INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:695–720 (DOI: 10.1002/fld.581)

Iterative solvers for quadratic discretizations of the generalized Stokes problem

R. Guénette*,†, A. Fortin, J. Labbé and J. P. Marcotte

Département de Mathématiques et de Statistique, GIREF, Université Laval, Qué., Canada G1K 7P4

SUMMARY

We present in this paper various iterative methods for the solution of large linear and non-linear systems resulting from the discretization of the generalized Stokes problem. A second-order ($O(h^2)$) $P_2 - P_1$ mixed finite element is used for the approximation of the velocity and the pressure. Solution strategies based on conjugate gradient-like methods, the Uzawa's and Arrow–Hurwicz's methods are presented. Schur complement methods are also explored in the context of a hierarchical decomposition of the velocity field. The ever present preconditioning problem is also addressed. The performance of these iterative methods will be discussed on complex flows of industrial interest. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: iterative solvers; preconditioning; generalized Stokes problems; mixed finite elements; polymer flows

1. INTRODUCTION

Polymer processes like injection molding, extrusion, coextrusion, calandring, etc., have many applications in the plastic industry. Numerical modelling of such processes is now playing an important role and this role will further increase in the near future. However, current numerical techniques prevent complete, accurate and reliable prediction of these processes. The difficulties are overwhelming: complex three-dimensional geometries, complex rheological behaviour of polymers, instabilities in certain operating conditions, etc.

The need for more effective solvers is clear, especially for three-dimensional problems. In most numerical simulations, polymers are assumed purely viscous and incompressible but even when viscoelastic effects are taken into account, Stokes-like problems are central to the solution. This is why we will focus on the development of efficient and accurate solvers for

*Correspondence to: R. Guénette, Département de Mathématiques et de Statistique, Université Laval, Qué., Canada G1K 7P4.
† E-mail: rguenet@mat.ulaval.ca

Contract/grant sponsor: National Sciences and Engineering Research Council (NSERC) of Canada
Contract/grant sponsor: Programme FCAR of the Province of Québec

Received 12 December 2000
Revised 21 April 2003

generalized Stokes problems. First, let us recall some basic concepts concerning the (generalized) Stokes problem.

Given a flow domain Ω, the problem is to find the velocity $\mathbf{u}$ and the pressure p satisfying

$$\begin{aligned} -\boldsymbol{\nabla}\cdot[2\eta(|\dot{\boldsymbol{\gamma}}(\mathbf{u})|)\dot{\boldsymbol{\gamma}}(\mathbf{u})]+\boldsymbol{\nabla}p=\mathbf{f} \quad &\text{in } \Omega \\ \boldsymbol{\nabla}\cdot\mathbf{u}=0 \quad &\text{in } \Omega \end{aligned}$$

with appropriated boundary conditions which will be specified for each flow problem of Section 5. The viscosity η is a function of the second invariant

$$|\dot{\boldsymbol{\gamma}}(\mathbf{u})|^2=2\dot{\boldsymbol{\gamma}}(\mathbf{u}):\dot{\boldsymbol{\gamma}}(\mathbf{u})=2\sum_{ij}\dot{\boldsymbol{\gamma}}(\mathbf{u})_{ij}\dot{\boldsymbol{\gamma}}(\mathbf{u})_{ij}$$

of the rate of strain tensor

$$\dot{\boldsymbol{\gamma}}(\mathbf{u})=\frac{\boldsymbol{\nabla}\mathbf{u}+(\boldsymbol{\nabla}\mathbf{u})^{\mathrm{T}}}{2}$$

The following viscoplastic behaviour will be considered in this paper

$$\eta(|\dot{\boldsymbol{\gamma}}(\mathbf{u})|)=\begin{cases}\eta_0, & \text{Newtonian case}\\ \eta_0|\dot{\boldsymbol{\gamma}}(\mathbf{u})|^{n-1}, & \text{power law}\\ \eta_0(1+\lambda^2|\dot{\boldsymbol{\gamma}}(\mathbf{u})|^2)^{(n-1)/2}), & \text{Carreau model}\end{cases}$$

where η_0, λ and n are rheological constants determined through curve of viscosity data. In the Newtonian case ($n=1$), the viscosity η_0 is constant while in the viscoplastic case ($0<n<1$), the viscosity depends on the velocity field, introducing a non-linearity. We will now consider the non-linear case.

For simplicity, we suppose that we have homogeneous boundary conditions $\mathbf{u}=0$ on the boundary Γ. Multiplying the conservation of momentum and mass equations by test functions $\mathbf{v}$ and q in appropriate functional spaces V and Q, and integrating by parts, the variational form is given by: find $\mathbf{u}\in V$ and $p\in Q$ satisfying

$$\begin{aligned} \int_\Omega 2\eta(|\dot{\boldsymbol{\gamma}}(\mathbf{u})|)\dot{\boldsymbol{\gamma}}(\mathbf{u}):\dot{\boldsymbol{\gamma}}(\mathbf{v})\,\mathrm{d}V-\int_\Omega p\boldsymbol{\nabla}\cdot\mathbf{v}\,\mathrm{d}V&=\int_\Omega\mathbf{f}\cdot\mathbf{v}\,\mathrm{d}V\\ \int_\Omega q\boldsymbol{\nabla}\cdot\mathbf{u}\,\mathrm{d}V&=0\end{aligned}$$

The Newton method is then applied in order to linearize the problem. Starting from an approximate solution $(\mathbf{u}_0, p_0)$ a Newton step requires the calculation of a correction $(\delta\mathbf{u}, \delta p)$ solution of

$$\begin{aligned} a_{u_0}(\delta\mathbf{u},\mathbf{v})-\int_\Omega\delta p\boldsymbol{\nabla}\cdot\mathbf{v}\,\mathrm{d}V&=\mathbf{R}_u((\mathbf{u}_0,p_0),\mathbf{v})\\ \int_\Omega q\boldsymbol{\nabla}\cdot\delta\mathbf{u}\,\mathrm{d}V&=\mathbf{R}_p((\mathbf{u}_0,p_0),q)\end{aligned} \tag{1}$$

where, assuming a Carreau model, the symmetric bilinear form a_{u_0} and the residual $R((u_0, p_0), v)$ are defined by

$$\begin{aligned} a_{u_0}(\delta\mathbf{u}, \mathbf{v}) &= \int_\Omega (2\eta_0(1+\lambda^2|\dot{\boldsymbol{\gamma}}(\mathbf{u}_0)|^2)^{(n-1)/2}\dot{\boldsymbol{\gamma}}(\delta\mathbf{u}) : \dot{\boldsymbol{\gamma}}(\mathbf{v}))\,\mathrm{d}V \\ &\quad + \int_\Omega (4\eta_0(n-1)\lambda^2(1+\lambda^2|\dot{\boldsymbol{\gamma}}(\mathbf{u}_0)|^2)^{(n-3)/2}(\dot{\boldsymbol{\gamma}}(\mathbf{u}_0) : \dot{\boldsymbol{\gamma}}(\delta\mathbf{u}))(\dot{\boldsymbol{\gamma}}(\mathbf{u}_0) : \dot{\boldsymbol{\gamma}}(\mathbf{v})))\,\mathrm{d}V \\ R_u((\mathbf{u}_0, p_0), \mathbf{v}) &= \int_\Omega \mathbf{f}\cdot\mathbf{v}\,\mathrm{d}V + \int_\Omega p_0\boldsymbol{\nabla}\cdot\mathbf{v}\,\mathrm{d}V - \int_\Omega 2\eta_0(1+\lambda^2|\dot{\boldsymbol{\gamma}}(\mathbf{u}_0)|^2)^{(n-1)/2}\dot{\boldsymbol{\gamma}}(\mathbf{u}_0) : \dot{\boldsymbol{\gamma}}(\mathbf{v})\,\mathrm{d}V \\ R_p((\mathbf{u}_0, p_0), q) &= -\int_\Omega q\boldsymbol{\nabla}\cdot\mathbf{u}_0\,\mathrm{d}V \end{aligned} \tag{2}$$

The approximate solution is then updated:

$$\mathbf{u}_1 = \mathbf{u}_0 + \delta\mathbf{u} \quad p_1 = p_0 + \delta p$$

and a new correction $(\delta\mathbf{u}, \delta p)$ is computed until convergence.

The bilinear form a_{u_0} is obviously symmetric but not positive definite unless we suppose the existence of a positive second plateau for the viscosity. This implies that the viscosity does not tend toward 0 even if $|\dot{\boldsymbol{\gamma}}(\mathbf{u})|^2$ is very large. We will thus make this assumption with no practical consequences in applications. In the linear case ($n = 1$) we recover the classical Stokes problem.

2. THE DISCRETE PROBLEM

The discretization of system (2) is delicate and requires discrete spaces V_h and Q_h satisfying a compatibility condition known as the inf–sup condition as described in Brezzi–Fortin [1]. Two stable discretizations will be considered in this work. The first one is a classical choice for three-dimensional problems because of its low number of degrees of freedom and relative accuracy. It is called the Mini element (see Reference [2]) and is presented in Figure 1. The velocity is linear (P_1) but enriched by a so-called bubble function of degree 4 attached to the barycentre of each element. The degrees of freedom associated with the bubble can be condensed out through classical Gaussian elimination (static condensation). A continuous linear approximation is used for the pressure. The Mini element satisfies the inf–sup condition and converges linearly ($O(h)$). In the next section, we will present a specific solver to be used with the Mini element. Convergence of the algorithm is satisfactory but as we shall see, the computed solutions are not fully satisfactory. Artificial boundary layers are present polluting the pressure solution. This is one of the reasons to consider second-order elements.

The second element is the well-known quadratic mixed element referred to as the Taylor–Hood element (see Figure 2). A continuous $P_2 - P_1$ is used for velocity and pressure fields. It is second-order accurate ($O(h^2)$) and we will focus on the development of numerical methods for this element. For reasons to be explained later, we will consider a hierarchical basis for the space of quadratic polynomials. A quadratic velocity field $\mathbf{u}$ can thus be decomposed into

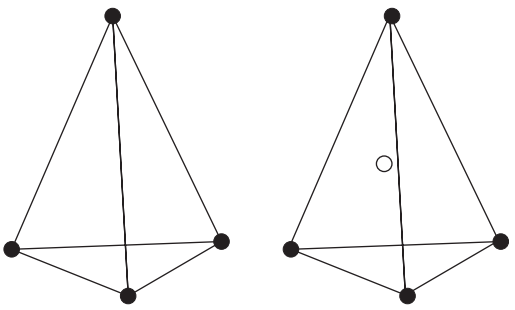

Figure 1. The Mini element.

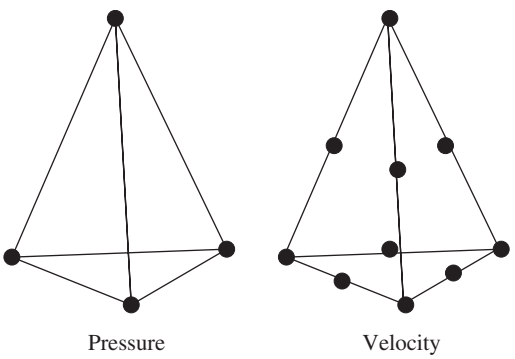

Figure 2. The Taylor–Hood ($P_2 - P_1$) element.

a linear part $\mathbf{u}_\mathrm{l}$ and a quadratic correction part $\mathbf{u}_\mathrm{q}$. In other words, we have

$$V_h = L_h \oplus W_h$$

where L_h is the space of continuous piecewise linear polynomials and W_h is a space of continuous piecewise quadratic corrections. The linear part $\mathbf{u}_\mathrm{l}$ is associated to the vertices of the tetrahedron while the quadratic part $\mathbf{u}_\mathrm{q}$ is related to the edges. This decomposition will be very useful for some iterative procedures that will be considered later.

The main objective of this paper is to develop efficient iterative solvers for quadratic element within the context of incompressible flows. Among conformal second-order mixed finite element for Stokes problem, Taylor–Hood is the cheapest element having the least degrees of freedom. Hence, we will focus on numerical strategies for this element. Very effective iterative schemes can be found in the literature (see Reference [3] for a good survey). The most effectives solvers are designed for low-order finite elements such as the $P_1 - P_1$ and makes use of multigrid or multilevel approach. Moreover, the implantation of these solvers is complicated and needs sophisticated expertise. The underlying goal is to propose efficient iterative solvers which are relatively easy to implement and applicable by third party.

For the Taylor–Hood element, the discrete system arising from the linearized generalized Stokes problem (1) can be written in matrix form as

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta p \end{bmatrix} = \begin{bmatrix} R_u \\ R_p \end{bmatrix} \tag{3}$$

where R_u and R_p refer to the residual vector to the velocity and the pressure variable, respectively. Let us recall that the matrix A is symmetric, positive definite and may depend on a previous solution $\mathbf{u}_0$, while the matrix B is rectangular. The complete system is however indefinite.

Iterative methods using Mini elements will also be needed for preconditioning reasons. Let us describe the matrix form associated with the Mini element. In principle, this discretization admits a similar form to (3). As mentioned above, the degrees of freedom are reduced to those associated with the linear part of the velocity and the pressure, i.e. equal order P_1 approximation. In term of these degrees of freedom, the system can be written as

$$\begin{bmatrix} A & B^t \\ B & -C \end{bmatrix} \begin{bmatrix} \delta u \\ \delta p \end{bmatrix} = \begin{bmatrix} R_u \\ R_p \end{bmatrix} \tag{4}$$

We still denote by the letters A and B the corresponding diffusion matrix and the discrete divergence matrix coming from the P_1 approximation.

The stabilization matrix C comes from the condensation process of the bubble functions associated to the barycentre of each element. It can also be seen as a stabilization term of the form

$$\sum_K \alpha_K \int_K \nabla p \cdot \nabla q \, \mathrm{d}v$$

with specific values α_K depending on the element as described in Reference [4]. The global matrix of system (4) is symmetric but still indefinite.

Before moving to iterative strategies, a few more remarks are necessary on the condensation process. In the non-linear case, one cannot eliminate the degrees of freedom associated to the bubble functions in a straightforward fashion. One way to cope with this problem is to neglect the influence of the bubble term in the non-linear expression of the viscosity. This special treatment has been first proposed by Coupez [5] and can be interpreted properly in terms of a three fields problem:

Find $(\mathbf{u}, p, \mathbf{d})$, where $\mathbf{d}$ stands for a piecewise constant tensor, satisfying

$$\int_\Omega 2\eta(|\mathbf{d}|)\dot{\boldsymbol{\gamma}}(\mathbf{u}) : \dot{\boldsymbol{\gamma}}(\mathbf{v}) \, \mathrm{d}V - \int_\Omega p \boldsymbol{\nabla} \cdot \mathbf{v} \, \mathrm{d}V = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}V$$

$$\int_\Omega q \boldsymbol{\nabla} \cdot \mathbf{u} \, \mathrm{d}V = 0$$

$$\int_\Omega \mathbf{d} : \mathbf{e} \, \mathrm{d}V = \int_\Omega \dot{\boldsymbol{\gamma}}(\mathbf{u}) : \mathbf{e} \, \mathrm{d}V$$

where $a:b=\sum_{i,j} a_{ij}b_{ij}$ stands for the usual double contraction operation on second-order tensors.

The last equation corresponds to a projection onto the space of symmetric tensor constant per element. Obviously, $\dot{\boldsymbol{\gamma}}(\mathbf{u}_l)$ is the solution of this projection problem where $\mathbf{u}_l$ is the linear part of the velocity. Similar ideas have been used for the simulation of viscoelastic fluids [6].

In this paper, we will present several iterative schemes for solving the (generalized) Stokes problem based on quadratic discretizations. These methods will be tested on classical three-dimensional benchmark flow problems. The most promising strategies will be applied to large-scale industrial 3D flows.

An outline of the paper is as follows. In Section 3, we shall present iterative solvers based on the usual velocity–pressure decomposition of the global system. The next section is devoted to iteratives methods based on the hierarchical decomposition of the quadratic finite element. Comparison between the various solvers and numerical results are presented in the last section.

3. ITERATIVE METHODS BASED ON THE (u, p) DECOMPOSITION

In the following we present different solution strategies for solving the linearized Stokes problems (3). In this section, u and p will stand for the correction variables with homogenous boundary conditions. This discrete problem takes the form of a saddle-point problem

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{5}$$

Several iterative schemes have been proposed in the literature for solving saddle-point problems, see References [7–12].

3.1. Uzawa algorithm

Uzawa's method is classical for solving saddle point problems. The algorithm can be written as

$$u_{k+1} = A^{-1}(f - B^t p_k)$$

$$p_{k+1} = p_k + \rho(Bu_{k+1} - g)$$

where ρ is a sufficiently small positive parameter. This algorithm can be interpreted as a gradient method on the dual problem, see Reference [13].

Elimination of u_{k+1} in the first equation leads to the following relation:

$$p_{k+1} = p_k - \rho(g - BA^{-1}f + BA^{-1}B^t p_k) \tag{6}$$

which is nothing but a fixed point-strategy for solving the linear problem

$$BA^{-1}B^t p = BA^{-1}f - g$$

The convergence of the classical Uzawa method is notoriously slow. To speed up the convergence, one need to introduce a preconditioner of the form

$$\begin{aligned} u_{k+1} &= A^{-1}(f - B^t p_k) \\ M(p_{k+1} - p_k) &= Bu_{k+1} - g \end{aligned} \tag{7}$$

where M is a preconditioning matrix assumed to be symmetric and positive definite.

In practice, the acceleration of the simple iteration (7) is done by applying the conjugate gradient method to system (6) preconditioned by the matrix M. The success of this method depends heavily on the spectral properties of the Schur complement matrix $BA^{-1}B^t$. Indeed, for stable discretizations of the Stokes problem, it can be shown that there exists bounds $\alpha > 0$ and $\beta > 0$ with

$$\alpha \leqslant \frac{(BA^{-1}B^t p, p)}{(M_p p, p)} \leqslant \beta \quad \forall p \neq 0$$

where M_p stands for the pressure mass matrix, see Reference [1]. Since the condition number of M_p is independent of h, the above inequality implies that the condition number of $BA^{-1}B^t$ is also independent of h. This makes the preconditioned conjugate gradient method very effective for solving the saddle-point problem (5).

Let us now present the principal ingredients for solving the complete problem:

$$M^{-1}BA^{-1}B^t p = M^{-1}(BA^{-1}f - g) \tag{8}$$

First, the right-hand side is computed:

1. Solve $Ad_1 = f$;
2. compute the product Bd_1 and put the result in vector d_2;
3. solve $Md_3 = d_2 - g$ and return d_3.

To apply the PCG method to problem (8), we have to define the product of the system matrix with a descent vector d. In our case, this product is done in many steps. For a given vector d (provided by the iterative algorithm) of the same dimension as p:

1. Compute the product $B^t d$ and put the result in vector d_1;
2. solve $Ad_2 = d_1$;
3. compute the product Bd_2 and put the result in vector d_3;
4. solve $Md_3 = d_4$ and return d_4.

(Note that in the above algorithm, vectors d_i are destroyed as soon as possible.) The preconditioning matrix M was chosen as the lumped diagonal mass matrix. This diagonal matrix is known to be spectrally equivalent to the pressure mass matrix. Hence, the nice convergence of the solver is essentially unchanged, see Reference [10]. For large-scale problem, the preconditioned (Jacobi, SSOR) conjugate gradient method was also applied to the solution of the velocity problem (step 2).

3.2. Preconditioned conjugate residual method

For symmetric indefinite systems such as (4) and (5), there is an iterative solver that was first advocated by Wathen and Silvester in a series of two papers [10, 11]. They have used the preconditioned conjugate residual method for solving the classical Stokes problem. This strategy was used by Coupez and Marie [14] for large-scale problems arising in the plastic industry (polymer flows).

Although the Uzawa algorithm is a reliable method for solving Stokes-like problems, it can be very costly since the inner problem A^{-1} must be solved very accurately. Indeed, in the last section, we will show that if we restrict the number of iterations for the inner iteration, the overall performance of the solver is considerably reduced. Hence, for large problems, a non-nested iterative strategy could be very attractive. Let us describe briefly the iterative method used by Wathen and Silvester. The symmetric indefinite system is solved globally by a Krylov subspace method known as the conjugate residual method (CR) or minimal residual method (MINRES) which can be derived from the GMRES method for the special case where the matrix is symmetric. This leads to a three-terms recurrence algorithm applicable to symmetric indefinite systems (see Reference [15] for more details).

Following the second paper of Wathen and Silvester [11], we have used a block preconditioner based on the natural decomposition of the velocity and pressure variables

$$\mathscr{M} = \begin{bmatrix} M_A & 0 \\ 0 & M_C \end{bmatrix}$$

where the matrix M_A is an approximation of the matrix A and the matrix M_C is spectrally equivalent to the mass pressure matrix in the case of the Taylor–Hood element or the stabilization matrix C for the Mini element.

3.3. Arrow–Hurwicz algorithm

The main drawback of the Uzawa's method (7) is the need to solve as exactly as possible the first equation. If an exact or a very accurate velocity solver is available, this is the method of choice. But for large-scale computations, an iterative solver must be chosen to compute the velocity at each iteration of Uzawa and a large number of iterations will be needed to obtain a good precision.

The starting point for the class of methods that will be introduced is the Arrow–Hurwicz's method for solving a saddle-point problem (see Reference [13]). The method as originally proposed can be viewed as a way to obtain steady-state solution from a time-dependent approach

$$\begin{aligned} \frac{\partial u}{\partial t} + Au + B^t p - f = 0 \\ \frac{\partial p}{\partial t} - Bu + g = 0 \end{aligned} \tag{9}$$

Using an explicit-implicit scheme for the time discretization of the above evolution system, this leads to the classical Arrow–Hurwicz algorithm

$$\begin{aligned} u_{k+1} &= u_k + \alpha(f - Au_k - B^t p_k) \\ p_{k+1} &= p_k + \rho(Bu_{k+1} - g) \end{aligned}$$

where α and ρ are positive parameters viewed as pseudo-time step combined with a scaling factor.

A class of preconditioned versions of the Arrow–Hurwicz scheme is given by

$$M_A(u_{k+1} - u_k) = f - Au_k - B^t p_k$$
$$M(p_{k+1} - p_k) = Bu_{k+1} - g$$

where M_A and M are symmetric positive definite matrices. In other words, the simple iteration looks like

$$\begin{bmatrix} M_A & 0 \\ B & -M \end{bmatrix} \begin{bmatrix} u_{k+1} - u_k \\ p_{k+1} - p_k \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix}$$

In practice, this scheme must be accelerated by a Krylov subspace method operating on the global system. The preconditioner corresponds to the block non-symmetric matrix

$$\mathscr{M} = \begin{bmatrix} M_A & 0 \\ B & -M \end{bmatrix}$$

This choice of $\mathscr{M}$ prevents us to choose a symmetric but indefinite iterative solver such as the conjugate residual method (see References [10, 15] for details on this Krylov subspace method). Instead, we have used the BiCGSTAB method first proposed by van der Vorst [16]. The lost of symmetry is largely compensated by the nice convergence of the solver and by the fact that it extends to the case of the non-symmetric Navier–Stokes system. Recall that the BiCGSTAB algorithm is a transpose-free version of the Biconjugate gradient method (BCG) which significantly smooths the convergence rate of the BCG method. A big advantage of BiCGSTAB over GMRES is the fact that the algorithm involves only a three-terms recurrence. Hence, the storage requirement is much less than GMRES and the cost per step is reduced, although the BiCGSTAB method is slightly less robust than GMRES.

Several choices are possible for the preconditioning matrices M_A and M, ranging from the expensive choices $M_A = A$ and $M = M_p$ to the cheapest $M_A = \mathrm{diag}(A)$ and $M = \mathrm{diag}(M_p)$ where $\mathrm{diag}(E)$ stands for the diagonal part of the matrix E. Numerical experiments will be given in the last section.

4. ITERATIVE METHODS BASED ON THE HIERARCHICAL DECOMPOSITION

The hierarchical decomposition approach takes full advantage of the hierarchical basis for the velocity field. Let us recall that the velocity field is decomposed into a linear part u_l and a quadratic correction u_q. The Stokes problem is written accordingly in the following form:

$$\begin{bmatrix} A_{ll} & B_{pl}^t & A_{lq} \\ B_{pl} & 0 & B_{pq} \\ A_{lq}^t & B_{pq}^t & A_{qq} \end{bmatrix} \begin{bmatrix} u_l \\ p \\ u_q \end{bmatrix} = \begin{bmatrix} f_l \\ g \\ f_q \end{bmatrix} \tag{10}$$

which can also be written as

$$\begin{bmatrix} \tilde{A} & \tilde{B}^t \\ \tilde{B} & A_{qq} \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} = \begin{bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \end{bmatrix} \tag{11}$$

where

$$\tilde{A} = \begin{bmatrix} A_{ll} & B_{pl}^t \\ B_{pl} & 0 \end{bmatrix}, \quad \tilde{B} = [A_{lq}^t \quad B_{pq}^t], \quad \tilde{u}_1 = \begin{bmatrix} u_l \\ p \end{bmatrix}, \quad \tilde{u}_2 = [u_q], \quad \tilde{f}_1 = \begin{bmatrix} f_l \\ g \end{bmatrix}, \quad \tilde{f}_2 = [\, f_q]$$

System (11) is similar to the original Stokes system (3), although the variables are now different. The matrix $\tilde{A}$ corresponds to an unstable $P_1 - P_1$ discretization of the Stokes problem. The matrix A_{qq} is symmetric, positive definite and very well conditioned.

It is thus possible to solve this problem using similar techniques as those developed in the previous section. We first consider global preconditioned solvers and then we propose a Schur complement method.

4.1. Global block preconditioned iterative method

We start once again from formulation (10) or (11) but this time we consider this system as a whole and try to solve it through preconditioning. The preconditioner takes the form of a block diagonal matrix

$$\mathscr{M} = \begin{bmatrix} M_1 & 0 \\ 0 & A_{qq} \end{bmatrix} \tag{12}$$

The upper left matrix M_1 is nothing but system (4) associated with the Mini element. Consequently, preconditioning is obtained in two steps. The inversion of the block M_1 is done by the PCR method presented in the Section 3.2. A preconditioned conjugate gradient algorithm is applied to the matrix A_{qq}. This type of preconditioner is, however, quite expensive since two iteratives solvers are called each time preconditioning is applied.

A less expensive preconditioner (although less efficient) can be obtained by using a preconditioner of the form

$$\mathscr{M} = \begin{bmatrix} \tilde{M}_1 & 0 \\ 0 & \tilde{A}_{qq} \end{bmatrix} \tag{13}$$

where matrices $\tilde{M}_1$ and $\tilde{A}_{qq}$ have similar spectral properties in comparison with M_1 and A_{qq}. These matrices can for example be the associated SSOR matrices. This way, preconditioning does not require any nested iterations.

Finally, an intermediate preconditioner can take the form

$$\mathscr{M} = \begin{bmatrix} M_1 & 0 \\ 0 & \tilde{A}_{qq} \end{bmatrix} \tag{14}$$

at the cost of an iterative method (PCR) for the first block.

4.2. Schur complement approach

Here again, system (11) is splitted into two parts. First, we write

$$\tilde{A}\tilde{u}_1 + \tilde{B}^t\tilde{u}_2 = \tilde{f}_1$$

while the second equation of system (11) gives

$$\tilde{u}_2 = A_{qq}^{-1}(\tilde{f}_2 - \tilde{B}\tilde{u}_1)$$

Replacing, we easily get

$$(\tilde{A} - \tilde{B}^t A_{qq}^{-1}\tilde{B})\tilde{u}_1 = \tilde{f}_1 - \tilde{B}^t A_{qq}^{-1}\tilde{f}_2 \tag{15}$$

The above system is similar to Equation (6) obtained with Uzawa's method. As is easily seen, this system has the same dimensions as a linear ($P_1 - P_1$ or Mini) discretization. The size of the global system is thus greatly reduced. Iterative methods such as GMRES or BiCGSTAB can then be applied to this system.

Preconditioning is again an important issue. There are many possibilities and we will focus on two of them. The first preconditioner comes easily to mind. Since our system is of the same size as the one obtained using a Mini discretization, the (Mini) matrix M_1 can certainly be a preconditioner. However, there is a cost to this preconditioner since each time the preconditioner is applied, the PCR algorithm is used. As we shall see, this is an effective preconditioner. Finally, as in the previous section, a less expensive variant of the matrix M_1 can be used such as the associated SSOR matrix.

5. NUMERICAL RESULTS

In this section, we will discuss the performance of the iterative solvers described in the previous sections. Since we are mainly interested in large-scale industrial problems, we will restrict our attention to three-dimensional flow problems. All the 3D results presented, unless explicitly stated, have been obtained with the $P_2 - P_1$, 10 nodes tetrahedra with continuous pressure, i.e. the Taylor–Hood mixed finite-element. A hierarchical basis decomposition of the P_2 interpolation subspace has been adopted throughout the paper. Hence, the quadratic part of the velocity refers to the correction part of the quadratic finite-element subspace. The stopping convergence criterion is based on the relative l_2-norm of the preconditioned residual

$$\frac{\|r_k\|}{\|r_0\|} < \varepsilon = 10^{-6}$$

Most of the computations were performed on Pentium-based computers except for an SUN Enterprise 3000 workstation. The object-oriented code MEF++developed at the *Groupe Interdisciplinaire de Recherche en Éléments Finis* (GIREF) from University Laval, was used for the finite-element part. All the numerical algebra used the Portable, Extensible Toolkit for Scientific Computation (PETSc) developed at Argonne National Laboratory [17].

5.1. Tests problems

Various test problems are necessary to assess the accuracy and robustness of the algorithms presented in the previous section. Some iteratives methods (and their corresponding precon-

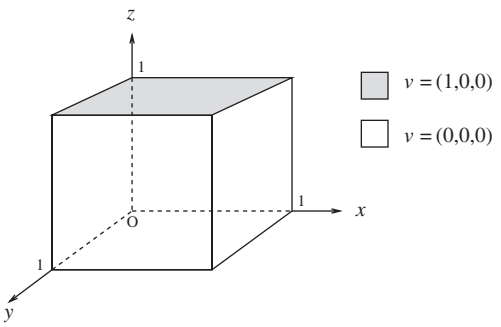

Figure 3. The 3D lid-cavity problem and the boundary conditions.

Table I. Features of the meshes for the 3D lid-driven problem.

Mesh	Number of elements	Number of pressure dof	Number of velocity dof (linear)	Number of velocity dof (quadratic)	Total number of dof
CAVITY1	48	27	81	294	402
CAVITY2	384	125	375	1812	2312
CAVITY3	3072	729	2187	12552	15468
CAVITY4	24576	4913	14739	93072	112724

ditioner) perform well on structured meshes but lose convergence on unstructured meshes. It is thus important to verify their performance on a wide range of problems.

3D lid-driven cavity: This is a classical test problem and a direct extension of the two-dimensional case. The geometry of the 3D lid-cavity problem and the boundary conditions are shown in Figure 3. No-slip boundary conditions are imposed on all sides but one, where a unit velocity field (tangent to the surface) drives the flow.

Since we want to study the performance of the various iterative solvers with respect to mesh refinement, we used four different meshes obtained by subdividing each edge by a factor of two from the previous mesh. Table I gives the different features of the four meshes in terms of the number of elements, nodes, vertices from which we deduce the number of velocity unknowns (linear and quadratic parts), the number of pressure unknowns, and the total number of degrees of freedom of the global system.

From this table, the cost of second-order element can be clearly seen in the last column. For the large meshes considered here, it appears that the total number of degrees of freedom for the Taylor–Hood discretization is approximatively 6 times the number of degrees of freedom with the Mini discretization.

Rectangular pipe flow: The second problem is the 3D rectangular pipe flow or the duct flow. This is a common geometry in polymer flow, for instance in injection moulding process. Contrary to the familiar cylindrical pipe flow, there is no simple closed form for the velocity profile even in the Newtonian case. The geometry and the boundary conditions are described

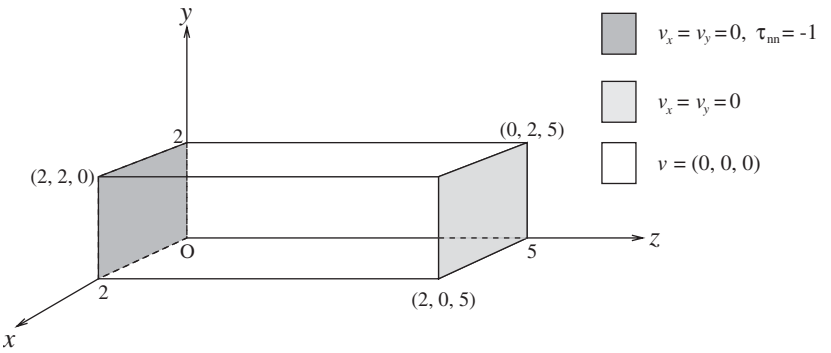

Figure 4. The rectangular pipe flow problem and the boundary conditions: $v = (v_x, v_y, v_z)$ and $\tau_{nn} = n \cdot \tau \cdot n$ where τ is the total stress tensor.

Table II. Features of the meshes for the rectangular pipe problem.

Mesh	Number of elements	Number of pressure dof	Number of velocity dof (linear)	Number of velocity dof (quadratic)	Total number of dof
PIPE1	120	54	162	663	879
PIPE2	960	275	825	4278	5378
PIPE3	7680	1701	5103	30444	37248
PIPE4	61440	11849	35547	229080	276476

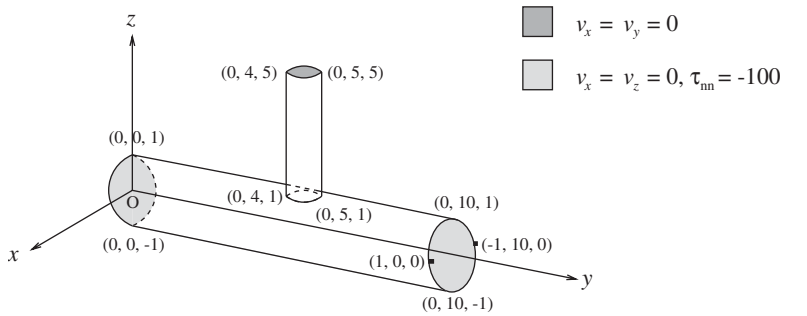

Figure 5. The T-geometry flow problem and the boundary conditions: $v = (v_x, v_y, v_z)$ and $\tau_{nn} = n \cdot \tau \cdot n$ where τ is the total stress tensor.

in Figure 4. A pressure gradient is imposed between the inlet and the outlet section of the boundary. Also, no transverse flow is imposed at the inlet and the outlet section. Once again, we have used four meshes obtained in a similar way as the previous 3D lid-driven geometry. Table II gives the different features of the four meshes in terms of the degrees of freedom with respect to the pressure, the linear and the quadratic correction parts of the velocity.

T-geometry: This problem concerns the flow at the junction of two cylinders, typically encountered in the design of die profiles in polymer processing. This is a first example of a problem where an unstructured mesh is needed. It is still an academic problem but as we shall

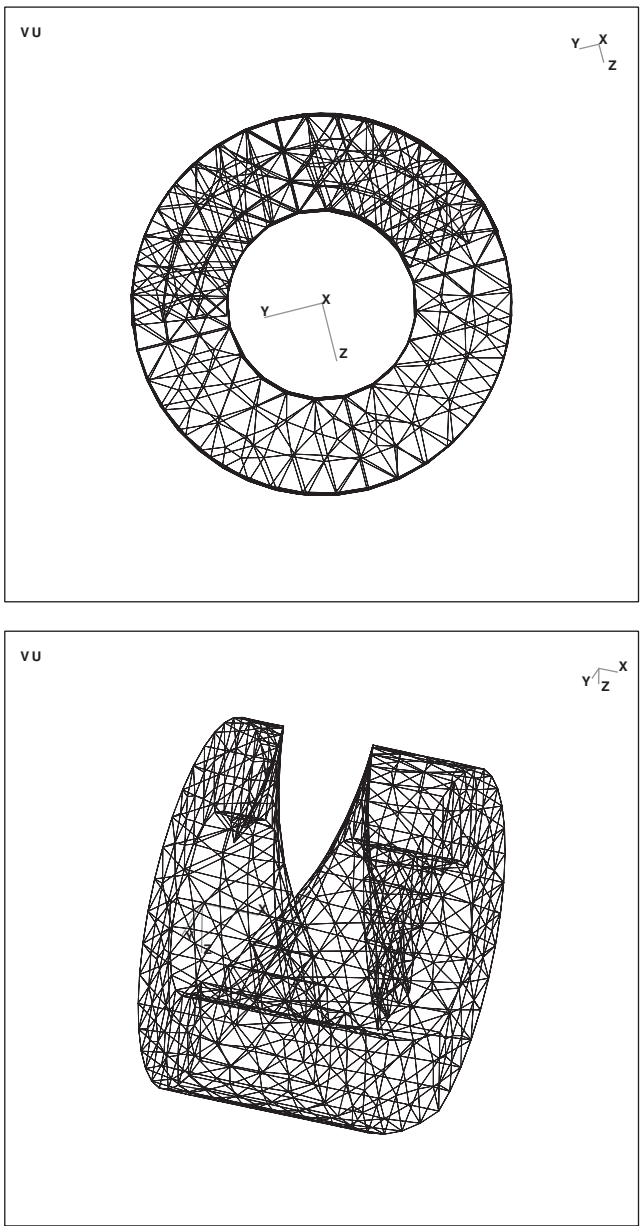

Figure 6. The geometry and the mesh of the single screw extruder.

see, some previously successful iterative methods failed on this problem. The geometry and boundary conditions are given in Figure 5. Basically, the fluid is entering at both extremities of the bottom (larger) cylinder and it gets out by the top of the transverse cylinder. Only one mesh has been used for this problem, see Table III for details.

Single screw extruder: Our ultimate goal is to develop a numerical strategy for the solution of non-Newtonian flow problems encountered in polymer processing. A typical example is the flow in a single flow extruder. Figure 6 shows the geometry from two different angles. As can be easily seen, the geometry is complex. A pressure gradient is imposed between the inlet and outlet section. No-slip conditions are imposed on the screw. Table III gives the features of the mesh used for the single screw extruder problem.

5.2. *Performance of the solvers: the* (u, p) *approach*

5.2.1. Uzawa's method. We first solve the test problems by the PCG (Uzawa) algorithm as described in Section 3.1. Various strategies for solving the inner problem (velocity) A^{-1} are compared. First, we solve as exactly as possible the inner problem by the PCG method forcing the convergence to a 10^{-8} reduction of l_2-norm of the (preconditioned) residual. Table IV gives the iteration counts and the ∞-norm of the residual for all the meshes. Table IV shows the nice property of the Uzawa method: the number of iterations of the PCG algorithm is independent of the mesh size. Good accuracy is also obtained with this method as indicated in the second column even if the stopping criterion is based on the preconditioned residual. Nevertheless, this method is an example of nested iterations. This strategy can be costly since the inner problem must be solved very accurately. Secondly, we limit the number of iterations for the inner (velocity) PCG solver say at 50. In this case, numerical evidence indicates (not shown here) that the overall accuracy is severely affected by the level of precision of the inner problem.

5.2.2. PCR method. Numerical experiments were performed for the preconditioned conjugate residual method presented in Section 3.2. The following block preconditioner was used

$$\begin{bmatrix} M_A & 0 \\ 0 & M \end{bmatrix}$$

where M represents the diagonal part of the pressure mass matrix M_p, i.e. $M = \mathrm{diag}(M_p)$. From the work of Wathen and Silvester [10, 11], it is known that the behaviour of this iterative solver is essentially independent of the choice for M if one choose a spectrally equivalent matrix to the original pressure mass matrix. The matrix M_A should be an approximation of the matrix A coming from the velocity part. We have tested two choices for the matrix M_A. The first choice makes use of the popular SSOR relaxation scheme on the matrix A. The second choice is the less expensive variant corresponding to the diagonal part of the matrix A (Jacobi preconditioner). Iteration counts and the ∞-norm of the residual are shown at Table V for the 3D lid-driven cavity problem and Table VI for the pipe flow. In all cases, the overall accuracy is good. For the Jacobi preconditioner, the iteration counts behave almost linearly with the mesh size, $O(h^{-1})$. The behaviour of the SSOR preconditioner is better according to the number of flops as given by the PETSc package, [17].

5.2.3. Arrow–Hurwicz method. Finally, we have tested our version of the Arrow–Hurwicz method as presented in Section 3.3. The global system is solved by the BiCGSTAB algorithm

Table III. Features of the meshes for the T-geometry and the single screw problem.

Mesh	Number of elements	Number of pressure dof	Number of velocity dof (linear)	Number of velocity dof (quadratic)	Total number of dof
T-GEOM	16726	3847	11541	67731	83119
SCREW	3769	1008	3024	16257	20289

Table IV. PCG iterations for the Uzawa method: 3D lid-driven cavity problem.

	Uzawa	
Mesh	Number of iterations	l_∞-Norm of the residual
CAVITY1	20	1.2×10^{-6}
CAVITY2	27	4.6×10^{-7}
CAVITY3	28	1.7×10^{-7}
CAVITY4	29	1.7×10^{-6}

Table V. PCR iterations: 3D lid-driven cavity problem.

	PCR method					
	SSOR preconditioner			Jacobi preconditioner		
Mesh	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$
CAVITY1	49	3.0×10^{-7}	0.053	60	3.9×10^{-7}	0.038
CAVITY2	136	2.9×10^{-7}	0.994	236	2.6×10^{-7}	1.016
CAVITY3	211	6.1×10^{-7}	11.35	374	2.6×10^{-7}	11.77
CAVITY4	252	2.0×10^{-7}	104.0	679	1.3×10^{-7}	163.2

with the following block triangular preconditioner decomposition

$$\begin{bmatrix} M_A & 0 \\ B & -M \end{bmatrix}$$

Again, we have chosen $M = \mathrm{diag}(M_p)$ where M_p is the pressure mass matrix. The matrix B is the discrete divergence matrix. Similar to the PCR approach, SSOR and Jacobi has been chosen for the M_A preconditioner. Tables VII and VIII give the iteration counts and the ∞-norm of the residual for the BICGSTAB algorithm in the case of the 3D lid-driven cavity problem and the pipe flow, respectively. According the values of the ∞-norm, the overall accuracy is still good with this method.

Table VI. PCR iterations: rectangular pipe flow problem.

PCR method						
	SSOR preconditioner			Jacobi preconditioner		
Mesh	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$
PIPE1	75	1.7×10^{-7}	0.189	109	2.7×10^{-7}	0.163
PIPE2	134	8.7×10^{-8}	2.368	244	1.4×10^{-7}	2.533
PIPE3	215	5.7×10^{-8}	28.44	441	4.0×10^{-8}	34.05
PIPE4	314	2.6×10^{-8}	321.2	853	1.0×10^{-8}	507.8

Table VII. BiCGSTAB iterations for the Arrow–Hurwicz method: 3D lid-driven cavity problem.

Arrow–Hurwicz method						
	SSOR preconditioner			Jacobi preconditioner		
Mesh	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$
CAVITY1	28	2.3×10^{-8}	0.059	39	5.8×10^{-8}	0.050
CAVITY2	31	1.5×10^{-7}	0.453	71	1.6×10^{-7}	0.607
CAVITY3	43	3.4×10^{-8}	4.636	108	2.2×10^{-7}	6.781
CAVITY4	70	3.6×10^{-8}	57.88	266	1.7×10^{-7}	127.9

Table VIII. BiCGSTAB iterations for the Arrow–Hurwicz method: rectangular pipe flow problem.

Arrow–Hurwicz method						
	SSOR preconditioner			Jacobi preconditioner		
Mesh	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$	Number of iterations	l_∞-Norm of the residual	Flops $\times 10^8$
PIPE1	23	2.8×10^{-8}	0.115	39	9.8×10^{-8}	0.115
PIPE2	27	1.2×10^{-7}	0.957	65	3.0×10^{-7}	1.342
PIPE3	43	0.5×10^{-8}	11.41	159	8.2×10^{-8}	24.52
PIPE4	76	1.9×10^{-8}	155.8	244	5.6×10^{-8}	290.8

From the tables, we observe a similar trend compared to the PCR approach, namely the linear behaviour with the mesh size for the Jacobi preconditioner although this effect is less pronounced. The real surprise comes from the SSOR preconditioner showing a very good behaviour with the mesh size. The numbers of flops indicate that the performance of the

SSOR preconditioner is better even if it is more costly than Jacobi. Although it is not constant with respect to the mesh size, its growth is far from the linear case. This combination of the BiCGSTAB algorithm and the above preconditioner with the SSOR matrix appear to be a very effective way to solve large flow problem. This method is roughly twice as fast as the PCR method on both test problems. Of course, instead of the SSOR, more sophisticated preconditioner can be use such as ILUT family or multigrid methods. In this paper, we have restricted our analysis to the most basic preconditioners.

5.3. Performance of the solvers: the hierarchical basis approach

In this subsection, we shall analyse the performance of the hierarchical basis solvers proposed in Section 4. Recall that these solvers make full use of the hierarchical decomposition of the continuous quadratic finite-element discretization of the Stokes problem. This introduces a non-standard block partition of the unknowns: the first block is composed of the linear part of the velocity and the pressure unknowns while the second block is associated to the quadratic correction unknowns of the velocity.

5.3.1. Global preconditioned method. At first, a global approach is adopted for solving the linear system (10). Let us recall from Section 4.1 that the preconditioner based on matrix (12) requires full resolution of a linear system with a Mini matrix M_1 and the solution of another linear system with matrix A_{qq}. In all computations, the global system (10) is solved by the restarted version of GMRES with the dimension of the Krylov subspace fixed to 30. We first tested the cubic cavity flow problem. As expected from the good conditioning of the matrix A_{qq}, we found that the number of iterations was independent on the mesh size (see Table IX). This is a nice feature of the solver similar to the Uzawa algorithm discussed previously. Unfortunately, the convergence is rather slow compared to the Uzawa's method. Table IX also shows the results with GMRES solver preconditioned by matrix (13). Among the methods based on a hierarchical decomposition and in terms of computational time, this method was the cheapest. On the cavity flow problem, the number of iterations was no longer independent on the mesh size. There is, however, some drawback to this approach. Apparently, the preconditioning matrix (13) is sufficient on structured meshes such as those we used on the cavity flow problem. We, however, found that on unstructured meshes, convergence was sometimes difficult and in some cases impossible to obtain. In particular, we were not able to get a fully converged solution for the intersection of two cylinders problem.

Finally, we proposed in (14) to use the Mini solver as discussed in Section 3 and the diagonal part of A_{qq} matrix as the preconditioning strategy for the global system (10). The results are summarized in the last column of the Table IX. We still found that the number of iterations was independent on the mesh size making it an efficient way to solve iteratively the Stokes problem. But it remains costly in terms of iteration counts. As we shall see later, a possible explanation of this failure relies on the pressure variable which is not well calculated with the Mini element.

5.3.2. Schur complement method. A Schur complement approach (see Section 4.2) is also possible for solving the global linear system (10). In fact, from the block partition system (10), we can eliminate the quadratic correction part of the velocity. The resulting Schur complement system has the same dimension as the $P_1 - P_1$ mixed finite-element discretization

Table IX. GMRES iterations for the global preconditioned method.

Hierarchical basis approach: global system						
	Preconditioned by 12		Preconditioned by 13		Preconditioned by 14	
Mesh	Number of iterations	l_∞-Norm of the residual	Number of iterations	l_∞-Norm of the residual	Number of iterations	l_∞-Norm of the residual
CAVITY1	85	9.2×10^{-5}	87	3.4×10^{-4}	105	4.8×10^{-5}
CAVITY2	117	6.5×10^{-5}	222	1.6×10^{-4}	117	6.8×10^{-5}
CAVITY3	113	1.2×10^{-4}	342	3.7×10^{-4}	115	1.7×10^{-4}
CAVITY4	115	9.4×10^{-5}	501	1.4×10^{-4}	119	7.5×10^{-5}
CYLINDERS	119	7.5×10^{-3}	Diverge		162	5.5×10^{-4}

Table X. GMRES iterations for the Schur decomposition method.

Hierarchical basis approach: Schur decomposition (GMRES solver)				
	Mini preconditioner		SSOR preconditioner	
Mesh	Number of iterations	l_∞-Norm of the residual	Number of iterations	l_∞-Norm of the residual
CAVITY1	20	3.1×10^{-5}	23	3.4×10^{-4}
CAVITY2	42	4.8×10^{-5}	73	1.6×10^{-4}
CAVITY3	41	8.5×10^{-5}	104	3.7×10^{-4}
CAVITY4	48	6.7×10^{-5}	156	1.4×10^{-4}
CYLINDERS	47	1.0×10^{-3}	138	1.4×10^{-2}

of the Stokes problem. Hence, the Mini solver can be used for preconditioning. As before, we examine the performance of the solvers on the cavity problem.

At first, the restarted GMRES(m) method ($m = 30$) is applied to system (15) and preconditioning is obtained by solving the Mini system by the (PCR) solver introduced in Section 3. The iteration counts are given at Table X and are similar to the ones produced by the Uzawa's method although slightly higher. For the cavity problem, the number of iterations was found to be independent of the mesh size.

Next, a less costly approach makes use of the same GMRES solver preconditioned by SSOR computed from the Mini matrix. This is a relatively efficient compromise since we obtain a convergent algorithm in all test cases (see Table X).

Finally, we slightly modify the preceding method by replacing the GMRES solver by the BiCGSTAB solver. The reason being that BiCGSTAB requires less memory space than GMRES. In this case, the number of iterations was reduced to 24, 45, 61 and 103 for the cavity problem (see Table XI). From these results, we can conclude that the last method is an effective iterative solver for the Stokes problem. Schur complement methods make use of nested iterations. In this case, the inner iteration are not costly since the inversion of

Table XI. BiCGSTAB iterations for the Schur decomposition method.

Hierarchical basis approach: Schur decomposition (BiCGSTAB solver)		
	SSOR preconditioner	
Mesh	Number of iterations	l_∞-Norm of the residual
CAVITY1	24	1.1×10^{-9}
CAVITY2	45	5.6×10^{-5}
CAVITY3	61	1.9×10^{-4}
CAVITY4	103	5.1×10^{-5}
CYLINDERS	76	1.6×10^{-3}

the A_{qq} matrix can be done efficiently by a PCG algorithm thanks to the good conditioning of the matrix A_{qq}. Nevertheless, the above solver based on the Schur complement method is less competitive than the Arrow–Hurwicz method of Section 3. As previously stated, the approximation of the pressure variable by the Mini element is possibly the cause of the failure.

5.4. Quadratic versus linear discretization

In this section, we shall compare the solutions given by the Taylor–Hood (quadratic) and the Mini (linear) discretizations. For this purpose, we extend to the 3D case a body force problem originally proposed by Pierre [4]. Let the velocity, the pressure and the body force given by

$$\mathbf{u} = (x^2, -xy, -xz)$$

$$p = x^2 + y^2 + z^2 + p_0$$

$$\mathbf{f} = (2(x-1), 2y, 2z)$$

defined on the unit cube $\Omega = [0,1]^3$. By imposing the appropriated boundary conditions on $\mathbf{u}$, $\mathbf{u}_{|\partial\Omega} = \mathbf{u}$, the pair $(\mathbf{u}, p)$ satisfies immediately the Stokes equations. Since the pressure is quadratic, discretization errors will be observed for both types of discretization.

The interest of this problem lies in the fact that the Mini solution exhibits parasitic pressures (see Reference [4] for the 2D case). Indeed, Figure 7 compares the pressure obtained by the quadratic Taylor–Hood element and the first-order Mini element. The velocity fields are also shown. The velocity obtained by the Mini element is much better than the computed pressure. Similar calculations for the pressure variable are reproduced in Figure 8 for the T-geometry. Now the computed pressure with the Mini element is better but still has some oscillations.

These examples show clearly that the Mini solution may not be reliable especially for the pressure variable. For instance, the calculation of the pressure drop using the Mini solution may be erroneous. Of course, the oscillations in the pressure can be attenuated by mesh refinement or by a correct choice of the stabilization parameters. Unfortunately, there is no general rule for choosing these coefficients.

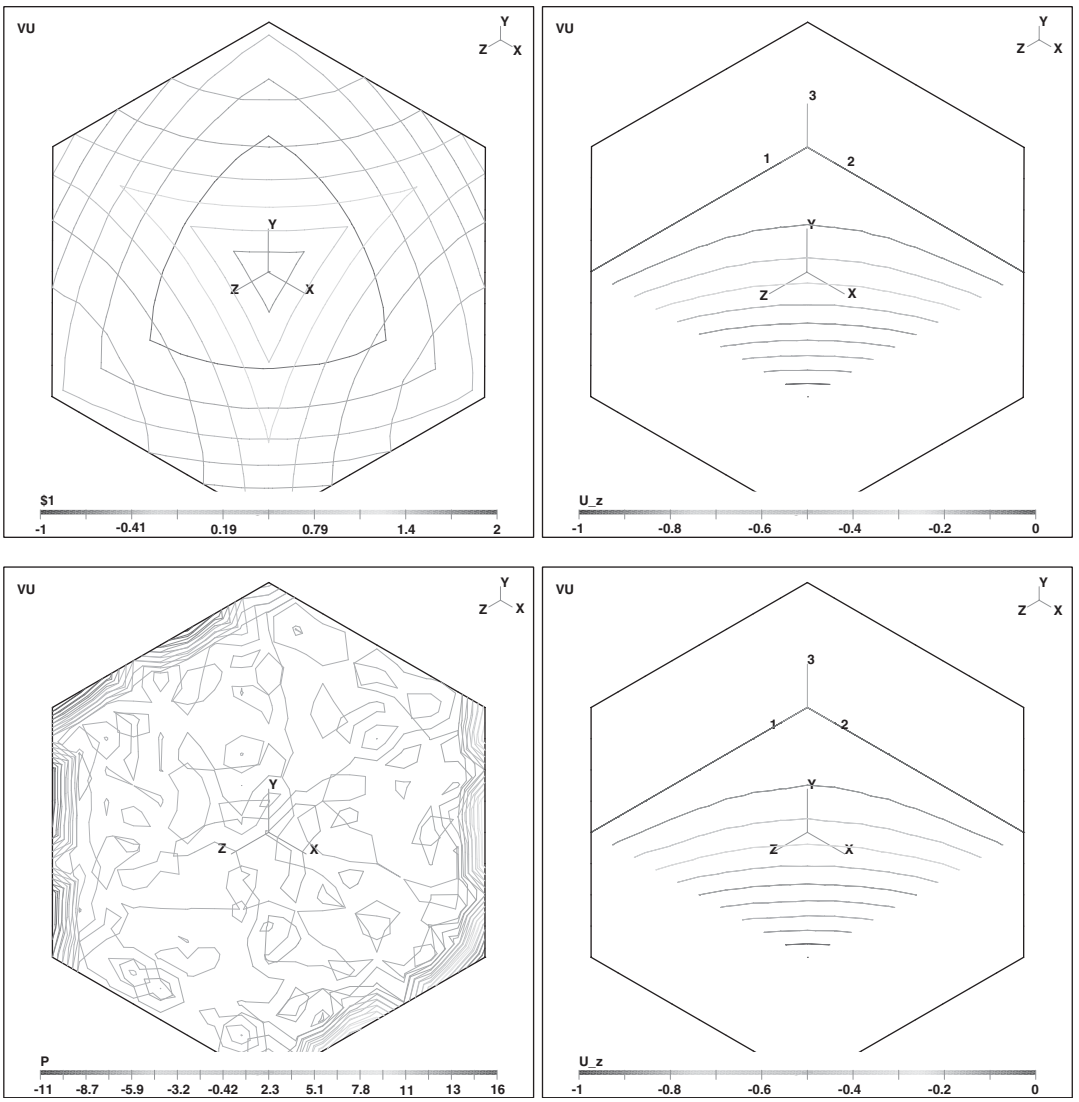

Figure 7. The body-force problem: (Top) level curves of the pressure at the boundary for the Taylor–Hood element; and level curves of the z-component of the velocity at $y = 0.5$ for the Taylor–Hood element; (Bottom) level curves of the pressure at the boundary for the Mini element; and level curves of the z-component of the velocity at $y = 0.5$ for the Mini element.

Our final remark is about the computational cost of quadratic simulations. We wish to compare the Arrow–Hurwicz solver based on the Taylor–Hood ($P_2 - P_1$) discretization with the PCR solver using the Mini element ($P_1 - P_1$). Of course, the number of degrees of freedom is much higher for the $P_2 - P_1$ element, about six times the total number of degrees of freedom for the linear case but the latter is only first-order accurate. Hence, we should

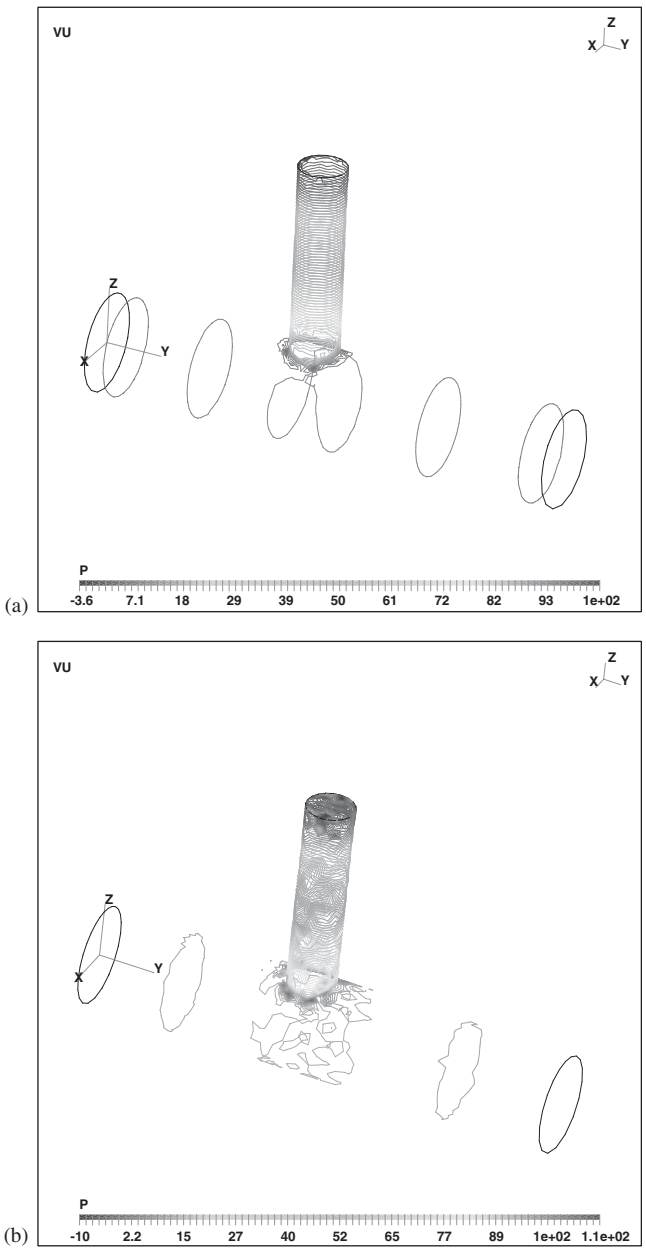

Figure 8. The T-geometry problem: (a) level curves of the pressure at the boundary for the Taylor–Hood element; (b) level curves of the pressure at the boundary for the Mini element.

compare results with approximatively the same numbers of degrees of freedom. For example, if we subdivide each edge by a factor of two, the last mesh for the cavity problem, we obtain a new mesh called CAVITY5 having 143, 748 degrees of freedom in the linear case. This is

similar to the quadratic discretization having 112,724 degrees of freedom on the CAVITY4 mesh. The Mini solver on CAVITY5 required approximatively 2.4×10^{10} flops while the Arrow–Hurwicz with the SSOR preconditioner has only 0.58×10^{10} flops or 1.27×10^{10} with the Jacobi preconditioner.

5.5. *Non-linear case*

Applications in polymer processing imposes to consider the non-Newtonian case where the viscosity is given by the Carreau model:

$$\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0(1 + \lambda^2|\dot{\gamma}(\mathbf{u})|^2)^{(n-1)/2}$$

Although the Carreau model is more realistic, the power-law model is also often used:

$$\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0(|\dot{\gamma}(\mathbf{u})|)^{n-1}$$

From a numerical standpoint, the power-law model is much more difficult to deal with since the viscosity goes to ∞ when $|\dot{\gamma}(\mathbf{u})|$ tends toward 0. The Newton method described in Section 1 was applied and the resulting linear systems were solved by the Uzawa algorithm (see Section 3.1) and the Arrow–Hurwicz method of Section 3.3. Both methods gave similar results although the Arrow–Hurwicz was a lot faster. Starting from a Newtonian solution ($n = 1$), the index n was decreased by steps of 0.1 until $n = 0.3$. The Newton method converges in less than five iterations for each value of n and the linear systems were always fully converged without any problem.

The effect of the parameter n is important. As n decreases the problem becomes more and more difficult to solve. Moreover, in regions of low shearing, the viscosity is very large and the velocity profile takes the form of a plug flow. This is exactly what can be see in Figure 9 for the rectangular pipe flow where isovalues of the velocity profiles on a plane at mid-height of the channel ($y = 1$) are shown. The difference between the Newtonian case and the viscoplastic case ($n = 0.3$) can be easily seen. In the Newtonian case, the velocity profile is more or less parabolic while in the viscoplastic case, an almost completely flat profile is observed except close to the boundaries.

Finally, we consider the flow in a section of a single screw extruder. This problem was chosen because the associated geometry and mesh are complex providing a good numerical test for iteratives solvers. A pressure gradient is imposed between the entrance and exit sections of the domain. Everywhere else, we impose a no-slip boundary condition. Figure 10 shows the velocity vectors for the Newtonian case $n = 1$ and the viscoplastic case $n = 0.3$. The main difference lies in the maximum modulus of the velocities which decreases with n.

6. CONCLUSION

We have developed different solution strategies for the numerical solution of three-dimensional problems arising from quadratic discretization of the Stokes problem. Uzawa and Schur complement methods are very robust but need an accurate solver for A^{-1}. In general, this is costly. Global approaches are much less time consuming. An iterative solver based on the hierarchical decomposition is an attractive idea but seems to be less efficient than the usual velocity–pressure counterpart. The failure relies on the approximation of the pressure variable.

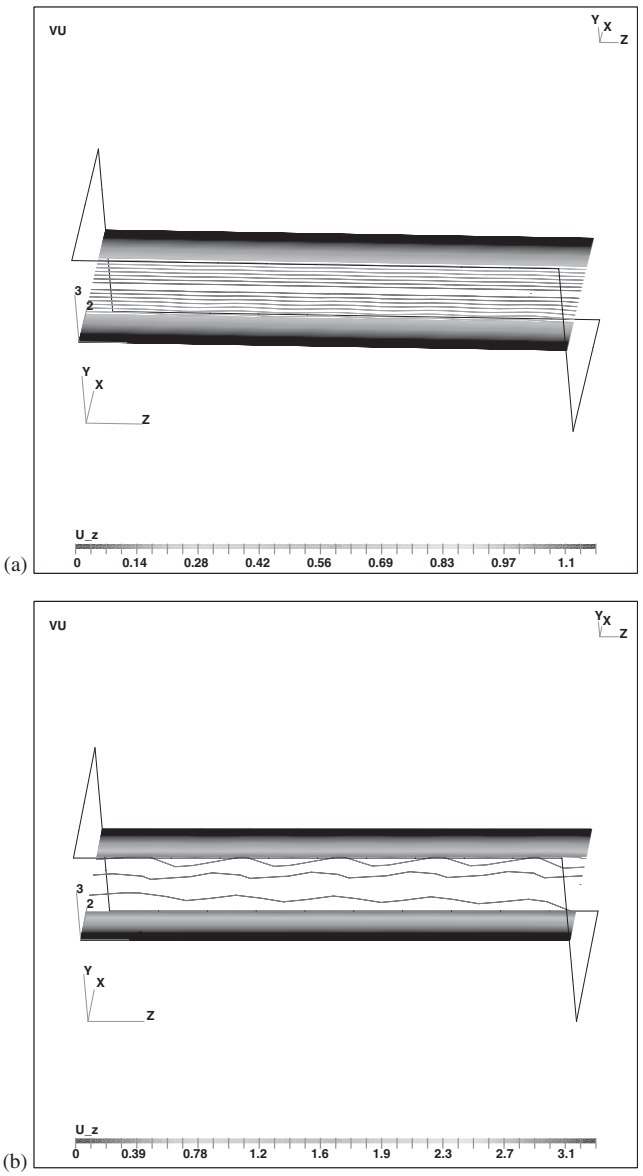

Figure 9. Rectangular pipe flow: velocity profile on plane $y = 1$ for (a) $n = 0$ and (b) $n = 0.3$.

In the near future, a more effective stabilized $P_1 - P_1$ solver will be used as the preconditioner. Among all the solvers considered in this paper, the Arrow–Hurwicz method accelerated by the BiCGSTAB algorithm and preconditioned by SSOR is the best choice. Nevertheless, this solver is not optimal with respect to the mesh size and improvement is necessary. The use of multigrid as preconditioner for the velocity part is certainly an issue to pursue. Moreover, the method does not involve any adjustable coefficient. Finally, the choice of quadratic

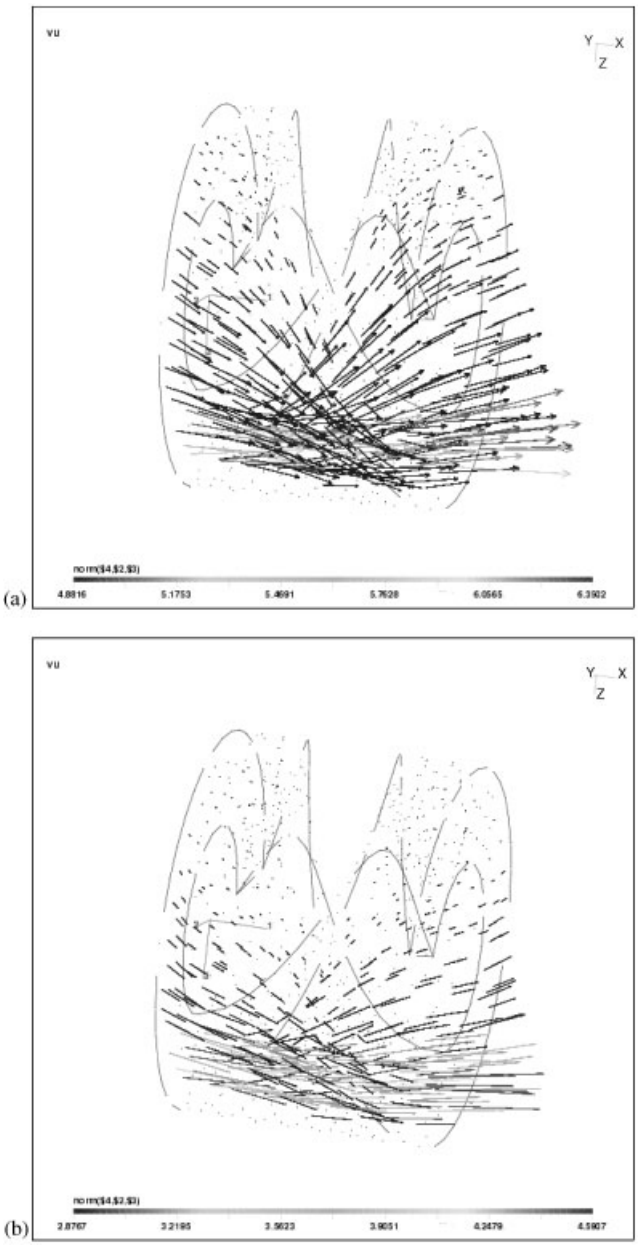

Figure 10. Single screw extruder: velocity profiles for (a) $n = 1$ and (b) $n = 0.3$.

discretization deserves some comments. Of course, quadratic elements are more expensive than first-order elements. But quadratic elements are much more accurate especially if one is interested in the pressure variable. For applications in polymer processing, the precise value of the excess pressure drop is often needed. Finally, if one compares the computational cost

of quadratic and linear simulations with two meshes having the same number of degrees of freedom, it is not so clear that the linear approximation will be cheaper, but it will certainly be less accurate.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support received from the National Sciences and Engineering Research Council (NSERC) of Canada and the Programme FCAR of the Province of Québec.

REFERENCES

1. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
2. Arnold DN, Brezzi F, Fortin M. A stable finite element for Stokes equations. *Calcolo* 1984; **21**:337–344.
3. Turek S. *Efficient Solvers for Incompressible Flow Problems*. Springer: Berlin, 1999.
4. Pierre R. Simple C^0 approximations for the computation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1988; **68**:205–227.
5. Coupez T. Stable-stabilized finite element for 3D forming calculation. *Internal Report*, *CEMEF*, 1996.
6. Guénette R, Fortin M. A new mixed finite element method for computing viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics* 1995; **60**:27–52.
7. Cahouet J, Chabard JP. Some fast 3-d finite element solvers for generalized Stokes problem. *Report EDF HE/41/87.03*, 1987.
8. Rusten T, Winther R. A preconditioned iterative method for saddle point problems. *SIAM Journal of Matrix Analysis and Applications* 1992; **31**:887–904.
9. Bank RE, Welfert BD, Yserentant H. A class of iterative methods for solving saddle point problems. *Numerische Mathematik* 1990; **56**:645–666.
10. Wathen AJ, Silvester DJ. Fast iterative solution of stabilised Stokes systems. Part I: using simple diagonal preconditioners. *SIAM Journal of Numerical Analysis* 1993; **30**:630–649.
11. Wathen AJ, Silvester DJ. Fast iterative solution of stabilised Stokes systems. Part II: using general block preconditioners. *SIAM Journal of Numerical Analysis* 1994; **31**:1352–1367.
12. Verfürth R. A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem. *IMA, Journal of Numerical Analysis* 1994; **4**:441–455.
13. Fortin M, Glowinski R. *Résolution Numérique de Problèmes aux Limites par des Méthodes de Lagrangien Augmenté*. Dunod: Paris, 1983.
14. Coupez T, Marie S. From a direct solver to a parallel iterative solver in 3D forming simulation. *International Journal of Supercomputer and Applications* 1997; **11**:205.
15. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS: Boston, MA, 1996.
16. van der Vorst H. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear system. *SIAM Journal on Scientific and Statistical Computing* 1992; **12**:631–644.
17. Balay S, Gropp W, McInnes LC, Smith B. PETSc 2.0 Users manual. *http://www-fp.mcs.anl.gov/petsc/*.

