INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14, 343–360 (1992)

TRANSIENT FINITE ELEMENT METHOD FOR CALCULATING STEADY STATE THREE-DIMENSIONAL FREE SURFACES

O. WAMBERSIE AND M. J. CROCHET

Université Catholique de Louvain, Unité de Mécanique Appliquée, 2 Place du Levant, B-1348 Louwain-la-Neuve, Belgium

SUMMARY

In order to reduce the cost of large three-dimensional calculations of steady state free surfaces, we have combined a time-dependent approach, a decoupling algorithm and a conjugate gradient solver along the lines introduced earlier by Gresho and Chan. The free surface is calculated separately by applying the kinematic condition to a number of faces defined on the undeformed surface. For the pseudo-time-marching technique we show that it is economical to adopt different time steps for the free surface calculation and the other fields. The accuracy of the method is tested on the well-known circular die problem; the method is then used to reveal the effects of inertia and shear thinning on square and rectangular dies.

KEY WORDS Free surfaces Extrusion Conjugate gradients Finite elements Three-dimensional free surfaces

1. INTRODUCTION

The goal of the present paper is the development of a cost-effective method for calculating three-dimensional extrusion flows of generalized Newtonian fluids, together with the prediction of the associated free surfaces.

The problem of three-dimensional free surface calculation has been considered by several authors over the last few years. Tran-Cong and Phan-Thien[1] have applied the boundary element method to extrusion processes and later extended their technique to viscoelastic flow problems.[2] In particular, Tran-Cong and Phan-Thien[3] were able to calculate the shape of a die required by an imposed cross-section of the free jet.

The use of finite elements for such problems was introduced by Shiojima and Shimazaki.[4,5] Their papers, however, contain little information about the numerical method and the mesh-updating technique.

Finite element techniques for three-dimensional extrusion problems have been introduced by Karagiannis *et al.*[6] They have extended to three dimensions the concepts introduced by Scriven and his collaborators (see e.g. Reference 7) for calculating free surfaces in two-dimensional flows, i.e. the use of spines for locating the free surface and for updating the mesh, together with the Newton–Raphson method for simultaneously solving the flow and the free surface. Non-isothermal flows have been considered in Reference 8, while coextrusion flows have been calculated in Reference 9. The method of spines produces some difficulties when the dies have sharp edges; it has been slightly modified in Reference 8 for calculating such flows.

0271–2091/92/030343–18$09.00

Received 11 February 1991
Revised 10 May 1991

A major concern is the cost of such calculations; in particular, one wishes to calculate the flow through a complex die together with the shape of the jet. The cross-section of the die is often complicated and requires a dense finite element mesh. Moreover, many polymers exhibit a viscoelastic character, which considerably increases the non-linear character of the problem and the number of variables.

Despite their inherent qualities, the methods referred to above may turn out to be very expensive for solving complex flows. Boundary elements do not give rise to sparse matrices, while a full Newton–Raphson method prevents decoupling and leads to very large active matrices in a frontal solver. Our goal here is to extend earlier progress in finite element methods to extrusion calculations in order to reduce the computational cost. The main ideas of our work are as follows: (i) approach to the steady state behaviour through a transient method; (ii) decoupling of the velocity components, pressure and free surface calculations; (iii) use of semi-implicit techniques; (iv) use of alternate solvers.

In the present paper we calculate steady state solutions; we have thus introduced a number of simplifications which might qualify the method as 'pseudo-transient' rather than transient. A minor effort would be necessary for obtaining an accurate time dependence.

We essentially rely upon the method elaborated by Gresho and Chan[10] and Gresho,[11] with a special emphasis on low-Reynolds-number flows. The resulting linear systems are characterized by positive definite, symmetric matrices; they are solved by means of the incomplete Cholesky conjugate gradient method.[12] We will necessarily be brief on the method recalled in Section 2 for calculating the velocity and pressure. In the meantime, further progress on projection methods has been accomplished by Gresho[13] and Gresho and Chan,[14] which would in particular improve the accuracy of a true time-dependent solution. The free surface is represented by a finite number of sheets with equations of the form $y=f(x, z, t)$ or $z=g(x, y, t)$, where x is the axial co-ordinate along the jet. The kinematic equation is then integrated implicitly for every sheet, with the use of spines. Edges are defined as the intersection of sheets; we may thus avoid the problem raised in Reference 8. The free surface integration is explained in Section 3.

A predictor–corrector scheme has been implemented with a view to an automatic calculation of the time step (see e.g. Reference 15). Very different time steps are found for the free surface and velocity field calculations. In Section 4 we show that the use of different time steps for both subproblems requires less computer time for reaching the steady state.

We then show in Sections 5–7 a number of illustrative examples which demonstrate in particular the power of finite elements for solving such flows.

2. BASIC EQUATIONS, SPACE AND TIME DISCRETIZATION

We wish to calculate three-dimensional flows of generalized Newtonian fluids for which the equations of motion are given as follows:

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)=-\nabla p+\nabla\cdot[\eta(\dot{\gamma})\,(\nabla\mathbf{v}+\nabla\mathbf{v}^{\mathrm{T}})], \qquad \nabla\cdot\mathbf{v}=0, \tag{1}$$

where $\mathbf{v}$ denotes the velocity vector, p the pressure and ρ the density. The shear viscosity η depends upon the local shear rate $\dot{\gamma}$, which is expressed as

$$\dot{\gamma}=[2\,\mathrm{tr}(\boldsymbol{dd})]^{1/2}, \tag{2}$$

where $\boldsymbol{d}$ is the rate-of-deformation tensor. In later sections we will consider a power-law fluid for

which

$$\eta = K\dot{\gamma}^{m-1}, \tag{3}$$

where K is the consistency factor and m the power index.

The flow domain is covered by a three-dimensional mesh of eight-node brick-like isoparametric elements. For the velocity field $\mathbf{v}$ we use a P^1–C^0 representation $\mathbf{v}^h$, while for the pressure, p^h denotes a P^0–C^{-1} representation. Let $\mathbf{V}$ and $\mathbf{P}$ denote the vectors of nodal velocities and nodal pressures respectively. The spatial discretization of the governing equations (1) is obtained by means of the Galerkin method. One then obtains the well-known set of constrained ordinary differential equations (see e.g. Reference 10)

$$\mathbf{M}\dot{\mathbf{V}} + \mathbf{N}(\mathbf{V})\mathbf{V} - \mathbf{K}(\mathbf{V})\mathbf{V} + \mathbf{C}\mathbf{P} = \mathbf{f}, \tag{4}$$

$$\mathbf{C}^{\mathrm{T}}\mathbf{V} = 0, \tag{5}$$

where a dot stands for the time derivative. In the system (4), (5), $\mathbf{M}$ stands for the mass matrix, $\mathbf{N}$ is the advective matrix, $\mathbf{K}$ is the diffusive matrix which depends upon $\mathbf{V}$ through $\dot{\gamma}$, while $\mathbf{C}$ is the gradient matrix and $\mathbf{C}^{\mathrm{T}}$ the divergence matrix. The nodal force vector $\mathbf{f}$ contains the contribution of the body and surface forces. The initial conditions have the form

$$\mathbf{V}(0) = \mathbf{V}^0, \tag{6}$$

with

$$\mathbf{C}^{\mathrm{T}}\mathbf{V}^0 = 0. \tag{7}$$

Our present goal is to solve steady state extrusion; we might thus as well solve the steady state equations

$$\mathbf{N}(\mathbf{V})\mathbf{V} - \mathbf{K}(\mathbf{V})\mathbf{V} + \mathbf{C}\mathbf{P} = \mathbf{f},$$

$$\mathbf{C}^{\mathrm{T}}\mathbf{V} = 0. \tag{8}$$

However, in view of the cost of direct solvers associated with (8) for three-dimensional flows, we wish to calculate the steady state as the limit of a time-dependent flow, although we are not basically interested by the time evolution of the flow (and of the free surface, to be considered in Section 3). In polymer extrusion problems the Reynolds number is typically very low; diffusion dominates advection in (4).

With a view to decoupling the transient algorithm, we obtain on the basis of (4), (5) the consistent discretized form of the Poisson equation for the pressure by multiplying both sides of (4) by the matrix product $\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}$, i.e.

$$(\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C})\mathbf{P} = \mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}[\mathbf{f} + \mathbf{K}(\mathbf{V})\mathbf{V} - \mathbf{N}(\mathbf{V})\mathbf{V}] - \mathbf{C}^{\mathrm{T}}\dot{\mathbf{V}}. \tag{9}$$

In view of (5), the time derivative of $\mathbf{C}^{\mathrm{T}}\mathbf{V}$ must vanish identically and thus $\mathbf{C}^{\mathrm{T}}\dot{\mathbf{V}} = -\dot{\mathbf{C}}^{\mathrm{T}}\mathbf{V}$. As long as the mesh does not deform, $\dot{\mathbf{C}}^{\mathrm{T}} = 0$ and the last term on the right-hand side of (9) should also vanish.

Let us assume that the velocity field is known at time t_n and takes the value $\mathbf{V}^n$. In our calculations we decouple velocity and free surface calculations. Thus at time t_n we consider a fixed mesh on which we calculate the new velocity field $\mathbf{V}^{n+1}$ at time $t_{n+1} = t_n + \Delta t$. We first calculate a pressure field $\tilde{\mathbf{P}}$ on the basis of (9), i.e.

$$(\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C})\tilde{\mathbf{P}} = \mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}[\mathbf{f}^n + \mathbf{K}(\mathbf{V}^n)\mathbf{V}^n - \mathbf{N}(\mathbf{V}^n)\mathbf{V}^n] + \mathbf{C}^{\mathrm{T}}\mathbf{V}^n/\Delta t. \tag{10}$$

All matrices are calculated at time t_n.

In writing (10), we have substituted

$$\dot{\mathbf{V}} = (\mathbf{V}^{n+1} - \mathbf{V}^n)/\Delta t. \tag{11}$$

We have imposed

$$\mathbf{C}^{\mathrm{T}}\mathbf{V}^{n+1} = 0 \tag{12}$$

in calculating $\tilde{\mathbf{P}}$ in view of (5), but we have acknowledged that $\mathbf{C}^{\mathrm{T}}\mathbf{V}^n$ may not vanish because $\mathbf{V}^n$ was calculated on the mesh at time t_{n-1}.

Next we discretize (4) by means of an implicit scheme for the diffusive terms and an explicit one for the advective terms, i.e.

$$\mathbf{M}(\mathbf{V}^{n+1} - \mathbf{V}^n)/\Delta t + \mathbf{N}(\mathbf{V}^n)\mathbf{V}^n - \mathbf{K}(\mathbf{V}^n)\,[\alpha \mathbf{V}^{n+1} + (1-\alpha)\mathbf{V}^n] + \mathbf{C}\tilde{\mathbf{P}} = \mathbf{f}^n; \tag{13}$$

for $\alpha = 0$ and 1 we obtain Euler's explicit and implicit schemes respectively (see e.g. Reference 16, Chap. 21). We use $\alpha = 1$ throughout the rest of the paper. We note that, for the sake of simplicity, $\mathbf{K}$ is calculated in terms of $\mathbf{V}^n$ without any effect on the steady state solution. We may thus rewrite (13) as

$$[\mathbf{I} - \Delta t\,\alpha \mathbf{M}^{-1}\mathbf{K}(\mathbf{V}^n)]\mathbf{V}^{n+1} = \mathbf{V}^n + \Delta t\,\mathbf{M}^{-1}[\mathbf{f}^n + (1-\alpha)\mathbf{K}(\mathbf{V}^n)\mathbf{V}^n - \mathbf{N}(\mathbf{V}^n)\mathbf{V}^n - \mathbf{C}\tilde{\mathbf{P}}]. \tag{14}$$

The use of an implicit scheme would not *a priori* require mass lumping as suggested by the inverse $\mathbf{M}^{-1}$ in (14); however, mass lumping makes it possible to calculate the matrix $\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C}$ on the left-hand side of (10). It also offers the advantage that the velocity field can be obtained with fewer iterations using an iterative solver, because the resulting matrix is more diagonally dominant.

The matrix $\mathbf{I} - \Delta t\,\alpha \mathbf{M}^{-1}\mathbf{K}(\mathbf{V}^n)$ is symmetric and positive definite and is thus endowed with desirable properties for the use of a conjugate gradient solver. However, its size is quite large since $\mathbf{V}$ contains the nodal values of all three velocity components. In order to work with smaller systems, we subdivide the matrix $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{K}^1 + \mathbf{K}^2, \tag{15}$$

where $\mathbf{K}^1$ is made of the diagonal square submatrices corresponding to individual velocity components, while $\mathbf{K}^2$ corresponds to those terms of $\mathbf{K}$ which couple the various velocity components. We have thus slightly modified (14), which now becomes

$$[\mathbf{I} - \Delta t\,\alpha \mathbf{M}^{-1}\mathbf{K}^1(\mathbf{V}^n)]\mathbf{V}^{n+1} = \mathbf{V}^n + \Delta t\,\mathbf{M}^{-1}[\mathbf{f}^n + (1-\alpha)\mathbf{K}^1(\mathbf{V}^n)\mathbf{V}^n + \mathbf{K}^2(\mathbf{V}^n)\mathbf{V}^n - \mathbf{N}(\mathbf{V}^n)\mathbf{V}^n - \mathbf{C}\tilde{\mathbf{P}}]. \tag{16}$$

We note that the system (16) may be solved through three much smaller subsystems with symmetric and positive definite matrices.

Once $\mathbf{V}^{n+1}$ has been calculated, it is interesting to verify how well our hypothesis (12) is satisfied by the solution of (16). Applying the operator $\mathbf{C}^{\mathrm{T}}$ on both sides of (16), we find, with the help of (10).

$$\mathbf{C}^{\mathrm{T}}\mathbf{V}^{n+1} = \alpha \mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{K}^1(\mathbf{V}^n)(\mathbf{V}^{n+1} - \mathbf{V}^n)\Delta t. \tag{17}$$

The error on the divergence of $\mathbf{V}^{n+1}$ is of order Δt^2; it vanishes in the steady state.

The systems (10) and (16) are solved by means of an iterative preconditioned conjugate gradient method as described in Reference 10. The preconditioning is an incomplete Cholesky decomposition coupled with a diagonal scaling.[12]

3. CALCULATION OF THE FREE SURFACE

In extrusion problems a part of the flow domain is surrounded by a free surface which is generally expressed by an equation of the form

$$h(\mathbf{x}, t) = 0. \tag{18}$$

Let $\mathbf{n}(\mathbf{x},t)$ denote the outer normal to the surface; the surface force acting on it is given by

$$\mathbf{t}=[-p_g+\sigma(1/R_1+1/R_2)]\mathbf{n}, \tag{19}$$

where p_g is the pressure in the outer gas, σ is the surface tension coefficient and $1/R_1+1/R_2$ denotes the Gaussian curvature. We will assume in the present paper that σ vanishes; a finite element representation of surface tension in three-dimensional problems may be found in Reference 17.

The kinematic condition expresses that the free surface is a material surface, i.e.

$$\frac{\mathrm{D}}{\mathrm{D}t}h(\mathbf{x},t)=\frac{\partial h}{\partial t}+\mathbf{v}\cdot\nabla h=0. \tag{20}$$

In order to discretize the kinematic condition, let us decompose the surface into a number of faces on which we may write (18) as either

$$h(\mathbf{x},t)=z-f(x,y,t)=0 \tag{21}$$

or

$$h(\mathbf{x},t)=y-g(x,z,t)=0, \tag{22}$$

where we select the Cartesian co-ordinate x in the direction of the jet. In Figure 1 we show the typical example of a jet with a convex cross-section where both faces are easily identified. For more general problems we may consider an arbitrary number of faces for describing the surface of the jet.

Let us consider in more detail the discretization and time evolution of a face given in the form (21). Equation (20) then becomes

$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}-w=0, \tag{23}$$

where u, v, w are velocity components in the x-, y- and z-directions respectively. A similar analysis holds for a surface given by (22). We select a finite element representation for the function f given as

$$f(x,y,t)=\sum_{i=1}^{k} f_i(t)\,\phi_i(x,y), \tag{24}$$

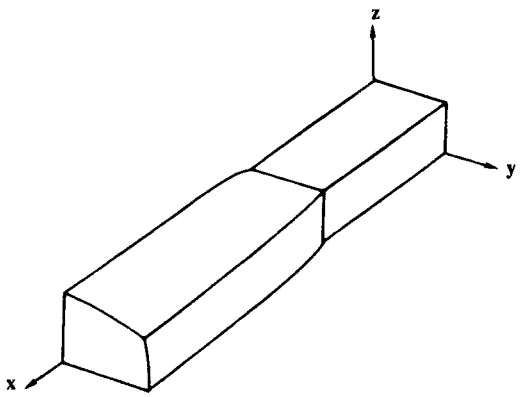

Figure 1. General configuration of extrusion die and extrudate

where k is the number of nodes on the face, $f_i(t)$ are the z-co-ordinates of the nodes and ϕ_i are P^1–C^0 shape functions. Applying the Galerkin formulation to (23), we obtain a set of equations of the form

$$\mathbf{M}'\dot{\mathbf{F}}+\mathbf{N}'(\mathbf{V})\mathbf{F}-\mathbf{M}'\mathbf{W}=\mathbf{0}, \tag{25}$$

where $\mathbf{F}$, $\mathbf{V}$ and $\mathbf{W}$ denote the vectors of nodal values of f, u, v and w on the face, $\mathbf{N}'(\mathbf{V})$ denotes the advective matrix and $\mathbf{M}'$ stands for the mass matrix. The time discretization of (25) is written as

$$\mathbf{M}'(\mathbf{F}^{n+1}-\mathbf{F}^n)/\Delta t+\mathbf{N}'(\mathbf{V}^{n+1})\mathbf{F}^{n+1}-\mathbf{M}'\mathbf{W}^{n+1}=\mathbf{0}. \tag{26}$$

The system (26) contains a small number of variables as compared to the total number of unknowns. Its size justifies the use of a direct solver for calculating $\mathbf{F}^{n+1}$.

The free surface algorithm has been tested on the classical problem of Newtonian extrudate swelling from a slit die, with an infinite extent in the transverse direction. Figure 2(a) shows the finite element mesh together with the shape of the free surface calculated by means of the Galerkin formulation (25) of the kinematic condition. One observes the appearance of small wiggles on the free surface near the exit of the die. We relate such wiggles to the fact that the kinematic condition (23) is of the hyperbolic type, for which the Galerkin formulation is not optimal. We have thus introduced the streamline upwind/Petrov–Galerkin formulation (SUPG)[18] for calculating the discretized kinematic equation (25). Figure 2(b) shows the shape of the free surface obtained with SUPG instead of Galerkin; we find that the free surface does not exhibit the wiggles shown in Figure 2(a).

The system (26) is solved separately for each face of the jet. In order to describe the procedure used for obtaining the total surface of the jet at time t_{n+1}, let us concentrate on Figure 3 to the outer surface of a jet flowing through a die of rectangular shape. We first calculate the co-ordinates z_i of the nodes of face 1. Next we calculate the co-ordinates y_i of the nodes of face 2. We note that the corner node belongs to both faces and receives two locations at time t_{n+1}. The final position of the corner is obtained by extrapolation. Next we relocate the nodes on the deformed faces along geometrical proportions selected at the outset. The nodes in a cross-section with constant x are then relocated on the basis of the outer surface nodes by means of either a proportionality rule or a Laplacian distribution. Since we are not interested in the transient solution of the problem, and in view of the smallness of the time steps, we find it unnecessary to recalculate the velocity field at the relocated nodes. The interest of subdividing the outer surface into a number of faces is that the corner nodes are free to move arbitrarily and are not bound to slide along a preselected spine.

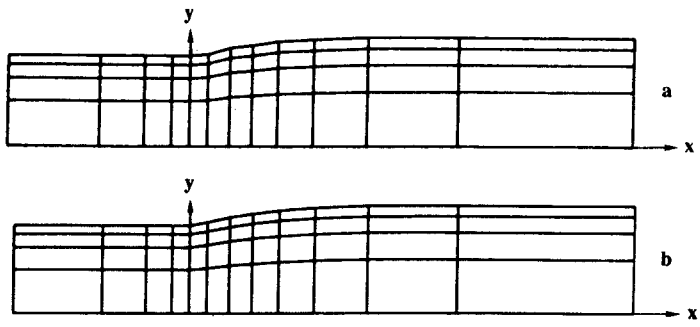

Figure 2. Calculation of planar die swell when free surface is calculated (a) without and (b) with a streamline upwind/Petrov–Galerkin formulation

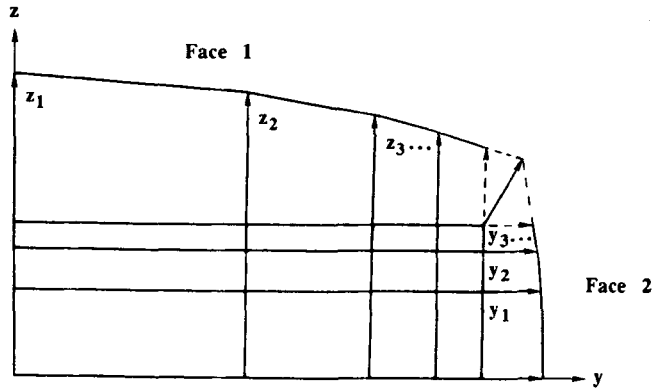

Figure 3. Relocation of surface nodes and corner node in cross-section of extrudate

The time-marching technique now consists of the following steps.

(i) At time t_n, calculate the pressure field $\tilde{\mathbf{P}}$ by solving (10).
(ii) Calculate the velocity field $\mathbf{V}^{n+1}$ by solving (16).
(iii) Calculate the new location of the free surface $\mathbf{F}^{n+1}$ by solving (26).
(iv) Relocate the nodes on the free surface and modify the finite element mesh.

4. SELECTION OF THE TIME STEP

An appropriate selection of the time step is a major ingredient for our transient calculation towards a steady state. In the early stage of our work we performed a number of calculations with a fixed step; we found it difficult to choose an appropriate time step for reducing the computer time while maintaining the stability of the algorithm.

The use of a predictor–corrector scheme has allowed us to implement a dynamic selection of the time step based on the comparison between the predicted and corrected values at the last iteration.[15] Let us briefly review its background. Let $\mathbf{x}(t)$ denote a function of time which at $t=0$ takes the value $\mathbf{x}_0$, while $\dot{\mathbf{x}}_0$ and $\ddot{\mathbf{x}}_0$ denote its first and second time derivatives. At time Δt_0 we have

$$\mathbf{x}(\Delta t_0)=\mathbf{x}_0+\Delta t_0\dot{\mathbf{x}}_0+\Delta t_0^2\ddot{\mathbf{x}}_0/2+O(\Delta t_0^3). \tag{27}$$

With a first-order predictor, starting from the correct values at time $t=0$, we have

$$\mathbf{x}^{\mathrm{p}}(\Delta t_0)=\mathbf{x}_0+\Delta t_0\dot{\mathbf{x}}_0, \tag{28}$$

where $\mathbf{x}^{\mathrm{p}}(\Delta t_0)$ denotes the predicted value. By means of a first-order implicit scheme the calculated value $\mathbf{x}^{\mathrm{c}}(\Delta t_0)$ is given by

$$(\mathbf{x}^{\mathrm{c}}-\mathbf{x}_0)/\Delta t_0=(1-\alpha)\dot{\mathbf{x}}_0+\alpha\dot{\mathbf{x}}(\Delta t)=\dot{\mathbf{x}}_0+\alpha\Delta t_0\ddot{\mathbf{x}}_0+O(\Delta t_0^2); \tag{29}$$

thus

$$\mathbf{x}^{\mathrm{c}}=\mathbf{x}_0+\Delta t_0\dot{\mathbf{x}}_0+\alpha\Delta t_0^2\ddot{\mathbf{x}}_0+O(\Delta t_0^3). \tag{30}$$

From (28) and (30) we find that

$$\mathbf{x}^{\mathrm{c}}-\mathbf{x}^{\mathrm{p}}=\alpha\Delta t_0^2\ddot{\mathbf{x}}_0+O(\Delta t_0^3), \tag{31}$$

while from (27) and (30) we have

$$\mathbf{x}^c - \mathbf{x}(\Delta t_0) = (\alpha - \tfrac{1}{2})\Delta t_0^2 \ddot{\mathbf{x}}_0 + O(\Delta t_0^3). \tag{32}$$

Finally, from the last two equations we have, when $\alpha \neq 0$,

$$\mathbf{d}_0 = \mathbf{x}^c - \mathbf{x}(\Delta t_0) = (\alpha - \tfrac{1}{2})/\alpha(\mathbf{x}^c - \mathbf{x}^p) + O(\Delta t_0^3), \tag{33}$$

where $\mathbf{d}_0$ denotes the difference between the calculated and exact values.

If we now perform an additional iteration with a time step Δt_1, we find that the error at time $\Delta t_0 + \Delta t_1$ is given by

$$\mathbf{d}_1 = (\alpha - \tfrac{1}{2})\Delta t_1^2 \ddot{\mathbf{x}}_1 + O(\Delta t_1^3); \tag{34}$$

thus on the basis of (32)–(34) we have

$$\frac{\|\mathbf{d}_1\|}{\|\mathbf{d}_0\|} \approx \frac{\Delta t_1^2 \|\ddot{\mathbf{x}}_1\|}{\Delta t_0^2 \|\ddot{\mathbf{x}}_0\|}, \tag{35}$$

where double bars denote a suitable norm. We wish to select a time step Δt_1 in order to minimize the error $\|\mathbf{d}_1\|$, which we evaluate by means of a scalar ε such that

$$\|\mathbf{d}_1\| = \varepsilon \|\mathbf{x}^c\|. \tag{36}$$

In view of the fact that $\|\ddot{\mathbf{x}}_1\| = \|\ddot{\mathbf{x}}_0\| + O(\Delta t_0)$, we obtain, with the help of (33), when $\alpha > \frac{1}{2}$,

$$\Delta t_1 = \Delta t_0 \left(\varepsilon \frac{\alpha \|\mathbf{x}^c\|}{(\alpha - \tfrac{1}{2}) \|\mathbf{x}^c - \mathbf{x}^p\|} \right)^{1/2}. \tag{37}$$

In our applications we select values of ε typically of the order of 10^{-3}–10^{-4}.

In the present application it is not easy to select a suitable norm because we are calculating variables of a different nature, i.e. the velocity field and the nodal co-ordinates on the free surface. We have resorted to a separate calculation of a time step Δt^v based on the velocity field and a time step Δt^f based on the free surface. Our first approach was to select

$$\Delta t = \text{Min}(\Delta t^v, \Delta t^f), \tag{38}$$

which is mandatory if one wishes to attain a given accuracy for the transient calculation. However, in the extrusion problems to be discussed below, we found that in general Δt^f is much larger than Δt^v on the basis of (37). It is then tempting to perform an iterative calculation towards the steady state with two different time steps: one for the velocity and pressure fields and one for the free surface. Quite clearly, we then lose the accuracy of the transient calculation, but numerical experiments show an important acceleration towards the steady state. The latter is attained when for each separate field the maximum relative error between two successive steps lies below a given scalar of the order of 10^{-6}.

In order to demonstrate the efficiency of our approach, let us consider the typical example of extrusion from a square die. The undeformed finite element mesh is shown in Figure 4. We consider the creeping flow of a Newtonian and a power-law fluid. The advective terms in the momentum equations are not taken into account, but we select a non-vanishing value of ρ for a 'transient' Reynolds number of one. We examine three strategies for the selection of the time step.

(i) We choose a fixed time step Δt such that

$$\Delta t = h/\bar{V}, \tag{39}$$

where h is the smallest element size in the flow direction while $\bar{V}$ is the average velocity in the channel.

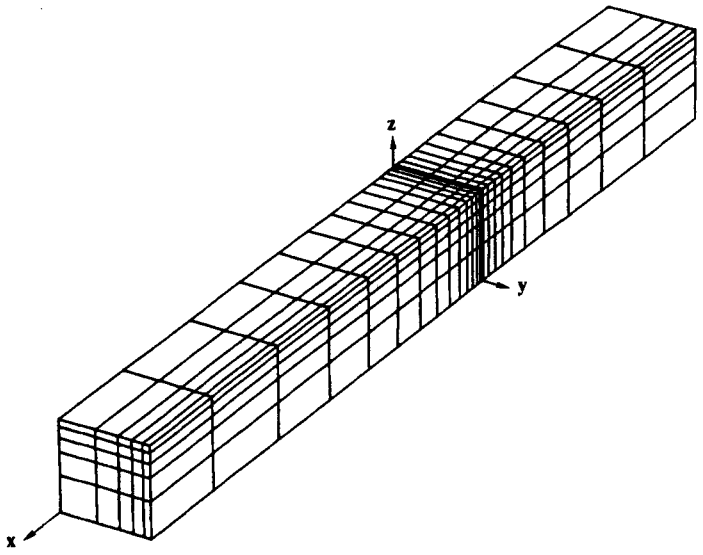

Figure 4. Finite element mesh for testing selection of time step

Table I. Number of time steps and computer time needed for reaching convergence with various iterative schemes

	Newtonian fluid		Power-law fluid	
Method	Iterations	Relative CPU time	Iterations	Relative CPU time
1	642	1·54	>2000	>5·42
2	675	2·04	>2000	>5·85
3	450	1	396	1

(ii) The time step is calculated on the basis of (37) and (38), with $\varepsilon = 10^{-4}$.
(iii) We adopt different time steps for calculating the free surface and the velocity field, with $\varepsilon = 10^{-4}$ for both fields.

In all three cases the relative convergence criterion towards the steady state is 10^{-6}.

Table I shows the number of time steps needed for reaching convergence for a Newtonian as well as a power-law fluid with a power index $m = 0{\cdot}2$. The computer time has been reduced by a factor of four in the Newtonian case and more than five in the power-law case when one uses different time steps instead of (38). These conclusions are evident in Figure 5, where we show the evolution of the transverse dimension of the extrudate as a function of the number of time steps. In Figure 5(a) for the Newtonian case we find that applying (38) leads to a slow approach towards the steady state. The situation is not as bad for a constant time step, but its selection is often an educated guess. However, with different time steps for both fields we find a small initial overshoot followed by a fast approach towards convergence. A similar behaviour is observed in Figure 5(b) for the power-law fluid.

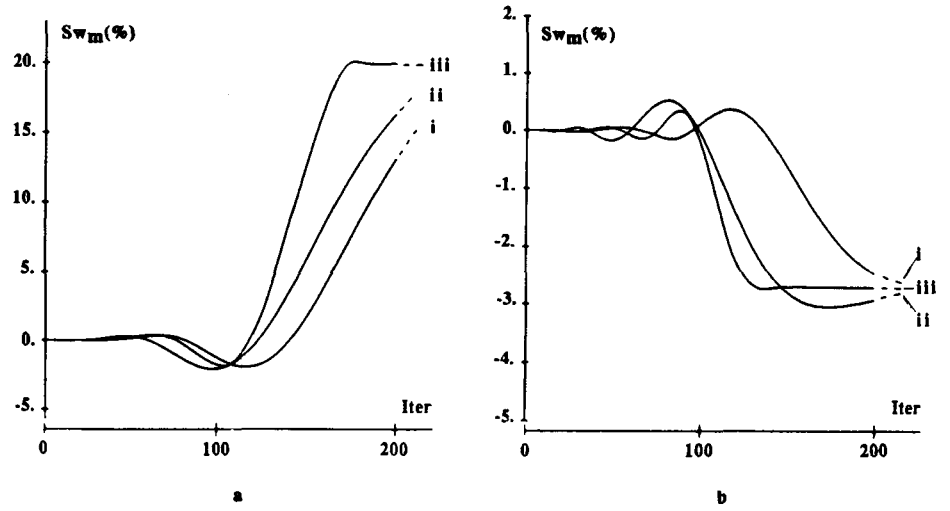

Figure 5. Evolution of swelling ratio as a function of number of iterations for (a) a Newtonian fluid and (b) a power-law fluid: (i) fixed time step; (ii) single optimized time step; (iii) separate optimized time steps

5. TEST PROBLEM: THE CIRCULAR DIE

The numerical simulation of extrusion from a circular die has been available for a long time.[19] For creeping Newtonian flow the calculation gives a swelling ratio of 12.7% with a highly refined mesh. The swelling ratio is defined as the relative increase of the radius of the extrudate to that of the die. Here we solve the same problem with a full three-dimensional geometry. However, we take into account the presence of two planes of symmetry and we perform the calculation on one-fourth of the flow domain.

We impose no-slip conditions on the wall of the tube and a fully developed velocity profile in the entry section. Vanishing normal and tangential surface forces are imposed on the surface of the jet and in the last section. The lengths of the tube and jet are respectively equal to three and four radii of the die.

We have solved the same problem with three finite element meshes with non-uniform spacing. The mesh size is multiplied by two-thirds from mesh 1 to mesh 2 and by one-half from mesh 1 to mesh 3. Figure 6 shows the circular and longitudinal cross-sections, while Table II contains the numerical data and the swelling ratio. The 'element size' corresponds to the ratio of the axial element length at the lip to the radius of the die. For calculating a creeping flow we again omit the advective terms in the momentum equations, but we preserve the velocity time derivatives with an equivalent Reynolds number of one. For all cases the calculation was interrupted when the relative change between two successive time steps was less than 10^{-6}.

With mesh 3 we find a swelling ratio of 13·07%. With a two-dimensional steady state formulation and an equal number of nodes in the plane of symmetry we have found a swelling ratio of 13·04%. In Figure 7 we show the contour lines of the axial velocity component in the plane of symmetry for meshes 1–3 and for the corresponding axisymmetric solution. The comparison is fully satisfactory.

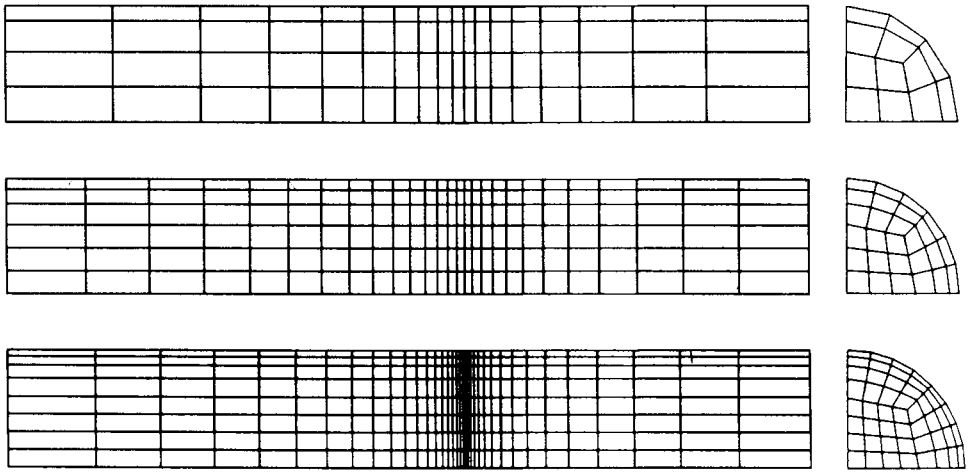

Figure 6. Finite element meshes used for testing accuracy of method with a circular die

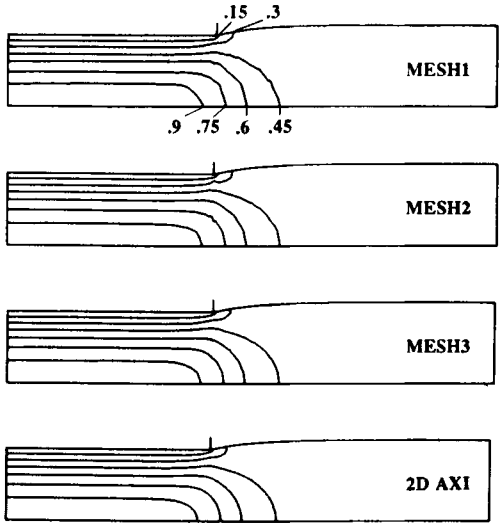

Figure 7. Contour lines of axial velocity field for meshes of Figure 6 and a two-dimensional solution on mesh 3

Table II. Numerical data for circular extrudate swell calculation

Mesh	Nodes	Elements	Degrees of freedom	Element size	Swelling ratio (%)
1	361	216	1354	0·04	13·39
2	1036	729	3949	0·026	13·36
3	2257	1728	8888	0·02	13·07

6. EXTRUSION FROM A SQUARE DIE

We consider the extrusion from a die of square cross-section with sides of length $2H$. The flow domain is characterized by a jet length equal to $6H$ and a die length of $4H$. We impose no-slip conditions on the walls of the die and vanishing surface forces on the jet. In the entry section we impose a biquadratic velocity field, which evolves to a fully developed profile within a short axial distance. In order to verify the convergence of our results, we have used three finite element meshes, which cover one-fourth of the flow domain in view of the symmetry of the problem. In Figure 8 we show the finite element distributions in the axial cross-section and in a plane of symmetry, while the numerical data are collected in Table III. For the square die we calculate two swelling ratios: Sw_m indicates the swelling along the median of the cross-section, while Sw_d indicates the swelling along the diagonal.

6.1. Creeping Newtonian flow

We have calculated the swelling of a creeping Newtonian jet with the three meshes of Figure 8. The swelling ratios are given in Table III, while a view of the deformed mesh based on mesh 2 is given in Figure 9(a). The figures in Table III are a clear indication of the good convergence properties of the method. We note that values of Sw_m and Sw_d of 18% and 3·4% respectively have been obtained in Reference 1 with a boundary element method. In Reference 6, values of 18·4% and 2·9% were found for Sw_m and Sw_d respectively on a mesh with 16 500 degrees of freedom. Mesh 3, with 6971 degrees of freedom, leads to swelling ratios of 18·5% and 2·9%.

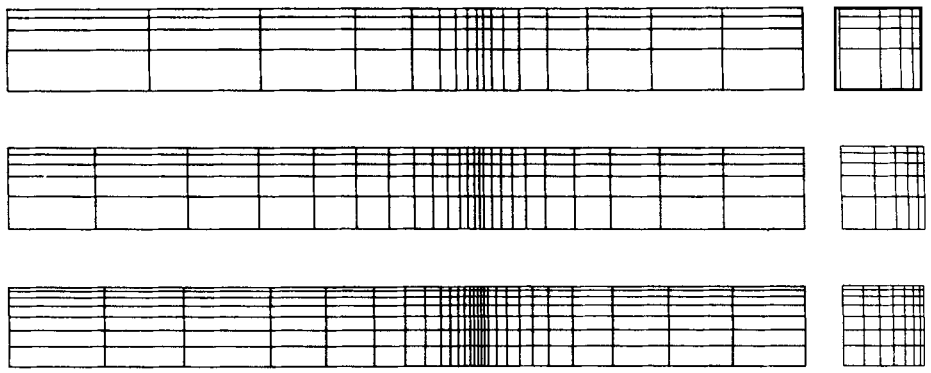

Figure 8. Three finite element meshes used for calculating extrusion from a square die

Table III. Numerical data for square extrudate swell calculation and creeping Newtonian swelling ratios

Mesh	Nodes	Elements	Degrees of freedom	Element size	Sw_m (%)	Sw_d (%)
1	450	272	1722	0·08	19·9	3·5
2	900	600	3480	0·05	18·9	3·0
3	1792	1323	6971	0·03	18·5	2·9

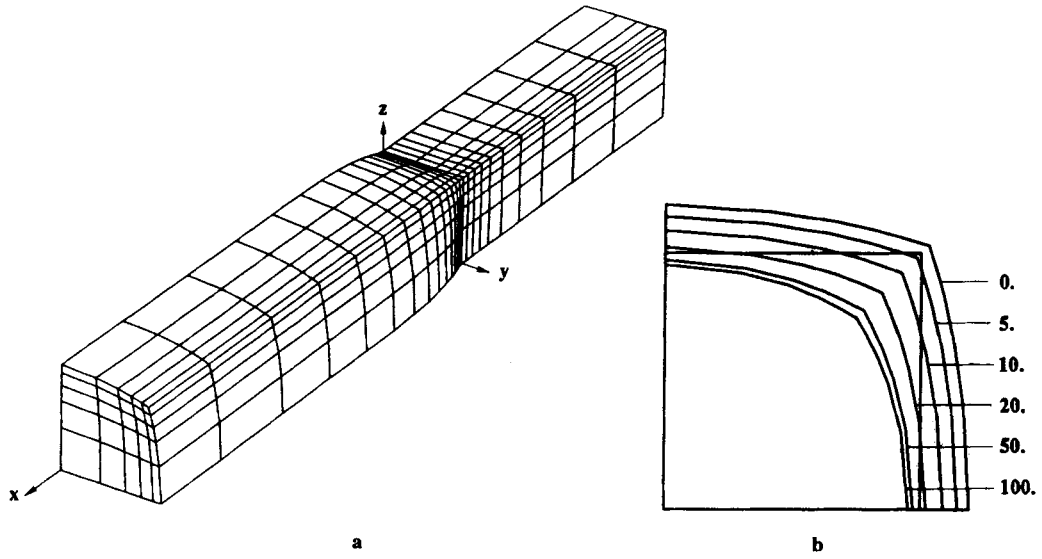

Figure 9. (a) Newtonian extrusion from a square die. (b) Fully developed cross-sections of extrusion for $Re = 0$–100.

6.2. *Newtonian flow with inertia*

The creeping flow results have been obtained by omitting the advective terms in the momentum equations. Let us now examine the interesting effects of inertia upon the shape of the extrudate. In what follows, the Reynolds number has been defined as

$$Re = \rho V_{max} H/\eta, \tag{40}$$

when V_{max} is the maximum axial velocity of the biquadratic inlet velocity profile. A more standard definition of the Reynolds number is given by

$$Re = \rho \bar{V} 2H/\eta, \tag{41}$$

where $\bar{V}$ is the average velocity in the die. By calculating the flow rate in our examples, we find that $\overline{Re} = 0{\cdot}85\,Re$.

The calculations have been performed with mesh 2 shown in Figure 8. In Figure 9(b) we show the successive jet profiles obtained when Re increases from 0 to 100, and the shape of the deformed mesh at $Re = 100$. It is interesting to observe that the jet is approaching a rounded shape at $Re = 50$ and that there is little difference between the shapes obtained at $Re = 50$ and 100. The swelling ratios Sw_m and Sw_d are given in Figure 10 as a function of the Reynolds number.

6.3. *Shear-thinning fluid*

A significant advantage of the present iterative technique is that the computer time required for calculating the flow of a shear-thinning fluid is essentially the same as for a Newtonian fluid. The velocity profile does indeed adapt to the power-law behaviour in the course of the time integration.

We have calculated the extrusion from a square die for a power-law fluid with a viscosity given by equation (3). In the absence of inertia effects the governing non-dimensional parameter is the

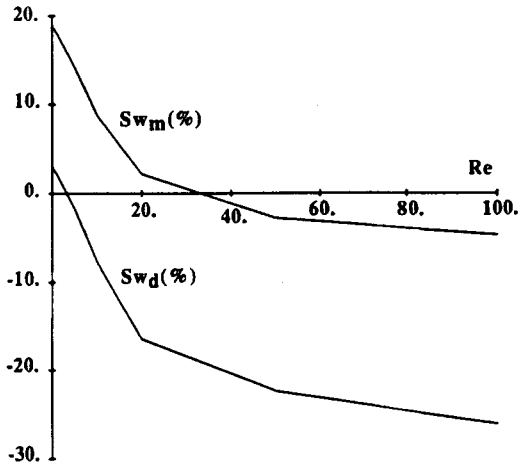

Figure 10. Evolution of swelling ratios as a function of Reynolds number

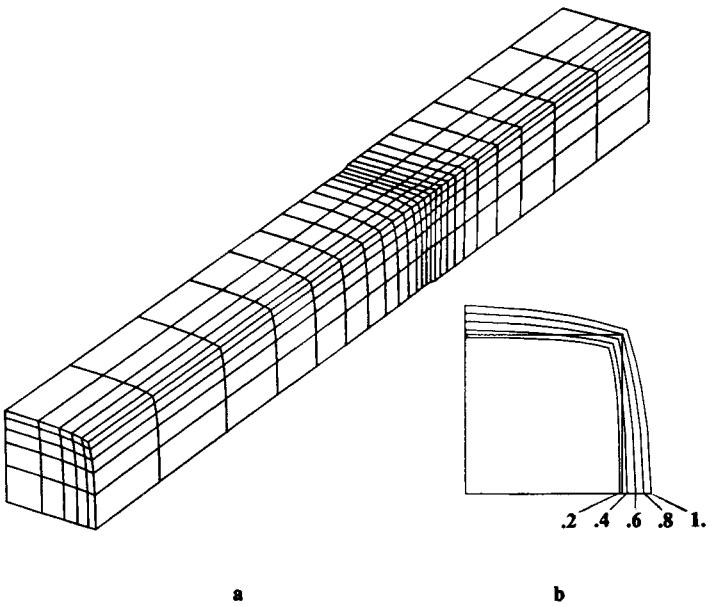

Figure 11. (a) Extrusion of power-law fluid from a square die. (b) Fully developed cross-sections of extrudate as a function of power index

power index m in (3). We have used mesh 2 in Figure 8 for calculating the flow at various values of the power index. In Figure 11 we show the successive shapes of the final cross-section of the jet when m decreases from 1 to 0·2. As we might expect, the shear-thinning behaviour gives rise to a final shape which is approaching a square. This is clearly visible in Figure 11, where we also show the shape of the deformed finite element mesh at $m=0{\cdot}2$. The values of the swelling ratios Sw_m and Sw_d are given in Table IV as a function of the power index.

Table IV. Swelling ratios as a function of power index

Power index m	$Sw_m(\%)$	$Sw_d(\%)$
1	18·9	3·0
0·8	14·0	0·2
0·6	8·7	−2·7
0·4	3·1	−5·8
0·2	−1·7	−8·5

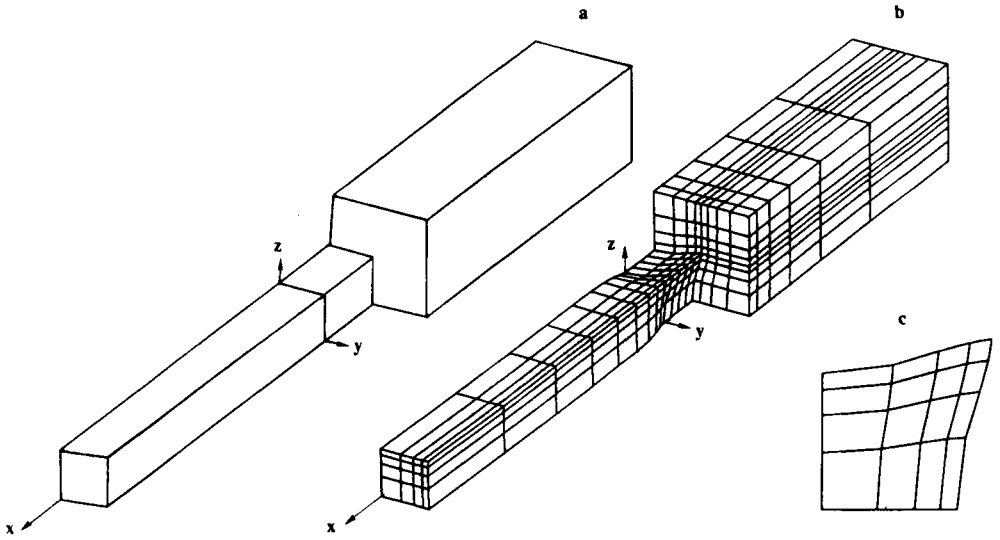

Figure 12. (a) Geometry of die. (b) Converged finite element mesh. (c) Cross-section of die for generating a square extrudate

6.4. Die design

In an earlier paper, Tran-Cong and Phan-Thien[3] have shown that it is necessary to use a star-shaped die in order to obtain a free jet of square cross-section. With the present example we wish to show that our method is able to generate the same type of result, together with the flow through a complex die.

The geometry of the problem is shown in Figure 12(a). A channel of square cross-section is linked to a short die which is itself followed by the jet. We consider a power-law fluid with an index m of 0·7. The initial cross-section of the die is square, while the jet exhibits a convex shape as shown in Figure 9. We can then apply a correction to each external node of the die in a direction opposite to the deviation from a square shape, relocate the internal nodes on the basis of a proportionality rule and recalculate the shape of the extrudate. After five iterations one obtains a relative deviation with respect to a square shape which is less than 10^{-6}.

The final shape of the die and jet is shown in Figure 12(b), while the cross-section of the die is shown in Figure 12(c). As one might expect, the bulging from a square die is compensated by a die with concave faces.

7. EXTRUSION FROM A RECTANGULAR DIE

In the present section we wish to calculate the extrusion from a rectangular die with a 2:1 aspect ratio. The method of Section 3 for describing the motion of the free surface is quite appropriate for a cross-section with corners moving in a direction which is *a priori* unknown. In Reference 6 it was assumed that corners move along previously defined spines; in such a way one removes a necessary degree of freedom for the motion of the corners. In the present paper both faces of the jet can move arbitrarily, while the corner line is defined as their intersection.

For solving the present problem, we have again considered three different meshes; their axial and longitudinal cross-sections are given in Figure 13; their numerical parameters are given in Table V.

In order to evaluate the swelling of the jet with respect to the die, we introduce four swelling ratios defined as follows:

$$Sw_a=(a-L)/L, \qquad Sw_b=(b-H)/H, \qquad Sw_c=(c-L)/L, \qquad Sw_d=(d-H)/H; \tag{42}$$

the lengths a, b, H and L are defined in Figure 14; c and d denote the horizontal and vertical projections of the displaced corner respectively. The values of the swelling ratios for the three meshes are given in Table V. The convergence properties are good, as in earlier examples; it is remarkable to observe that the results with a coarse mesh differ little from those with a refined mesh. A view of the deformed mesh and the final shape of the cross-section obained with mesh 2 are shown in Figure 15.

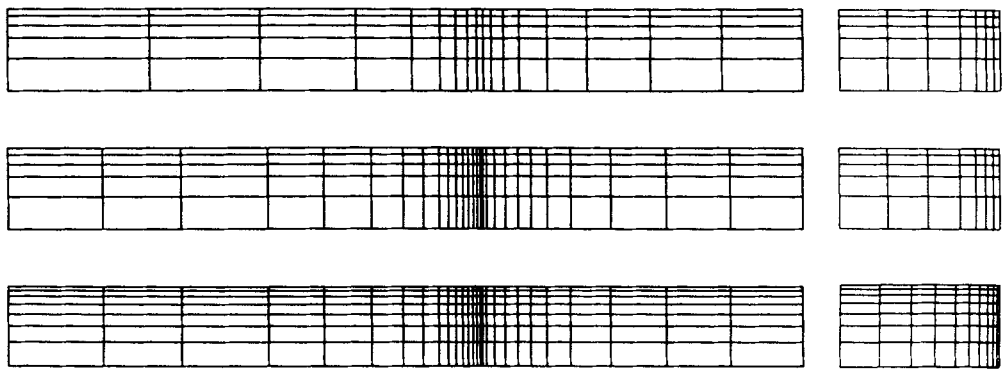

Figure 13. Three finite element meshes for calculating extrusion from a rectangular die

Table V. Numerical parameters and swelling ratios for extrusion from a rectangular die

Mesh	Nodes	Elements	Degrees of freedom	Sw_a (%)	Sw_b (%)	Sw_c (%)	Sw_d (%)
1	864	595	3327	14·5	24·7	9·98	−11·1
2	1344	945	5215	14·1	24·3	9·77	−11·09
3	2464	1890	9605	14·1	23·3	9·72	−10·65

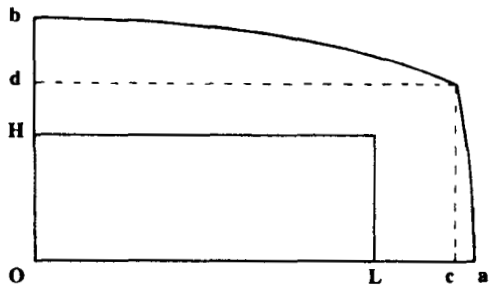

Figure 14. Characteristic geometrical quantities of extrudate from a rectangular die

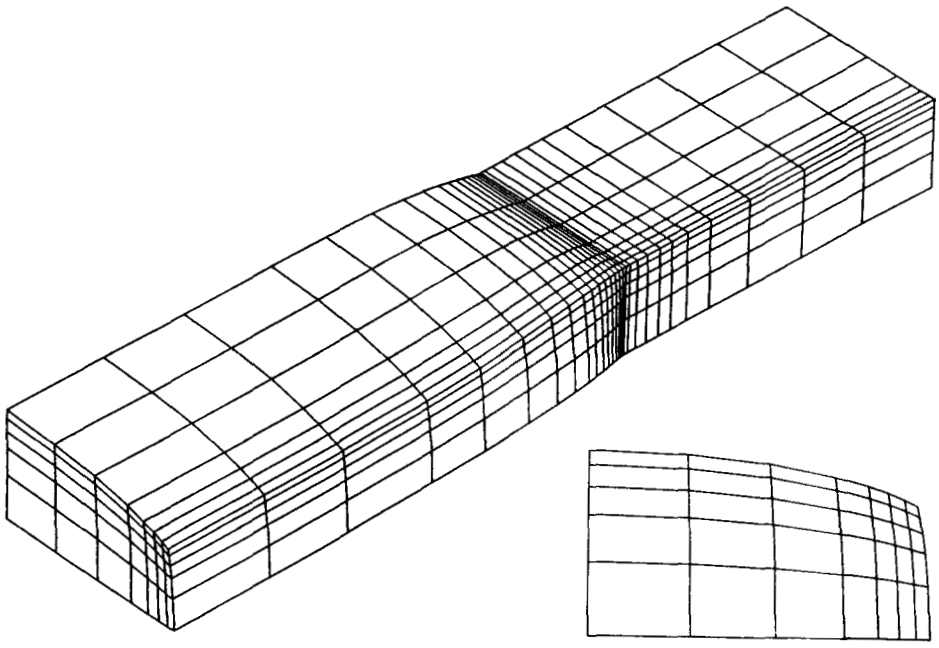

Figure 15. Deformed mesh and fully developed cross-section of extrudate from a rectangular die

8. CONCLUSIONS

We have demonstrated that the algorithm introduced in Reference 10 by Gresho and Chan can be extended to the numerical prediction of three-dimensional free surfaces. The combination of the pseudo-transient marching technique and the conjugate gradient solver allows one to limit the amount of memory needed for solving three-dimensional problems. In particular, the use of different time steps for calculating the velocity field and the free surface reduces the number of iterations needed for obtaining a converged steady state. In a later paper the method will be extended to the simulation of three-dimensional viscoelastic free surfaces, for which the stress field must be calculated together with the velocity and free surface.

ACKNOWLEDGEMENT

One of us (O.W.) wishes to acknowledge partial support from the 'Services de Programmation de la Politique Scientifique' in Belgium.

REFERENCES

1. T. Tran-Cong and N. Phan-Thien, 'Three-dimensional extrusion processes by boundary element method: 1. An implementation of high-order elements and some Newtonian results', *Rheol. Acta*, **27**, 21–30 (1988).
2. T. Tran-Cong and N. Phan-Thien, 'Three-dimensional study of extrusion processes by boundary element method: 2. Extrusion of a viscoelastic fluid', *Rheol. Acta*, **27**, 639–648 (1988).
3. T. Tran-Cong and N. Phan-Thien, 'Die design by a boundary element method', *J. Non-Newtonian Fluid Mech.*, **30**, 37–46 (1988).
4. T. Shiojima and Y. Shimazaki, 'Three-dimensional finite element analysis for a Maxwell fluid using the penalty function method', *Proc. NUMETA Conf.*, Pineridge, Swansea, 1987, pp. D37, 1–8.
5. T. Shiojima and Y. Shimazaki, 'Three-dimensional finite element method for extrudate swells of a Maxwell fluid', *J. Non-Newtonian Fluid Mech.*, **34**, 269–288 (1990).
6. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional extrudate swell of creeping Newtonian jets', *AIChE J.*, **34**, 2088–2094 (1988).
7. H. Saito and L. E. Scriven, 'Study of coating flow by the finite element method', *J. Comput. Phys.*, **42**, 53 (1981).
8. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional non-isothermal extrusion flows', *Rheol. Acta*, **28**, 121–133 (1989).
9. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional studies on bicomponent extrusion', *Rheol. Acta*, **29**, 71–87 (1990).
10. P. M. Gresho and S. Chan 'A new semi-implicit method for solving the time-dependent conservation equations for incompressible flow', in C. Taylor, M. D. Olson, P. M. Gresho and W. G. Habashi (eds), *Numerical Methods in Laminar and Turbulent Flow*, Pineridge, Swansea, 1985, pp. 3–21.
11. P. M. Gresho, 'Time integration and conjugate gradient methods for the incompressible Navier–Stokes equations', *Proc. 6th Int. Conf. on Finite Elements in Water Resources*, Lisbon, 1986, Springer Verlag, pp. 3–29.
12. H. A. Van der Vorst, 'A vectorizable variant of some ICCG methods', *SIAM J. Sci. Stat. Comput.*, **3**, 350–356 (1982).
13. P. M. Gresho, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory', *Int. j. numer. methods fluids*, **11**, 587–620 (1990).
14. P. M. Gresho and S. T. Chan, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation', *Int. j. numer. methods fluids*, **11**, 621–660 (1990).
15. P. Gresho, R. Lee and R. Sani, 'On the time-dependent solution of the incompressible Navier–Stokes equations in two and three dimensions', in *Recent Advances in Numerical Methods in Fluids , Vol. 1*, Pineridge, Swansea, 1980, pp. 27–81.
16. O. C. Zienkiewicz, *The Finite Element Method*, 3rd edn, McGraw-Hill, New York, 1977.
17. K. J. Ruschak, 'A three-dimensional linear stability analysis for two-dimensional free boundary flows by the finite-element method', Comput. Fluids, **11**, 391–402 (1983).
18. A. Brooks and T. Hughes, 'Streamlines upwind/Petrov Galerkin formulation for convection-dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comput. Methods Mech. Eng.*, **32**, 199–259 (1982).
19. R. E. Nickell, R. I. Tanner and B. Caswell, 'The solution of viscous incompressible jet and free-surface flows using finite-elements methods', *J. Fluid Mech.*, **65**, pt. 1, 189–206 (1974).