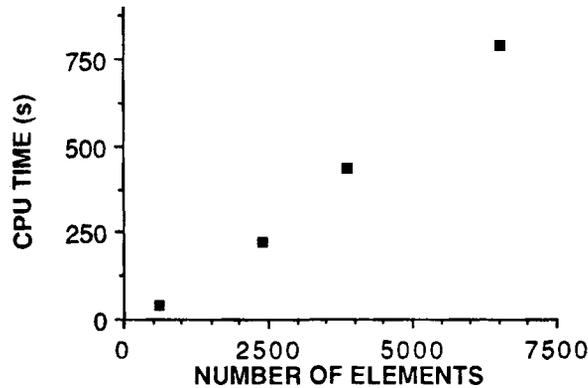


Figure 5. CPU time to build a 3D mesh of tetrahedral elements, measured on a Convex 220

inevitably increases with the problem size, the size and complexity of that neighbourhood remain constant and so does the number of intersection checks. The sum of both search and intersection check times is always above 80% of the total meshing time, clearly identifying these two areas as critical. The overall performance of the resulting mesh generator is linear at approximately 500 elements per minute on a Convex C220 (Figure 5).

2.5. Mesh validity

In order for the finite element problem to be well posed, the discretization of the computational domain must be valid. For a discretization T_h of a region \mathcal{W} with t elements K_i , this means¹⁵

$$\begin{aligned} \forall K_i \in T_h, \quad \text{volume}(K_i) &\neq 0, \\ K_i \cap K_j &= \emptyset, \quad \forall i, j, \quad i \neq j, \\ \forall \partial K_i \text{ face of } K_i, \quad (\exists j \neq i / \partial K_i &= \partial K_j) \text{ or } (\partial K_i \in \Omega_b \text{ boundary of } \Omega). \end{aligned}$$

The first two conditions are dealt with by using the perturbation approach, the verification of the second one being the task of the intersection procedure described above. The third condition is dealt with globally by checking Euler's formulae, given in Reference 23. The count of vertices v , edges e and faces f can be incremented as meshing progresses, and the final count with t tetrahedra should give $v + f - e = t + 1$. In addition, if the subscript 'b' is used for boundary items, the relations $2v_b - f_b = 4$ and $v_b = f_b - 2t + 2$ should be verified.

In practice, the condition of non-zero volume is not stringent enough to ensure an accurate solution. It is well known that the error bound on the solution of finite element problems depends on the shape of the elements. For the Stokes problem and linear elements the *a priori* error bound is $O(h)$ for perfectly regular tetrahedra, but increases to h^2/ρ for ill-shaped elements, where ρ is the diameter of the ball inscribed in the element.¹⁵ When a neighbourhood point rather than the ideal one is chosen as the apex of a new element, as mentioned in the previous subsection, the regularity $\sigma = h/\rho$ of the created element is of concern. The computation of the regularity is costly and it is approximated by α , called the aspect ratio (see Appendix). The histogram of the aspect ratio before Laplace smoothing for several meshes is plotted in Figure 6.

If the aspect ratio of an element is above 2.0, a variation of Laplace smoothing¹⁹ is performed on the vertices of that element after completion of meshing. Conventional Laplace smoothing of a point M of a mesh is the replacement of M by G, the centroid of the nodes linked to M by an edge. This is known to sometimes destroy the validity of the mesh and was modified as follows. Prior to smoothing a point M, of co-ordinate vector OM, the aspect ratios of the surrounding elements are computed. M is then replaced by the point on the line MG that approximately minimizes the maximum aspect ratio of all surrounding elements. This point has $OM + \lambda MG$ for co-ordinate vector, where λ belongs to $[0, 1]$ and is a multiple of 0.1. This Laplace smoothing is typically performed on 5% of the elements and its cost is small compared to the total meshing time. Meshes smoothed in this way show every little spread of aspect ratio and are therefore well suited for the generation of finite elements over the flow domain. These finite elements and the corresponding solution are detailed in the next section.

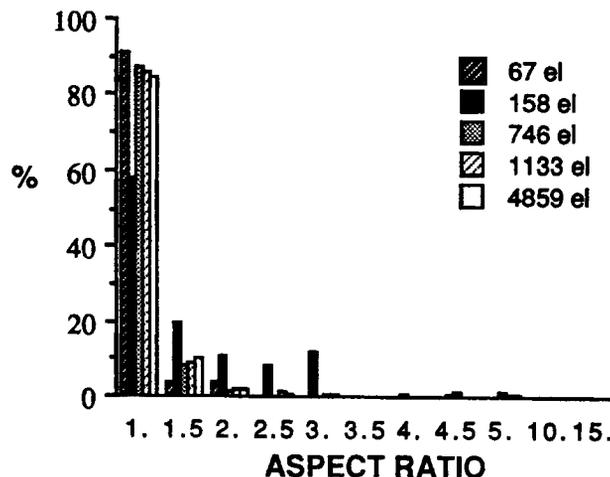


Figure 6. Distribution of the aspect ratios in 3D meshes of a cube before Laplace smoothing

3. EQUATIONS AND FINITE ELEMENT APPROXIMATION

3.1. Newtonian fluids

The steady state motion of a viscous incompressible flow in a domain Ω follows

$$\rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p + \nabla \tau^D = \rho \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

where ρ is the density, \mathbf{v} is the velocity and p is the pressure. Appropriate boundary conditions and a definition of the deviatoric of the stress tensor τ^D in terms of the velocity field are required to solve this system. For a Newtonian fluid τ^D is given by

$$\tau^D = -2\mu \dot{\gamma}, \quad (3)$$

where

$$\dot{\gamma} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T). \quad (4)$$

The solution of (1), (3), (4) under the constraint (2) can be carried out using an exact penalty method¹⁶ which strictly enforces the incompressibility constraint. Let us detail it for the steady state Stokes problem with Dirichlet boundary conditions, obtained by neglecting the inertial term in (1). A saddle point problem is defined by the augmented Lagrangian functional

$$\inf_{\delta \mathbf{v} \in \mathbf{V}}, \sup_{\delta p \in Q} \left\{ \frac{\mu}{\rho} \int_{\Omega} |\dot{\gamma}(\mathbf{v})|^2 d\Omega - \int_{\Omega} p \nabla \cdot \mathbf{v} d\Omega + \frac{r}{2} \int_{\Omega} |\nabla \cdot \mathbf{v}|^2 d\Omega - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega \right\}. \quad (5)$$

\mathbf{V} and Q are the spaces of velocity and pressure test functions respectively and r is a penalty parameter. Taking the first variation of (5) with respect to both velocity and pressure leads to the system

$$2 \frac{\mu}{\rho} \int_{\Omega} \dot{\gamma}(\mathbf{v}) : \dot{\gamma}(\delta \mathbf{v}) d\Omega - \int_{\Omega} p \nabla \cdot \delta \mathbf{v} d\Omega + r \int_{\Omega} |\nabla \cdot \mathbf{v}| \cdot \delta \mathbf{v} d\Omega = \int_{\Omega} \mathbf{f} \cdot \delta \mathbf{v} d\Omega, \quad \forall \delta \mathbf{v} \in \mathbf{V}, \quad (6)$$

$$\int_{\Omega} \delta p \nabla \cdot \mathbf{v} d\Omega = 0, \quad \forall \delta p \in Q. \quad (7)$$

Defining the discrete velocity vector \mathbf{U} , the discrete pressure P and the matrices \mathbf{A} and \mathbf{B} , (6), (7) is equivalent to the algebraic system

$$(\mathbf{A} + r\mathbf{B}^T\mathbf{B})\mathbf{U} + \mathbf{B}^T P = \mathbf{F}, \quad (8)$$

$$\mathbf{B}\mathbf{U} = 0. \quad (9)$$

Using the Galerkin procedure, the velocity trial functions Φ_i are chosen equal to $\delta \mathbf{v}$ and the terms of the matrix \mathbf{A} are

$$A_{ij} = \int_{\Omega_e} \frac{\mu}{2\rho} (\nabla \phi_i + \nabla \phi_i^T) : (\nabla \phi_j + \nabla \phi_j^T) d\Omega_e. \quad (10)$$

The pressure trial functions ψ_i are chosen equal to δp and the terms of the matrix \mathbf{B} are

$$B_{ij} = - \int_{\Omega_e} \psi_j \nabla \phi_i^T d\Omega_e. \quad (11)$$

This is solved in an iterative fashion, first solving for the velocity

$$(\mathbf{A} + r\mathbf{B}^T\mathbf{B})\mathbf{U}^{n+1} = \mathbf{F} - \mathbf{B}^T\mathbf{P}^n, \quad (12)$$

then the pressure

$$\mathbf{P}^{n+1} = \mathbf{P}^n + r\mathbf{B}\mathbf{U}^{n+1}. \quad (13)$$

It can be easily checked that once the algorithm has converged, \mathbf{U} and \mathbf{P} are solutions of the Stokes problem and their value is independent of r . Convergence occurs in one iteration if r is large enough. It will be seen later that in order to preserve the compatibility between the interpolation spaces \mathbf{V} and \mathbf{Q} , a discontinuous pressure is used. In such a case, (13) can be solved at the elementary level and the pressure is absent from the overall algebraic system, greatly reducing its size. For large 3D problems this is best solved using an iterative method such as the one proposed in Reference 25. Finally, when the inertial term in (1) is not negligible, it is added to (12). For the low-Reynolds-number flows presented here it is added to the right-hand side as a forcing term and a fixed point method is used. For large Reynolds number it should be added to the left-hand side, thus creating a non-linear problem to be solved with an appropriate iterative method.

3.2. Non-Newtonian fluid

The viscous behaviour of polymeric fluids is characterized by large variations of the viscosity with the rate of strain. In the application presented here the viscosity of the polymer is modelled by a power law and the Newtonian viscosity μ defined in (3) is replaced by a non-linear function of the second invariant of the rate-of-strain tensor:

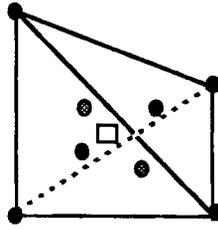
$$\mu^* = m|\dot{\gamma}|^{n-1}. \quad (14)$$

This non-linearity is again dealt with using an augmented Lagrangian,¹⁷ leading to a linear global system for \mathbf{v} and a pointwise system for τ^D .

3.3. Discretization

The discretization of the mixed problem requires care, since the incompressibility condition should be satisfied at the elementary level. In 3D, several hexahedral brick elements have been tested successfully on benchmark Navier–Stokes problems,²⁶ but the literature on tetrahedral elements is confined to the non-trivial derivation of the trial function basis.^{27, 28} The tests we have conducted so far show that the P1–P0, the simplest conforming tetrahedral Lagrange element with linear velocity at the vertices and discontinuous pressure at the centroid, exhibits pressure locking. In order to suppress this behaviour, additional degrees of freedom at the face centroids are required, resulting in 24 velocity degrees of freedom. If the pressure is kept P0, the enriched element is called P1⁺–P0 (Figure 7). If the pressure is linear defined at the centroid, the element is called P1⁺⁺–P1.²⁹

These non-conforming elements are theoretically shown²⁹ to provide the same asymptotic accuracy as conforming elements with fewer degrees of freedom. The extra degrees of freedom are nonetheless a burden on the computational cost, as can be shown using Euler's formulae. With the notation defined above, the velocity degrees of freedom for a given problem meshed with P1–P0 elements is $3v$, while the maximum number of boundary conditions is $3v_b$. The same values for a P1⁺–P0 mesh can be determined to be respectively $3(2t + v + v_b - 2)$ and $3(3v_b - 4)$, a tremendous increase over the P1–P0. The expense of going to non-conforming quadratic elements is even larger, and their use in confined geometries can only be practical if they are also of quadratic geometry, thus enabling the approximation of complex geometries with very few elements. The present work is restricted to the P1⁺–P0 element.



- ● velocity P1+ 24 dof
- pressure P0 1 dof

Figure 7. Degrees of freedom (dof) of a P1⁺-P0 tetrahedron

3.4. Error estimate

The Stokes problem was shown to be equivalent to a saddle point problem in Section 3.1. This equivalence is strictly valid only on the continuum Ω . The same treatment applied to the discrete problem generates extra boundary integrals inside the domain, called inter-element jumps. Since they were neglected, they appear in a residual-type *a posteriori* error bound ε of the error expressed in the following norm, where the subscript h represents the variables in discrete space.^{10, 12}

$$\|p - p_h\|_0 + \|\mathbf{v} - \mathbf{v}_h\|_1 < \varepsilon. \quad (15)$$

This error bound ε is the RMS of the contributions η_K over the elements K of the discretization:

$$\varepsilon = \left(\sum_K \eta_K^2 \right)^{1/2}. \quad (16)$$

These can be computed from the solution field and can be written for 3D problems, dropping the subscript h for the sake of clarity, as

$$\eta_K^2 = C \left(h^3 \|\mathbf{f} + \eta \nabla^2 \mathbf{v} - \nabla p\|^2 + \sum_{\partial K} [h^2 |(p - \tau_{nn}) - (p - \tau_{nn})^*|^2] \right). \quad (17)$$

The second term is the sum of inter-element normal stress jumps, the starred value being computed in the neighbouring element of K . The same error bound is also valid for non-Newtonian fluids with some conditions on the integrability of \mathbf{f} .¹¹ The properties of ε are illustrated for 2D examples in Reference 10. Furthermore, properties of the local components η_K of the error bound ε are given in Reference 12. When computed for 3D linear elements with constant pressure and no body force, the only non-zero terms are the inter-element jumps, and η_K is straightforward to evaluate during post-processing.

The variation of ε with increasing P1⁺-P0 mesh size for the flow of a Newtonian fluid at $Re = 10^{-6}$ in a 3D tube is plotted in Figure 8. The velocity profile is known to be parabolic and the pressure drop linear in such a case, and the true error norm can be computed. It is plotted along with ε and can be seen to increase linearly with h for small mesh sizes. The departure from linearity for the coarse mesh can be attributed to the geometrical approximation: the cylindrical geometry is not well represented by a polyhedron with few sides. The convergence rate of ε itself is not quite

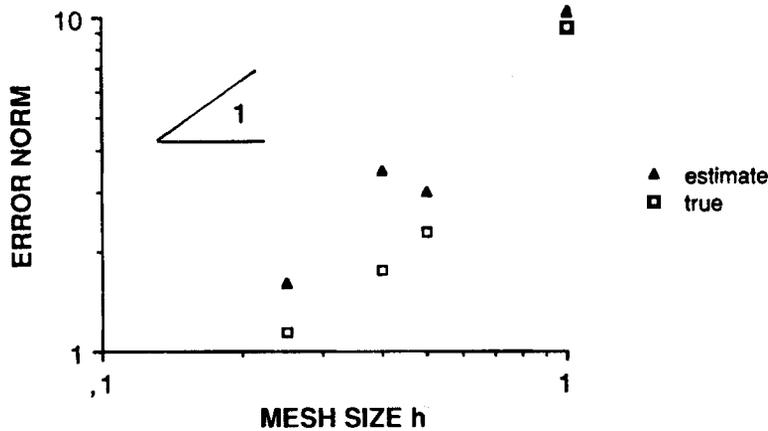


Figure 8. True error norm and estimate e for the $P1^+-P0$ solution of the flow of a Newtonian fluid in a cylinder

unity, and this may be due to element shapes, as explained earlier. Extra steps are taken when using ε in the adaptive cycle to minimize the effects of these variations.

This type of error estimate can be generalized to the Navier–Stokes equations.¹² For the very-low-Reynolds-number flows encountered in polymeric forming processes, the magnitude of the non-linear convective term is small and the above results can be used for practical purposes without any modification.

4. ADAPTIVE CYCLE

4.1. Transition operator

The adaptive solution process is designed to provide a solution of chosen accuracy at minimum computer cost and with minimum engineer's involvement. While the standard practice in use in finite element analysis today can be seen as an open loop, the adaptive process is a closed-loop cycle (Figure 9). Following the generation of the mesh K_0 and the finite element solution of the discretized problem, the error is estimated by η_K . The transition operator uses the error estimate field η_K and the error level e prescribed by the engineer to generate a field of desirable mesh size:

$$h_1 = \left(\frac{e}{\eta_K} \right) h_0. \quad (18)$$

This transition operator is valid for perfectly regular meshes, but for real meshes one should introduce the regularity of individual elements to prevent spurious refinement where the locally high values of η_K are due to element distortion rather than size. The transition operator becomes

$$h_1 = \left(\frac{e}{\eta_K} \right) \sigma h_0. \quad (19)$$

The discontinuous mesh size field defined element by element is then averaged onto the nodes of K_0 and becomes a continuous nodal field $h_1(x, y, z)$. It has been our observation that the mesh size field thus generated exhibits strong gradients. Gradients of magnitude higher than 0.5 often lead to the failure of the mesh generator. Smoothing is therefore performed on $h_1(x, y, z)$ by averaging the

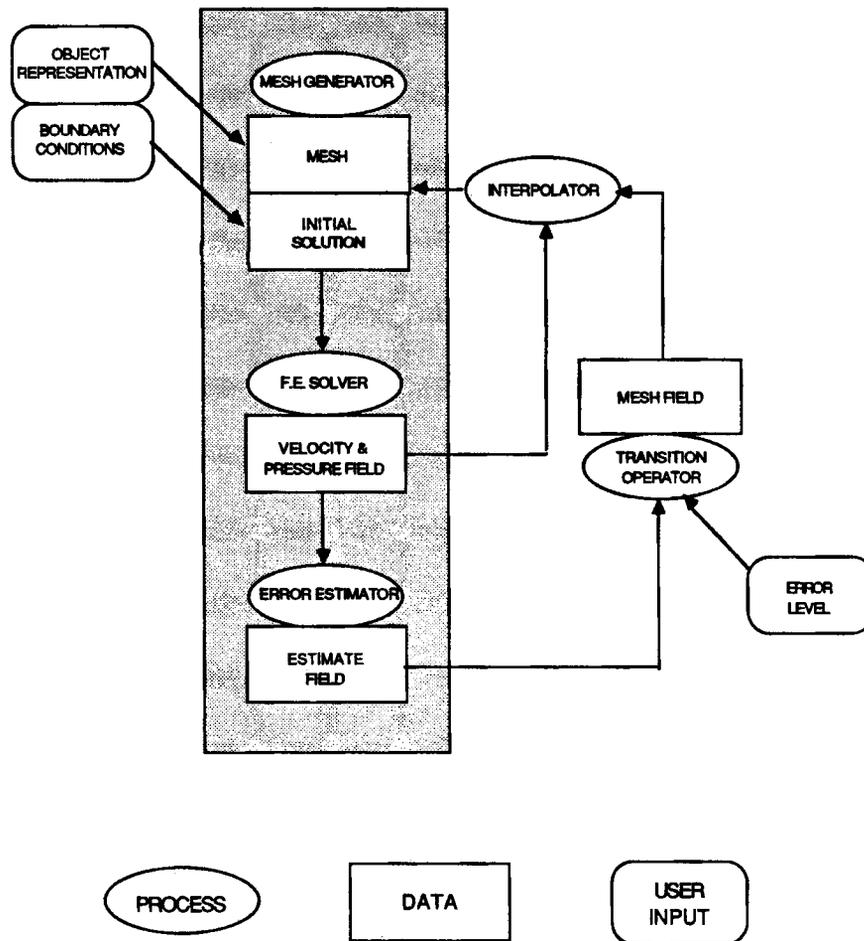


Figure 9. Adaptive cycle: an ellipse represents a process, a rectangle data and a rounded rectangle interaction with the user and databases

nodal values on each element and assigning this average to its centroid. This new discontinuous mesh size field is then averaged onto the nodes as above. This procedure iteratively leads to the elimination of large gradients. The standard deviation of the function $h_1(x, y, z)$ is observed to decrease linearly with each iteration, which indicates that too many iterations will remove the spatial variability of h_1 that allows adaptive meshing. Two iterations were found to be sufficient. The smoothed mesh field h_1 defined at the vertices of K_0 is the background from which the new mesh K_1 is created. To determine the desired mesh size of a new element from h_1 , interpolation is performed on K_0 by locating the centroid of the base of each new element before its apex is built. Once the new mesh is completed, interpolation is also used to generate an initial solution on it for time-dependent problems or iterative finite element solvers. The cycle is then complete.

4.2. Interpolator

The implementation of the interpolator requires special care. The segment tree,³⁰ a nested binary structure in which the mesh is stored, is the core of the interpolator. It allows the location of

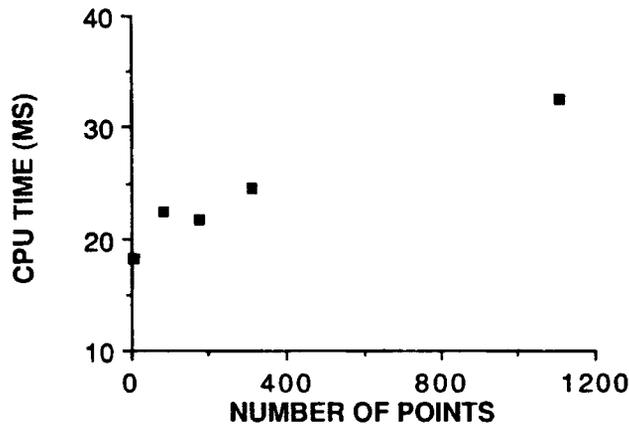


Figure 10. CPU time to locate a point in a 3D mesh using the segment tree, measured on an Apollo DN4000. The points stored in the segment tree are the vertices of the mesh

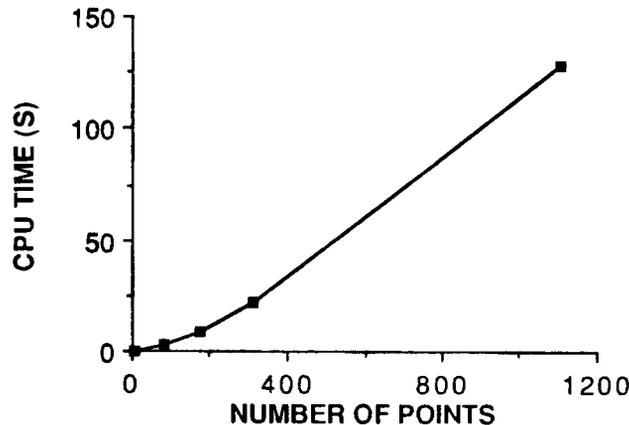


Figure 11. CPU time to build a segment tree for a 3D mesh, measured on an Apollo DN4000. The points stored in the segment tree are the vertices of the mesh

a point given by its co-ordinates optimally.²⁰ Average times for location of points in the tetrahedral 3D discretization of a unit cube are plotted in Figure 10. The increase of location time with mesh size is polylogarithmic, to be compared with a linear increase if a linear list of elements was used.

A new segment tree needs to be created for each mesh, and it is important that the time taken to create a segment tree be itself checked. It is plotted in Figure 11 and can be observed to be superlinear. Moreover, it is small compared to the solution time of the finite element problem. It should be noted that the implementation of the segment tree makes use of dynamic allocation and cannot be easily vectorized.

The closed-loop process continues until the error level set by the engineer is reached. This may be done theoretically in a single cycle, since the error on mesh K_1 should be equipartitioned and have the value ϵ everywhere. It is not so in practice, and the cycle can be used iteratively. To this end, intermediate meshes can be created automatically using intermediate error levels in the

transition operator. The iterative nature of the process calls for interaction at every cycle with both the object representation and the boundary conditions on that representation. Using the previous mesh as a definition of the object is not appropriate, since geometrical features not present in a coarse mesh may appear in a finer mesh. The object and boundary conditions are stored in a database totally independent of the segment tree, which only contains the mesh. Furthermore, neither this database nor the segment tree is needed during the finite element solution, and a linear list of all elements can be used then, reducing the space requirement to a minimum. However, this forces computation and assembly of the matrix for each cycle.

5. APPLICATION TO POLYMER MELT FLOW

5.1. Extrusion rheometer

To test the validity of the finite element solution with the above discretization, the pressure drop for the flow of a polyethylene in an extrusion rheometer is first studied. Isothermal experimental results for this problem are reported by Lenk³¹ for an axisymmetric cone of 5.4° angle and 40 mm length. The computational domain used for the numerical test is a cone of length 20 mm (Figure 12). No use is made of symmetry to reduce the size of the computational domain. Entrance and exit cylindrical sections are added to provide developed flow at the outlet and to allow proper Dirichlet boundary conditions at the inlet. The rheology of the fluid at 190°C is modelled³¹ using a power law, with a consistency index $m = 2.19$ SI and a power-law index $n = 0.4$. The developed power-law velocity profile is imposed at the inlet, with an average velocity of 0.66 mm s^{-1} , corresponding to a piston speed of 4 cm min^{-1} .

The total pressure drops computed with a 725 $P1^+ - P0$ mesh and a 2082 $P1^+ - P0$ mesh (Figure 13) are listed in Table I. The contributions of the 20 mm long exit duct and the conical section are isolated. The flow domain being axisymmetric, a 2D finite element code was used to conduct a mesh refinement analysis. The 2D value tabulated does not vary with further mesh refinement and can be considered to be a reference. It can be seen that the solution in 3D tends towards that reference value, but that a large number of degrees of freedom are needed to achieve

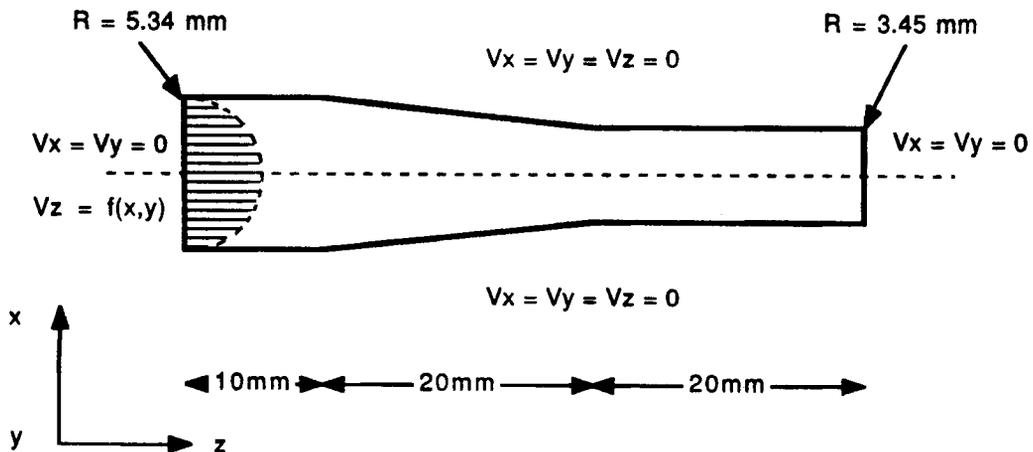


Figure 12. Cross-section of geometry and boundary conditions of extrusion rheometer

Table I. Pressure drop for a power-law fluid in the cone of an extrusion rheometer

Pressure drop (kPa)	3D 725 elements	3D 2082 elements	2D	Lubrication analytical	Lenk ³¹ experimental
ΔP total	77.0	71.9	66.0	66.5	—
ΔP cone 20 mm	26.3	25.9	22.8	22.8	—
ΔP tube 20 mm	45.9	40.5	36.9	36.7	—
ΔP cone 40 mm	—	—	—	129.0	170.1

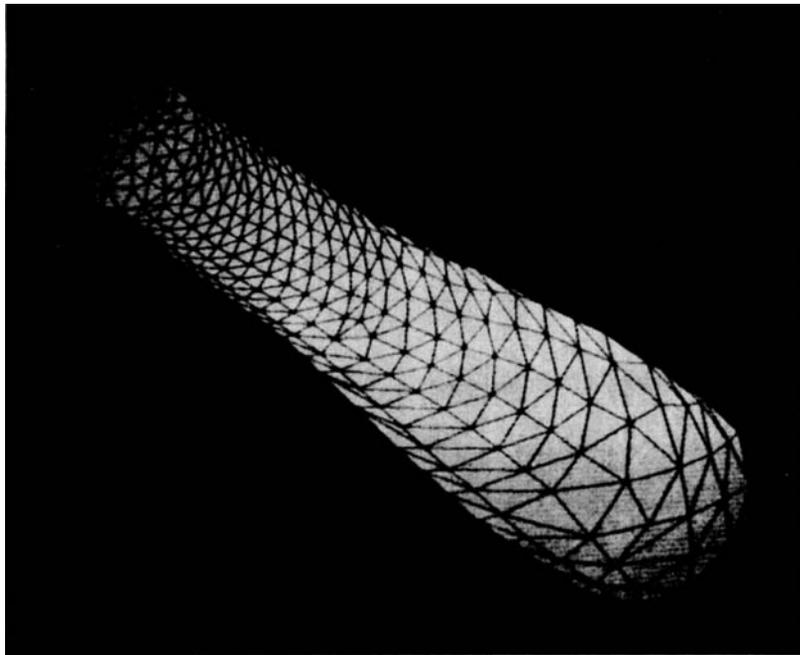


Figure 13. Mesh of extrusion rheometer

proper accuracy. This is due in part to the geometrical approximation involved in replacing curved surfaces with polyhedra. The study of the complete domain allows us to conclude that there is no major effect due to the non-symmetry of the 3D mesh on the solution.

The axisymmetry and small conical angle of the case also allow analytical pressure drop computations to be performed using the lubrication approximation.³² The close agreement in Table I between the lubrication approximation, the 2D solution and fine mesh result in 3D is a validation of the code for the rheological model chosen. On the other hand, a 32% discrepancy between the experimental and the lubrication results can be observed for the 40 mm cone, as reported in Reference 31. This might be attributed to two causes, experimental error and the choice of rheological model, since the viscosity of real fluids, unlike the prediction of the power law, reaches a limit value for zero shear.

5.2. Abrupt contraction

The adaptive process can best be studied in a flow domain with a geometrical singularity. An abrupt contraction was chosen (Figure 14) with Neuman boundary conditions at the inlet. The rheology of the fluid is as above. The initial user-defined mesh of 766 elements is coarse and uniform. The corresponding pressure along its axis of symmetry is plotted in Figure 15. The total error ϵ estimated on that mesh is 4.85% of the total solution norm. This does not take into account geometrical approximations. The error field η_k on that mesh is represented in Plate 1. The areas of high error are the contraction plane itself as well as the exit duct, where the number of free nodes per section is only about seven, a small value to approximate a non-Newtonian velocity profile. An error level of 1.6% is then chosen by the user and the mesh size field required to obtain that error level is computed and smoothed. An upper bound of 1.2 and a lower bound of 0.3 are set for the

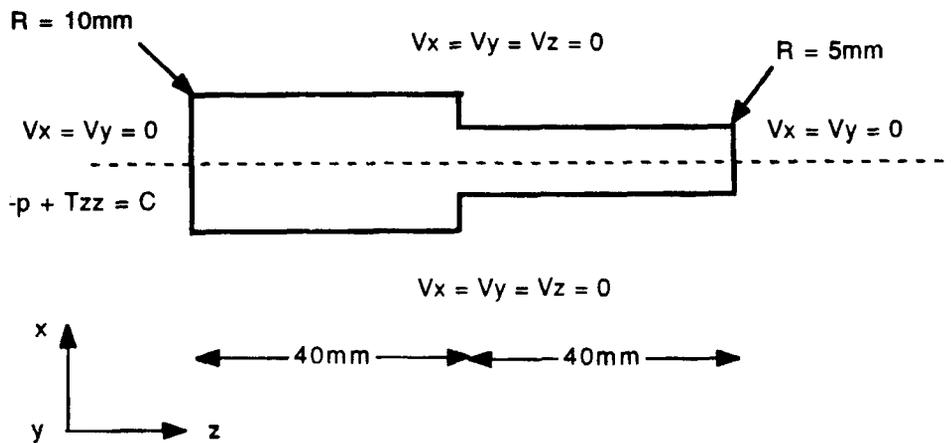


Figure 14. Cross-section of geometry and boundary conditions of 2:1 abrupt contraction

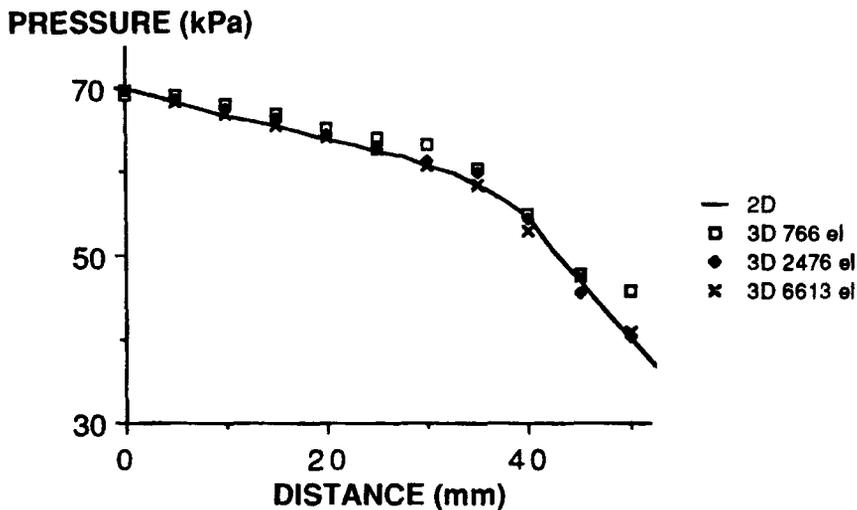


Figure 15. Pressure along the axis of symmetry of 2:1 contraction

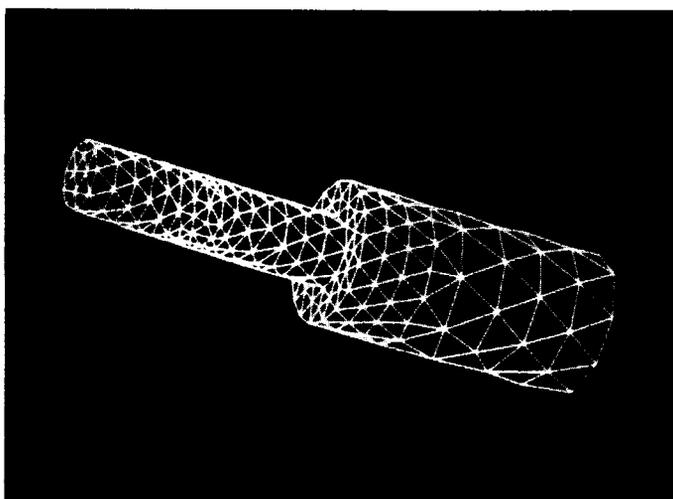


Figure 16. Adaptive mesh of 2:1 contraction

ratio h_1/h_0 . This prevents undue refinement that would increase the number of elements beyond manageable values, as well as excessive coarsening that would lead to a worsening of the geometrical approximation. The resulting mesh of 2476 elements is shown in Figure 16 and the corresponding error field in Plate 2. The total error is 2% and the spread of the error is now less than in the original mesh. This equipartitioning of the error field is characteristic of adaptive meshes. The pressure profiles corresponding to this adaptive mesh, a finer uniform 3D mesh and a 2D reference solution obtained by a mesh refinement analysis are plotted in Figure 15. The agreement between the three solutions is good and only the initial 3D mesh leads to a visible departure from the reference.

The computational speed of the finite element solver itself can be illustrated by the results for the adaptive mesh of the abrupt contraction. For 2476 elements the number of equations to solve is 17 568 for the velocity alone, coupled with 2476 pressure unknowns. This was solved in 54 min on a Convex C220.

Finally, a view of the inside of the contraction (Plate 3) allows us to check the distribution of the aspect ratios of the elements for this graded mesh. Most, i.e. 93%, are less than 2.0, 1.0 being the value for an equiangular tetrahedron. This can be further checked by the distribution on two additional cross-sections (Plate 4). Such a high uniformity can only be achieved in confined geometries by smoothing the mesh size field before interpolation. It was checked that any increase in the number of distorted elements affects the results of the computation.

6. CONCLUSIONS

The present paper has shown that global remeshing can be used as the core of an adaptive strategy for the solution of viscous flows in 3D. The fast generation of valid triangulations has been achieved by the use of tree-like structures and both topological and geometrical representations of the domain and its boundaries. The mesh generator allows meshing of confined flow domains without partitioning. The quality of the mesh is monitored with an aspect ratio that indicates regularity of the mesh in a well-defined sense other than a purely geometrical sense. The mesh size is locally determined by a nodal field derived from an error estimate. Extensions of the method

could provide for definition of the mesh size field using any scalar quantity, allowing for adaptivity with respect to other criteria.

The non-inertial character of the flow leads to a dependence of the error on both the velocity and pressure, and this was successfully dealt with by using an *a posteriori* error estimate well suited for the mixed problem. Not only does the error estimate drive the adaptive cycle, it can also increase the confidence placed in numerical analysis results. The adaptive strategy presented here is therefore a step towards a better use of the engineer's time.

APPENDIX

The upper bound on the error for the finite element of the Stokes problem increases with the condition number of the elementary Jacobian matrices. For linear tetrahedra this in turn is bound by σ , called the regularity of the element:

$$\sigma = h/\rho, \tag{20}$$

where h is the maximum of all edge lengths over the element and ρ is the diameter of its inscribed sphere. This purely geometrical criterion can be computed by recalling that the centre of the inscribed sphere is by definition the intersection of normals to the four faces. It is equidistant from all four faces and its co-ordinate vector \mathbf{x} is the solution of the system

$$\mathbf{n}_i \cdot (\mathbf{x} - \mathbf{g}_i) = r, \quad \forall i \in [1, 4], \tag{21}$$

where \mathbf{g}_i is a point on face i and \mathbf{n}_i is the unit normal to face i . The distance r is of course the radius of the inscribed sphere, equal to $\rho/2$. The point \mathbf{g}_i can be chosen as the centroid of face i and the system can be detailed as

$$\mathbf{n}_i \begin{pmatrix} l_i \\ m_i \\ n_i \end{pmatrix}, \quad \mathbf{g}_i \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix}, \quad \mathbf{x} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{22}$$

$$l_i x + m_i y + n_i z - r = l_i u_i + m_i v_i + n_i w_i, \quad \forall i \in [1, 4]. \tag{23}$$

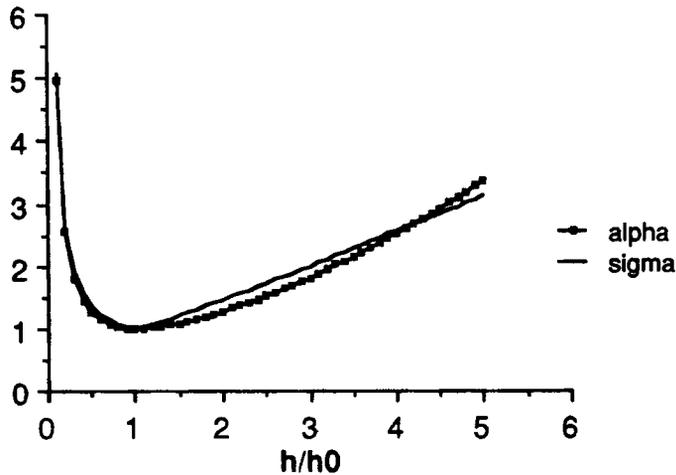


Figure 17. Regularity σ^* and aspect ratio α of a tetrahedron with an equiangular base as its height varies. The height of the tetrahedron is normalized by h_0 , the height of an equiangular tetrahedron, and σ^* is normalized by its value for the same tetrahedron

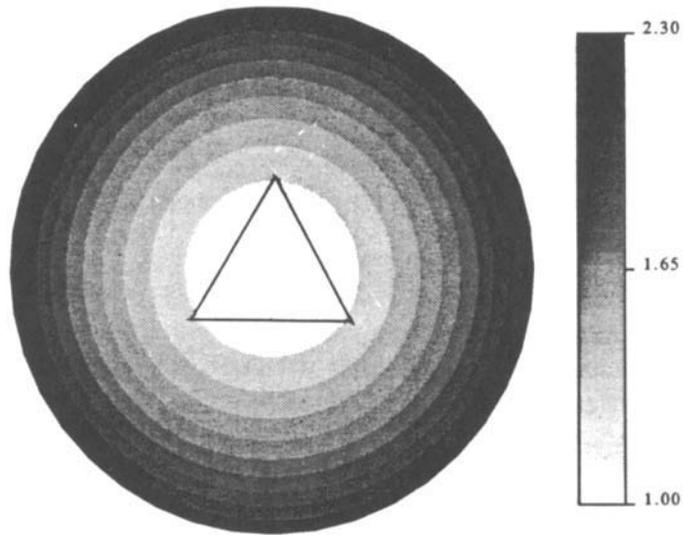


Figure 18. Isolines of σ^* in a plane at a distance h_0 from the base of the tetrahedron

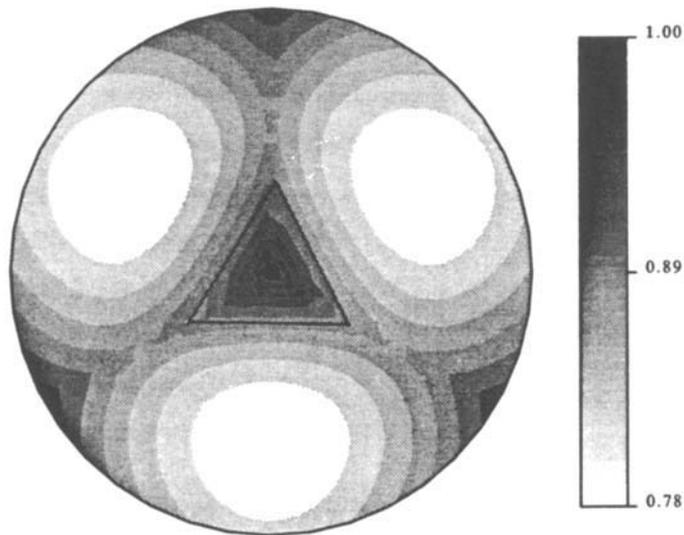


Figure 19. Isolines of α/σ^* in a plane at a distance h_0 from the base of the tetrahedron

Performing Gaussian elimination on the system of unknowns x , y , z and r leads directly to the value of the fourth unknown r and to σ .

Two sensitivity analyses were carried out to study the variations of σ with the deformation of a tetrahedron away from the equiangular shape. Let us introduce the values h_0 and σ_0 , the height of an equiangular tetrahedron and its regularity respectively. This leads to the definition

$$\sigma^* = \sigma/\sigma_0, \quad (24)$$

which is equal to unity for an equiangular tetrahedron. In Figure 17, σ^* is plotted for variations of the height h of an element with a fixed equiangular base. It can be seen that the error increases for the smallest distortion of the element. In Figure 18 the apex of the tetrahedron moves in a plane at a distance h_0 from its fixed equiangular base and the isolines of σ^* in that plane are plotted. The projection of the base of the tetrahedron is also drawn.

The computation of the regularity σ^* requires the computation of the unit normals and is too costly to be used in the mesh generator. An estimate α of σ^* can be computed:

$$\alpha = \frac{\sqrt{2} \bar{h}^3}{12 V}, \quad (25)$$

where \bar{h} is the average edge length over the element and V is the element volume. Since the volume is known from the computation of the determinant of the Jacobian, the aspect ratio α is cheaper to compute than σ . The behaviour of α is seen to compare well with that of σ^* when the height of the element varies (Figure 17). In Figure 19 the isolines of α/σ^* in the h_0 -plane show that the ratio remains reasonably close to unity when the apex moves in that plane. On this basis of these comparisons, the aspect ratio α is used in lieu of the regularity σ throughout the programme.

REFERENCES

1. A. Jameson, 'The evolution of computational methods in aerodynamics', *ASME J. Appl. Mech.*, **50**, 1052–1070 (1983).
2. R. Löhner, 'Finite elements in CFD: what lies ahead', *Int. j. numer. methods eng.*, **24**, 1741–1756 (1987).
3. I. Babuška and W. C. Rheinboldt, 'Error estimates for adaptive finite element computations', *SIAM J. Numer. Anal.*, **15**, 736–754 (1978).
4. R. E. Bank and A. W. Sherman, 'Algorithms for the multilevel solution of finite element equations', *Proc. SIAM Sparse Matrix 1978*, 1978, pp. 62–89.
5. P. Zave and W. C. Rheinboldt, 'Design of an adaptive parallel finite element system', *ACM Trans. Math. Softw.*, **5**, 1–17 (1979).
6. J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz, 'Adaptive remeshing for compressible flow computations', *J. Comput. Phys.*, **72**, 449–466 (1987).
7. O. C. Zienkiewicz and J. C. Zhu, 'A simple error estimate and adaptivity procedure for practical engineering analyses', *Int. j. numer. methods eng.*, **24**, 337–357 (1987).
8. O. C. Zienkiewicz, Y. C. Liu and G. C. Huang, 'Error estimation and adaptivity in flow formulation for forming problems', *Int. j. numer. methods eng.*, **25**, 23–42 (1988).
9. J. Peraire, J. Peiro, L. Formaggia, K. Morgan and O. Zienkiewicz, 'Finite element Euler computations in three dimensions', *Int. j. numer. methods eng.*, **26**, 2135–2159 (1988).
10. E. Abdallas, 'Résolution performante du problème de Stokes par mini-éléments, maillages auto-adaptatifs et méthodes multigrilles', *Ph.D. Thesis*, Ecole Centrale de Lyon, 1987.
11. J. Baranger and H. El Amri, 'Estimateurs d'erreur pour le calcul adaptatif d'écoulements quasi-newtoniens', Université de Lyon, 1989.
12. R. Verfürth, 'A posteriori estimates for the Stokes equation', *Numer. Math.*, **55**, 309–325 (1989).
13. I. Bey and U. Geganbach, 'The CAD*I interface for solid model exchange', *Comput. Graph.*, **12**, 181–190 (1988).
14. T. Smithers, 'AI-based design versus geometry-based design', *Comput. Aided Design*, **21**, 141–150 (1989).
15. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Approximation*, Series in Computational Mathematics, Vol. 5, Springer, Berlin, 1980.
16. M. Fortin and R. Glowinski, *Méthodes de Lagrangien Augmenté*, Dunod, Paris, 1982.
17. P. Tanguy, M. Fortin and L. Choplin, 'Finite element simulation of dip coating', Parts I and II', *Int. j. numer. methods fluids*, **4**, 441–475 (1984).
18. S. H. Lo, 'A new mesh generation scheme for arbitrary planar domains', *Int. j. numer. methods eng.*, **21**, 1403–1426 (1985).
19. J. C. Cavendish, D. A. Field and W. H. Frey, 'An approach to automatic three-dimensional finite element mesh generation', *Int. j. numer. methods eng.*, **21**, 329–347 (1985).
20. H. H. Dannelongue and P. A. Tanguy, Efficient data structures for adaptive remeshing with the FEM, *J. Comput. Phys.*, in the press (1990).
21. D. F. Roger and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, 1976.
22. M. A. Jerry and M. S. Shephard, 'Automatic three-dimensional mesh generation by the modified-octree technique', *Int. j. numer. methods eng.*, **20**, 1965–1990 (1984).

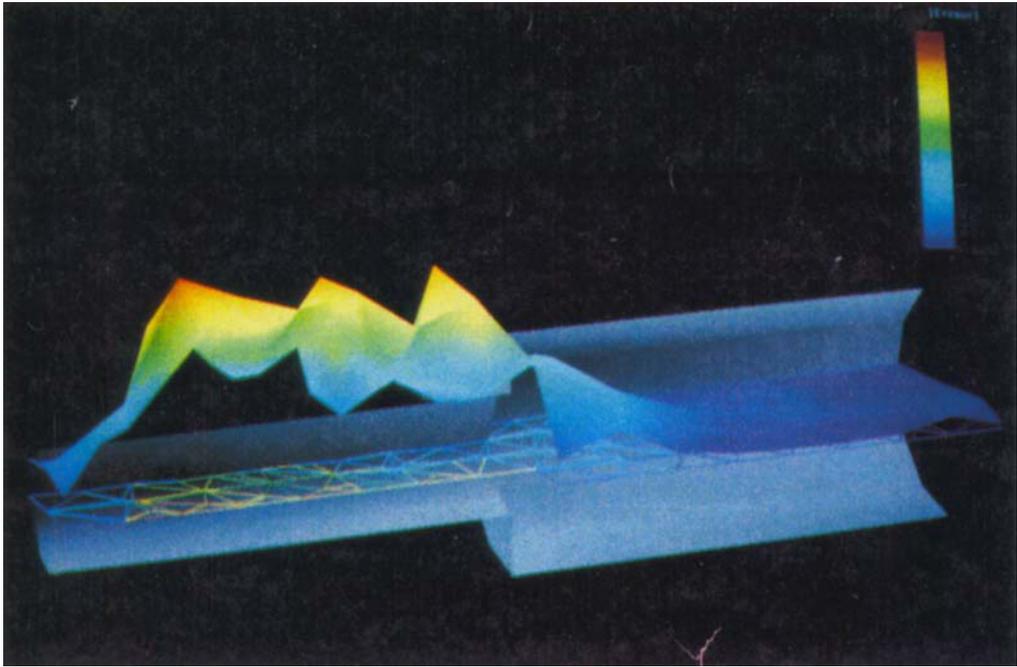


Plate 1. Error field on a cross-section of the coarse mesh of 2:1 contraction from 0% (blue) to 13.6% (red)

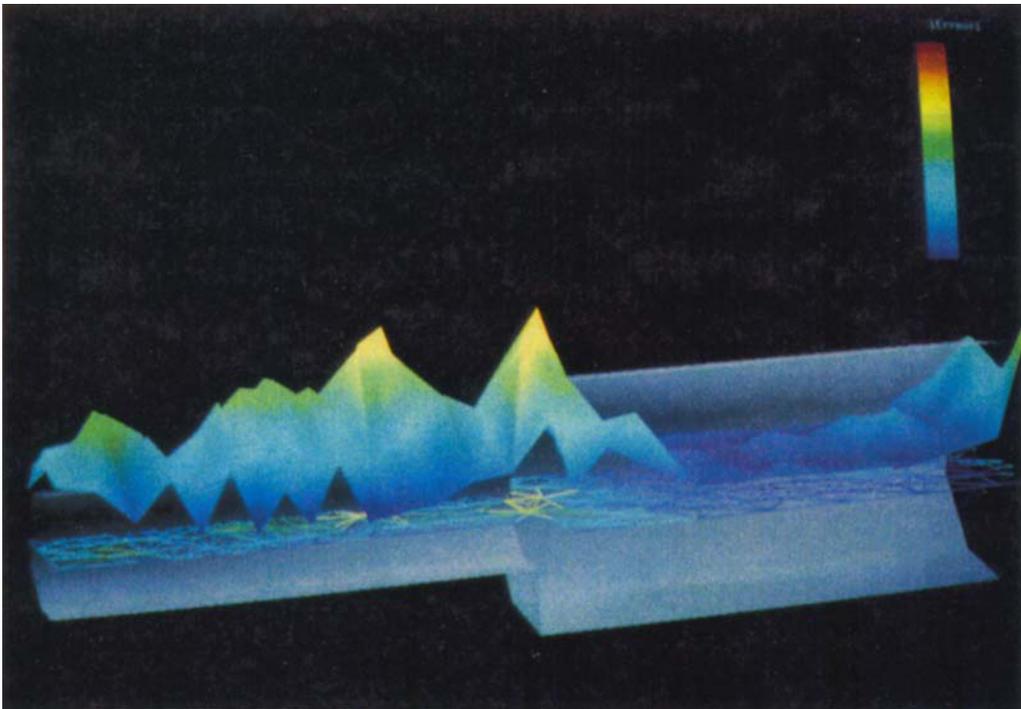


Plate 2. Error field on a cross-section of the adaptive mesh of 2:1 contraction from 0% (blue) to 7.5% (red)

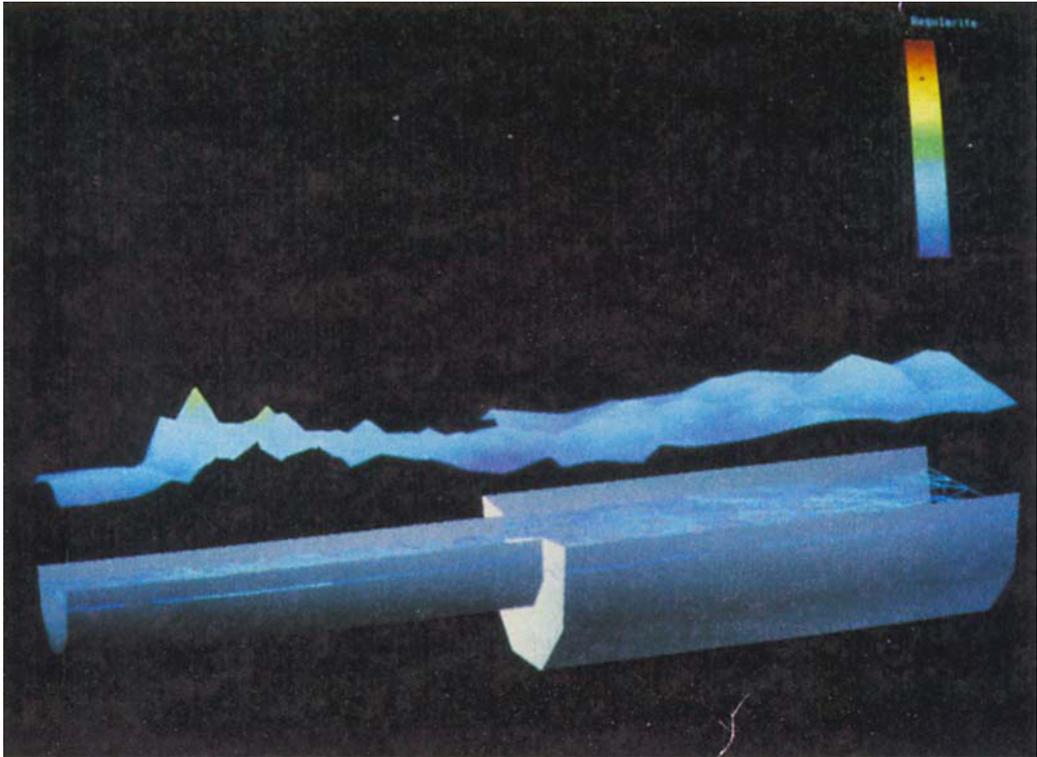


Plate 3. Aspect ratios on a cross-section of the adaptive mesh of the 2:1 contraction from 1.0 (blue) to 3.0 (red)

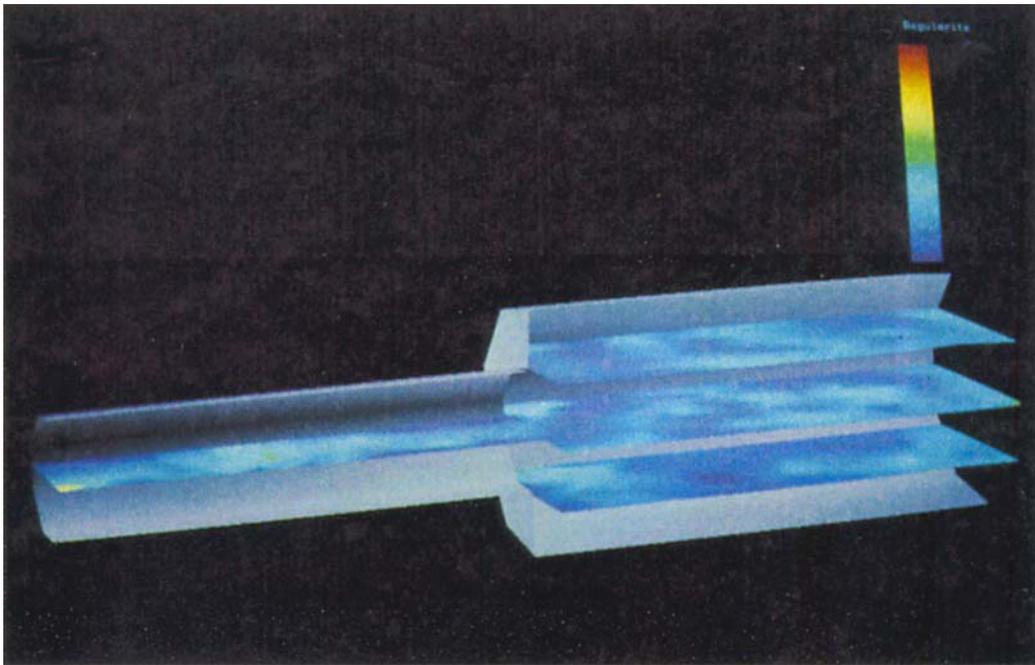


Plate 4. Aspect ratios on three cross-sections of the adaptive mesh of 2:1 contraction from 1.0 (blue) to 3.0 (red)

23. M. Berger, *Geometry, Vols 1 and 2*, Springer, New York, 1980.
24. R. L. Löhner, 'Some useful data structures for the generation of unstructured grids', *Commun. Appl. Numer. Methods*, **4**, 123–135 (1988).
25. M. Robichaud, P. A. Tanguy and M. Fortin, 'An iterative implementation of the Uzawa algorithm for three-dimensional fluid flow problems', *Int. j. numer. methods fluids*, **10**, 429–442 (1990).
26. M. Robichaud and P. A. Tanguy, 'Comparison of 3D finite elements for fluid flow', *Commun. Appl. Numer. Methods*, **3**, 223–233 (1987).
27. M. Crouzeix and P. A. Raviart, 'Conforming and non-conforming finite element methods for solving the stationary Stokes problem', *RAIRO Anal. Numer.*, **7** (R3), 33–76 (1973).
28. M. Fortin, 'A three-dimensional quadratic non-conforming element', *Int. j. numer. methods fluids*, **1**, 347–364 (1985).
29. F. Bertrand, M. Gadbois and P. A. Tanguy, 'Tetrahedral elements for fluid flow', *Int. j. numer. methods eng.*, submitted.
30. V. K. Vaishnavi, 'Computing point enclosures', *IEEE Trans. Comput.*, **C-31**, 22–29 (1982).
31. R. S. Lenk, 'Rheology and die design', in A. Wheelan and D. J. Dunning (eds), *Development in Plastics Technology*, Applied Science, 1982.
32. R. B. Bird, O. Hassager and R. Armstrong, *Dynamics of Polymeric Liquids, Vol. 1*, Wiley, Chichester, 1977.