Received January 2005
Revised November 2005
Accepted November 2005

On iterative solvers for non-Newtonian flow equations

Oleg Iliev and Joachim Linn
Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany
Mathias Moog
MAGMA Giessereitechnologie GmbH, Aachen, Germany
Dariusz Niedziela
Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany, and
Vadimas Starikovicius
Vilnius Gediminas Technical University, Vilnius, Lithuania

Abstract

Purpose – This study proposes to develop and investigate different iterative solvers for non-Newtonian flow equations.

Design/methodology/approach – Existing approaches for the time discretization of the flow equation and for an iterative solution of the discrete systems are discussed. Ideas for further development of existing preconditioners are proposed, implemented and investigated numerically.

Findings – A two-level preconditioning, consisting of a transformation of the original system in the first step and subsequent preconditioning of the transformed system is suggested. The GMRES iterative method, which usually performs well when applied to academic problems, showed dissatisfactory performance for the type of industrial flow simulations investigated in this work. It was found that the BiCGStab method performed best in the tests presented here.

Research limitations/implications – The iterative solvers considered here were investigated only for a certain class of polymer flows. More detailed studies for other non-Newtonian flows should be carried out.

Originality/value – The work presented in this paper fills a gap related to the usage of efficient iterative methods for non-Newtonian flow simulations.

Keywords Iterative methods, Solutions, Newton method, Flow

Paper type Research paper

1. Introduction

Efficient numerical simulation of non-Newtonian flows is a non-trivial task. While the numerical methods for Newtonian flows are, in general, well studied, this is not the case for non-Newtonian fluids. Here we discuss the numerical solution of a class of flow problems for generalized Newtonian fluids. In this case the scalar viscosity is not a

This work is partially supported by HYKE: a Research Training Network (RTN) on "HYperbolic and Kinetic Equations: Asymptotics, Numerics, Analysis" financed by the European Union in the 5th Framework Programme "Improving the Human Potential" (IHP). Project Reference: Contract Number: HPRN-CT-2002-00282.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 16 No. 5, 2006
pp. 602-616

0961-5539
DOI 10.1108/09615530610669139

constant but depends on the temperature and on the effective strain rate (as well as on the pressure in certain cases). As a result, the momentum equations are strongly coupled through their viscous terms (recall that for Newtonian fluids the coupling of the momentum equations is only through the convective terms and the pressure). In the Newtonian case, projection methods for decoupling the momentum and the continuity equations are often used (for details and further references on fractional time step projection methods of Chorin type, or on SIMPLE-type algorithms, Gresho and Sani, 1998; Turek, 1999; Fletcher, 1991; Ferziger and Peric, 1999). Decoupling methods might not be efficient in the non-Newtonian case when the momentum equations are strongly coupled through the viscous terms. An alternative to the segregated (decoupled) solvers are the so-called coupled solvers (Turek, 1999; Deng *et al.*, 2001; Ferry, 2002; Silvester *et al.*, 2001; Axelsson and Neytcheva, 2003; Benzi, 2004). In this case the momentum and the (transformed) continuity equation are solved together. In the current paper we study the performance of certain iterative solvers for solving such coupled systems arising in the discretization of non-Newtonian flow equations.

The remainder of the paper is organized as follows. The next section presents the governing equations. The discretization is discussed in Section 3. Iterative methods and preconditioners for solving the coupled system of equations are explained in Section 4. Section 5 is devoted to numerical experiments and their analysis. In the last section we give a summary and discussion of our work and draw some conclusions.

2. Governing equations

If we consider unsteady non-isothermal Navier-Stokes equations describing weakly compressible flow of a liquid with variable viscosity, e.g. like the fluids described by the Cross WLF model, the equations for momentum conservation have the following form:

$$\frac{\partial(\rho u_i)}{\partial t} + \mathrm{div}(\rho u_i u) = -\frac{\partial p}{\partial x_i} - \frac{2}{3}\frac{\partial}{\partial x_i}(\mu \mathrm{div}(u)) + \mathrm{div}(2\mu\dot{\gamma}_i) \tag{1}$$

The u_i, $i = 1, 2, 3$ are the components of the velocity vector, ρ denotes the density, μ is the viscosity, and:

$$\dot{\gamma} = \frac{1}{2}(\nabla u + (\nabla u)^T)$$

is the shear rate tensor. Mass conservation is governed by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0, \tag{2}$$

and the conservation of energy is provided by the energy equation (written with respect to enthalpy h):

$$\frac{\partial(\rho h)}{\partial t} + (\vec{u}, \nabla T) = \mathrm{div}(\kappa \nabla T) + \mu\Phi_V + L\frac{\partial f_s}{\partial t}. \tag{3}$$

The last two terms account for dissipative heating and the latent heat released during phase transition, respectively.

The equations of state for the density and viscosity have to be added to the system of conservation equations. The density $\rho(p, T)$ depends on pressure and temperature. The viscosity $\mu(T, |\dot{\gamma}|, p)$ depends on temperature and the effective rate $|\dot{\gamma}|$ of the strain tensor. Accounting for a pressure dependence is also desirable in certain cases. Specific rheology and pvT models can be selected from the SIGMASOFT© material database (see www.sigmasoft.de for detailed information on this subject).

The unknowns which have to be determined from the above system of equations are temperature T, the velocity vector $\vec{u}$, density ρ, pressure p and viscosity μ.

The system describes, in particular, the flow and solidification of a polymer liquid. The simulation of polymer molding and solidification is of special interest for us, and the principle goal of our investigations is to accelerate the flow solver in SIGMASOFT©. In fact, a relatively small time step has to be used in simulating polymer solidification due to the existing free boundaries during mold filling and subsequent solidification of the polymer melt. The most CPU consuming part of the simulation is the calculation of the flow at each time step. Motivated by this, we consider below the flow equations decoupled from the heat equation. The latter is not discussed here. We note that the main non-isothermal effects to be handled here are the temperature variations occurring mainly in the region near the walls of the mold, which lead to a significant variation – possibly over several orders of magnitude – in the viscosity by means of its temperature dependence, and result in a strong coupling of the momentum equations through the viscous terms.

We denote the operators corresponding to the convective and viscous terms in the momentum equations by $C_{\vec{u}}$ and $D_{\vec{u}}$. Obviously these operators depend on the unknowns as $C_{\vec{u}} = C_{\vec{u}}(\vec{u}), D_{\vec{u}} = D_{\vec{u}}(\mu(T, |\dot{\gamma}|, p))$. Further we denote the operators corresponding to the gradient and divergence by G and B, respectively. Using this notation we are able to rewrite the Navier-Stokes equations in the following way:

$$\frac{\partial(\rho\vec{u})}{\partial t} + C_{\vec{u}}\vec{u} - D_{\vec{u}}\vec{u} + Gp = 0, \tag{4}$$

$$B\vec{u} = 0. \tag{5}$$

Note that weak compressibility does not play an essential role in the type of liquid polymer flows we consider here, therefore, we will not pay special attention to it. Although we solve the weakly compressible equations, we restrict our discussion to the incompressible case, which is sufficient for our purposes. For a more detailed study on methods for weakly compressible flows (in cases when compressibility is important) we refer to Core and Angot (2002), Churbanov (2003) and Ferziger and Peric (1999) and references therein.

3. Discretization

Finite volume method on a staggered grid is applied for discretizing in space the above system of equations (Hattel, 2005 or Fletcher, 1991). Particular form of this space discretization is not discussed here, instead we concentrate on discretization in time and later on, on iterative methods for solving the discretized system. Let us shortly discuss three different time discretization strategies: fractional time step *projection* methods, fractional time step *coupled* method, and implicit discretization.

First, consider fractional time step projection methods. These methods consist of solving the momentum equations in two sub-steps. Projection methods (e.g. Chorin type methods, Gresho and Sani, 1998 for a detailed discussion) treat pressure explicitly at the first step, which allows to decouple the momentum and the continuity equations (the last is used for obtaining an equation with respect to the pressure or the pressure correction, respectively). A variant of such a projection method looks as follows:

$$(\rho\vec{u})^{n+(1/2)} + \tau(C_{\vec{u}}\vec{u}^{n+(1/2)} - D_{\vec{u}}\vec{u}^{n+(1/2)}) = (\rho\vec{u})^{n} - \tau G p^{n},$$

$$(\rho\vec{u})^{n+1} - (\rho\vec{u})^{n+(1/2)} = -\tau[G p^{n+1} - G p^{n}], \quad B(\rho\vec{u})^{n+1} = 0.$$

We use the superscript n to denote the values at the old time level, and $n + 1$ to denote the values at the new one. The variable $\tau := t^{n+1} - t^{n}$ denotes the time step. The discrete operators are denoted by the same notation as introduced above for the continuous ones. The viscosity function in the diffusive terms and the velocity in the convective terms are both treated explicitly.

The sum of the first and the second equations above yields a discretization of the momentum equations. Applying the divergence operator (denoted by B in our case) to the second equation, and using the third (continuity) equation, we obtain:

$$\tau B G \delta p = -B(\rho\vec{u})^{n+1} + B(\rho\vec{u})^{n+(1/2)} = B(\rho\vec{u})^{n+(1/2)},$$

where $\delta p = p^{n+1} - p^{n}$. The result of the above Chorin type discretization is a Poisson-type equation for the pressure correction, which is decoupled from the momentum equations. There exists an extensive mathematical literature concerning first and second order fractional time step discretization, the incremental and non-incremental form of equations, stability, splitting of the boundary conditions, etc. Some recent results as well as further references can be found, e.g. in Brown *et al.* (2001), Minev (2001) and Armfield and Street (2002). Note that in the Newtonian case, the momentum equations discretized in this way are decoupled and can be solved consecutively. However, this is not the case for non-Newtonian fluids. An attempt to discretize off-diagonal viscous terms explicitly will lead to severe restrictions on the time step.

Fractional time step coupled method are an alternative for the projection methods. In coupled methods the pressure is treated implicitly in the momentum equations. Here we use an approach of this type:

$$(\rho\vec{u})^{n+(1/2)} = (\rho\vec{u})^{n} - \tau C_{\vec{u}}\vec{u}^{n} - \tau G p^{n},$$

$$(\rho\vec{u})^{n+1} - (\rho\vec{u})^{n+(1/2)} = \tau[D_{\vec{u}}\vec{u}^{n+1} - G p^{n+1} + G p^{n}], \quad B(\rho\vec{u})^{n+1} = 0.$$

We obtain again a Poisson-type operator with respect to the pressure correction from the second and third equations, but in this case the system remains coupled, i.e. the second and the transformed third equation have to be solved simultaneously. In fact, in this case we decouple only the convective and viscous transport, but the viscous and pressure forces remain coupled.

An interesting fractional time step discretization is suggested in Vabishchevich and Samarskii (2000), but this is not discussed here.

Finally, if an implicit discretization is used instead of the fractional time step approach, one gets the following equations:

$$(\rho\vec{u})^{n+1} + \tau(C_{\vec{u}}\vec{u}^{n+1} - D_{\vec{u}}\vec{u}^{n+1}) = (\rho\vec{u})^{n} - \tau G p^{n+1}, \quad B(\rho\vec{u})^{n+1} = 0.$$

4. Iterative methods for the discretized system

In this section we consider iterative methods for the solution of the coupled system of the discretized equations. Such a system arises after the implicit discretization, or after the fractional time step coupled discretization. We rewrite the large-scale system of linear algebraic equations to be solved at each time step $t = t^{n+1}$ as follows, introducing an obvious change in our notation to display the structure of the system more clearly:

$$\begin{pmatrix} A & \tau G \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} S_{\vec{u}} \\ 0 \end{pmatrix}. \tag{6}$$

Further on we use the abbreviated notations:

$$Lv = b$$

with $v = (\vec{u}, p)^t$. Let us briefly discuss projection and coupled approaches for an iterative solution of the above system.

4.1 Iterative projection methods

In this subsection we discuss iterative projection methods like SIMPLE. (A good systematization and detailed references for these methods can be found, e.g. in Turek's, 1999.) One way to derive such methods in the purely incompressible case proceeds as follows. Solving the first equation with respect to velocity yields:

$$\vec{u} = A^{-1}(S_u - \tau G p). \tag{7}$$

Substituting this result in the second equation, we obtain:

$$\tau B A^{-1} G p = B A^{-1} S_u. \tag{8}$$

As the matrix A^{-1} is not sparse, solving the above equation directly is prohibitively expensive. Instead, an iterative procedure (the preconditioned Richardson method) can be set-up:

$$p^{i+1} = p^i - M^{-1}(\tau B A^{-1} G p^i - B A^{-1} S_u) \tag{9}$$

This means that at each iteration we solve:

$$M \delta p = -(\tau B A^{-1} G p^i - B A^{-1} S_u) \tag{10}$$

The preconditioner should be spectrally close to $\tau B A^{-1} G$, and at the same time easily invertible. The usual choice is:

$$M = \tau B D^{-1} G,$$

where D is a diagonal matrix, so that M is identical to discretization of a second order elliptic operator. For example, in SIMPLE the choice is:

$$D = \mathrm{diag}\{A\}. \tag{11}$$

Let us now investigate the structure of the matrix A in a more detailed way for the cases of either constant or variable viscosity. In these cases the matrix acquires the following forms:

- Constant viscosity:

$$A = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{pmatrix}.$$

- Variable viscosity:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}.$$

The above representation shows that in the case of strong off-diagonal blocks the approximation of A with a diagonal matrix D should be rather bad. Note that even in the Newtonian case (block-diagonal matrix A) the decoupling approach might not be successful. An example are flow computations on stretched grids (Turek, 1999).

4.2 Iterative methods for the coupled system

An alternative to the decoupling approach are the coupled solvers. There are different coupled solvers. Few of them work with non transformed system (e.g. the Vanka approach Turek, 1999), others first transform the system and solve it afterwards. In the CFD literature coupled solvers are frequently applied as smoothers within nonlinear multigrid flow solvers (Turek, 1999; Ferry, 2002; Deng *et al.*, 2001). The coupled solvers require more memory compared to the segregated solvers. Some software developers (e.g. Fluent) provide both types of solvers. In general the area of applicability of coupled solvers is rather restricted. It should be mentioned that some authors consider coupled solvers to have advantages for steady state solutions, and decoupled techniques to be being preferable for unsteady problems (Turek, 1999; Oosterlee, 1993). However, very few particular cases have been analyzed completely, and further studies are necessary in order to confirm or reject the above statement. Another point are observations stating that

coupled solvers are more robust for some flow problems and less robust for others, but additional work is needed in order to study and classify the advantages of the two approaches, too.

In this paper we discuss coupled solvers based on proper preconditioning of the coupled system (equation (6)). A detailed discussion and further references concerning such type of the coupled solvers can be found, e.g. in Silvester *et al.* (2001), Axelsson and Neytcheva (2003) and Benzi (2004). The specifics of our approach are described below. A right block triangular preconditioner is used in Silvester *et al.* (2001) (the statement in that paper is that left, right, or double sided preconditioning are almost identical). The preconditioner they use is in the form:

$$M = \begin{pmatrix} A & R \\ 0 & S \end{pmatrix}.$$

So that:

$$LM^{-1} = \begin{pmatrix} I & -(R + \tau G)S^{-1} \\ BA^{-1} & -BA^{-1}RS^{-1} \end{pmatrix}.$$

We do not discuss the specific choice of operators R and S here, the interested reader will find them in Silvester *et al.* (2001).

Several preconditioners for equation (6) are carefully analyzed within the recent paper (Axelsson and Neytcheva, 2003). Among them are block Gauss-Seidel preconditioning, a preconditioner based on congruence transformation, and a two-sided block incomplete preconditioner. Considering a specific example, e.g. the block Gauss-Seidel preconditioner is given by:

$$M = \begin{pmatrix} D_1 & 0 \\ B & D_2 \end{pmatrix}.$$

So that:

$$M^{-1}L = \begin{pmatrix} D_1^{-1}A & \tau D_1^{-1}G \\ -D_2^{-1}B\left(D_1^{-1}A - I\right) & -\tau D_2^{-1}BD_1^{-1}G \end{pmatrix}.$$

For the particular choice of operators D_1 and D_2, as well as for a discussion on some other preconditioners, we refer to Axelsson and Neytcheva (2003).

In general (Barrett *et al.*, 1994) applying a preconditioned iterative method to solve a system of equations is equivalent to applying a non-preconditioned method for the solution the transformed system. The above approaches (Silvester *et al.*, 2001; Axelsson and Neytcheva, 2003) are examples of this type. Instead of using this approach, we use a two stage approach to solve equation (6). In the first stage we transform the system using a matrix like the preconditioners mentioned above. In the second stage, instead of using an unpreconditioned iterative method, we use a preconditioned one. Of course, this two-stage procedure can be written and analyzed as a one-stage one. We postpone such an analysis for another paper. Here we concentrate

on the algorithmical part and on the numerical study of the performance of the preconditioners. So, at the first stage we use the matrix:

$$M^{-1} = \begin{pmatrix} D & 0 \\ -BD & I \end{pmatrix}.$$

so that:

$$M^{-1}L = \begin{pmatrix} DA & \tau DG \\ B - BDA & -\tau BDG \end{pmatrix}.$$

The aim of this transformation is to obtain "good" blocks at the main diagonal of the transformed system. It is clear, that the choice $D = A^{-1}$ will lead to a block triangular system, but the operator BDG will have a full matrix in this case. So, like in SIMPLE, we select:

$$D = (\mathrm{diag}\{A\})^{-1}$$

Thus, we obtain a transformed system:

$$\tilde{L}v = \tilde{b}, \tag{12}$$

where $\tilde{L}$ is not a block triangular matrix. However, the blocks on the main diagonal are easily invertible (they are similar to a discretization of the Poisson equation). This system can be also written down as:

$$\begin{pmatrix} \tilde{A} & \tau\tilde{G} \\ \tilde{B} & \tilde{\Lambda} \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ b_2 \end{pmatrix}. \tag{13}$$

At the second stage we apply block triangular preconditioners to the transformed system.

5. Numerical experiments

Numerical experiments were performed in order to evaluate the performance of the linear solvers applied in the simulation of liquid polymer flow. We choose a simple test geometry shown in Figure 1 consisting of u-shaped pipe with a contraction and an expansion in the middle. It is well known that plastic melts do not obey Newtonian behavior, especially near the freezing temperature. Plastic melts show shear thinning behavior, to name one of the non-Newtonian flow properties commonly observed for this type of fluids. This leads to a necessity of using specific material models for the viscosity including temperature, shear rate and pressure dependence. Several rheology models are available in the material database of SIGMASOFT© for instance, the Cross-WLF model (Gramann *et al.*, 2001). It belongs to the so-called generalized Newtonian models, and it is used in our simulations. During the process the viscosity changes by several orders of magnitude.

We solve an unsteady problem, starting simulations from the liquid being at rest, and calculating until the steady state is reached. At each time step the heat equation is decoupled from the system due to an explicit treatment of the dissipative terms and of

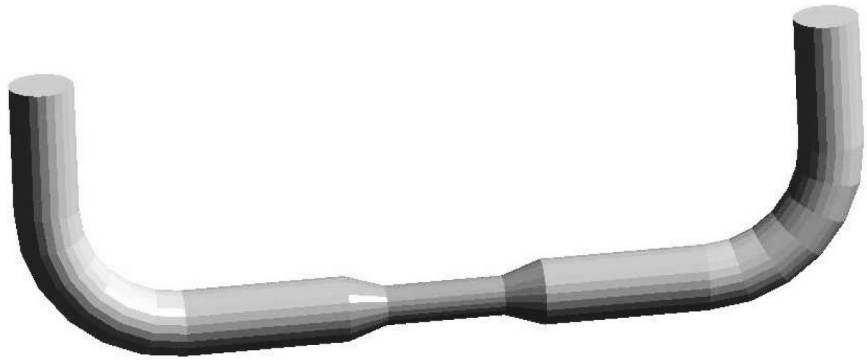

Figure 1.
Geometry for evaluating the linear solvers

the velocity in the convective term of the heat equation. The coupled momentum and continuity equations are transformed (as it was described at the end of the preceding section) and after that solved by a preconditioned iterative method. Several methods and preconditioners are applied. More precisely, the used iterative methods are:

- *CGS*. Conjugate gradient squared method;
- *BiCGStab*. Bi-conjugate gradient method, stabilized; and
- *GMRES(m)*. Generalized minimal residuum method.

We note that GMRES is the preferred method in Silvester *et al.* (2001) and Axelsson and Neytcheva (2003). As it will be shown below, in general GMRES is not suitable for us. The simulations show that good convergence can be achieved only using long sequences, which is impossible in 3D simulations due to the memory limitations. Note, that Silvester *et al.* (2001) and Axelsson and Neytcheva (2003) use "long enough" sequences, but the examples they solve are more of academic type.

Concerning the choice of the preconditioners, we have used three preconditioners for the transformed system (equation (13)). These are block diagonal, upper triangular (denoted by $T-$ in below equations), and lower triangular (denoted by TL). That is:

$$\begin{pmatrix} M_{11} & 0 \\ 0 & M_{22} \end{pmatrix}, \quad T = \begin{pmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{pmatrix}, \quad TL = \begin{pmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{pmatrix}$$

The following variants for the blocks in the preconditioners were used:

- $M_{12} = \tau\tilde{G}$; $M_{21} = \tilde{B}$;
- $M_{11} = \mathrm{ILU}_{\tilde{A}}$; $M_{11} = \mathrm{BRILU}_{\tilde{A}}$ for various α (calculated for diagonal blocks); $M_{11} = \tilde{A}^{-1}$; and
- $M_{22} = \mathrm{ILU}_{\tilde{\Lambda}}$; $M_{22} = \mathrm{RILU}_{\tilde{\Lambda}}$ for various α; $M_{22} = \tilde{\Lambda}^{-1}$.

In the case when the diagonal blocks have to be inverted, either CGS, Jacobi, GMRES, each preconditioned by RILU(α) or block RILU(α), were used. Additionally, algebraic multigrid, AMG (Stuben, 2001) was also used for inverting $\tilde{\Lambda}$.

We investigated the performance of the linear solvers with respect to grid size, flow type, accuracy, Reynolds number and the size of the time step. More precisely, Table I shows the following tests:

Here Δp stands for the pressure difference between the inlet and outlet. Figure 2 shows an impression about the non-Newtonian case: the variations in the viscosity are plotted there (recall that viscosity is a constant for Newtonian fluids). The variations in the viscosity here are moderate. In the simulations of solidification of real 3D plastic parts, the variation in the viscosity range over several orders of magnitude.

Below we show several tables with results from simulations. In all cases, the block diagonal preconditioner performed worse compared to the triangular ones, therefore it is not included in the tables.

First, we show four tables presenting results obtained without the usage of AMG.

As can be seen in Table II, GMRES with short restart sequences does not provide good results. The best results here are achieved by BiCGStab preconditioned with an upper triangular matrix, where the block M_{11} is a block RILU (0.8) factorization for the

Grid	Coarse, 4,584 fluid cells	Finer, 38,040 fluid cells
Flow type	Newtonian (constant μ.)	Non-Newtonian (variable μ)
Accuracy	Lower, $\varepsilon = 10^{-3}$	Higher, $\varepsilon = 10^{-8}$
Velocity	Slow, $\Delta p = 10$	Faster, $\Delta p = 30$
Time step	Smaller (0.005 s)	Larger (0.1 s)

Table I.
Performed tests for Venturi pipe flow

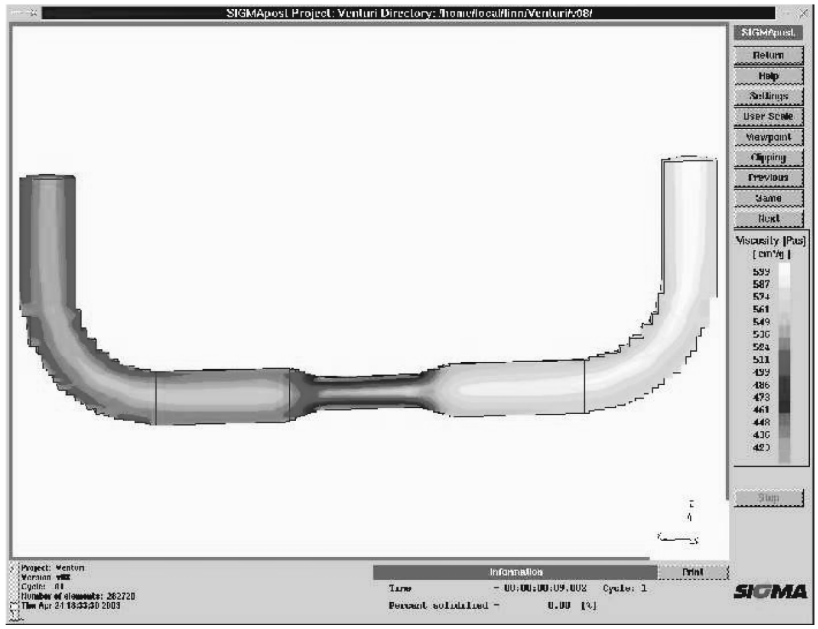

Figure 2.
Viscosity

main sub blocks of $\tilde{A}$, and $\tilde{\Lambda}$ is approximately inverted by two Jacobi iterations preconditioned with RILU (0.4) factorization. Performing more or less Jacobi iterations gives worse results.

GMRES was not used on the coarse grid (Table III) because of the bad results it showed on the fine grid. The best iterative method and the best preconditioner on the coarse grid are the same as on the finer grid.

Further, we show results for a non-Newtonian flow with higher variation in the viscosity (Table IV). This is a non-isothermal flow.

Figure 3 shows the behavior of some linear solvers during all the time steps. It can be seen that the convergence is more or less uniform during the time stepping.

Table II.
Number of iteration and CPU time for non-Newtonian fast flow on fine grid, $\varepsilon = 10^{-8}$

	CGS		BiCGStab		GMRES(10)	
Preconditioner	Iter.	CPU	Iter.	CPU	Iter.	CPU
T-BILU-ILU	228	31	232	31	>500	
T-BRILU0.8-RILU0.4	188	26	149	20	>500	
T-BRILU0.8-Jac2RILU0.4	138	23	115	18	>500	
T-CGS6BRILU0.8-RILU0.4	72	282	70	284	122	398
TL-BILU-ILU	224	37	213	34	>500	
TL-BRILU0.8-RILU0.4	180	30	142	23	>500	
TL-BRILU0.8-Jac2RILU0.4	138	27	115	22	>500	
TL-CGS6BRILU0.8-RILU0.4	60	247	46	195	140	308

Table III.
Number of iteration and CPU time for non-Newtonian fast flow on coarse grid, $\varepsilon = 10^{-8}$

	CGS		BiCGStab		GMRES(10)	
Preconditioner	Iter.	CPU	Iter.	CPU	Iter.	CPU
T-BILU-ILU	79	1.1	67	0.9		
T-BRILU0.8-RILU0.4	62	0.9	58	0.8		
T-BRILU0.8-Jac2RILU0.4	54	0.9	47	0.7		
T-CGS6BRILU0.8-RILU0.4	47	7.6	46	7.4		
TL-BILU-ILU	79	1.3	68	1.1		
TL-BRILU0.8-RILU0.4	61	1.0	56	0.9		
TL-BRILU0.8-Jac2RILU0.4	52	1.0	44	0.8		
TL-CGS6BRILU0.8-RILU0.4	43	6.9	33	5.6		

Table IV.
Number of iteration and CPU time for non-isothermal non-Newtonian fast flow on coarse grid, $\varepsilon = 10^{-8}$

	CGS		BiCGStab		GMRES(10)	
Preconditioner	Iter.	CPU	Iter.	CPU	Iter.	CPU
T-BILU-ILU	185	25.2	140	18.4	1,063	90.4
T-BRILU0.8-RILU0.4	129	17.6	113	14.8	889	75.6
T-BRILU0.8-*Jac2* RILU0.4	124	20.4	116	18.4	2,628	263.9
T-*CGS6* BRILU0.8-RILU0.4	90	262.5	83	239.7	133	220.2
TL-BILU-ILU	172	28.3	150	23.8	393	88.6
TL-BRILU0.8-RILU0.4	135	22.1	114	18.0	209	46.2
TL-BRILU0.8-*Jac2* RILU0.4	125	24.0	132	24.6	186	43.5
TL-*CGS6* BRILU0.8-RILU0.4	52	162.2	42	125.4	73	125.6

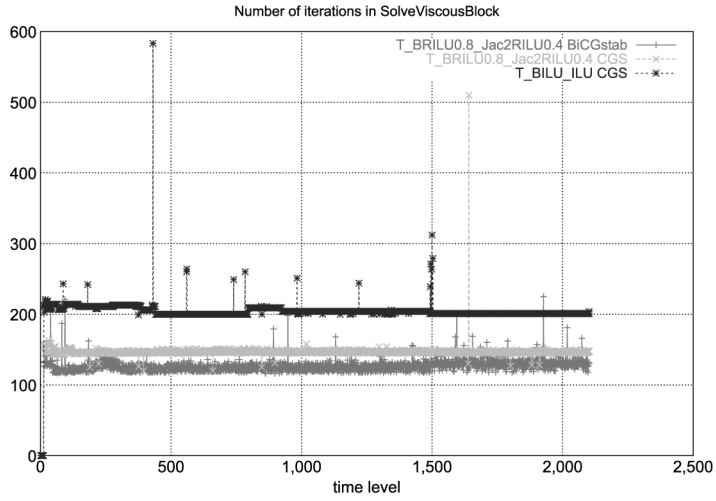

Figure 3.
Number of iterations at each time step

Preconditioner	CGS Iter.	CGS CPU	BiCGStab Iter.	BiCGStab CPU	GMRES(10) Iter.	GMRES(10) CPU
T-BILU-ILU	45	6.2	16	2.0	60	5.1
T-BRILU0.8-RILU0.4	17	2.3	16	2.0	36	3.1
T-BRILU0.8-Jac2RILU0.4	28	4.6	15	2.4	121	12.3
T-CGS6BRILU0.8-RILU0.4	18	51.8	12	32.1	29	44.6
TL-BILU-ILU	43	7.1	27	4.2	56	9.3
TL-BRILU0.8-RILU0.4	16	2.7	14	2.2	41	5.9
TL-BRILU0.8-Jac2RILU0.4	49	9.4	29	5.4	43	6.9
TL-CGS6BRILU0.8-RILU0.4	1	2.7	1	1.4	1	4.0

Table V.
Number of iteration and CPU time for non-isothermal non-Newtonian fast flow on coarse grid, $\varepsilon = 10^{-3}$

In certain cases there is no need to solve the linear system arising in the discretization of unsteady PDEs very accurately. In Table V we show results obtained when low accuracy, namely $\varepsilon = 1 \times 10^{-3}$, is required. We see that in this case the lower triangular preconditioner with an exact inverting of the viscous block shows the best results. This means that the convergence of the iterative methods is different when using the two different triangular preconditioners. The lower triangular preconditioner ensures fast reduction of the residual at the beginning of the iterative process, but later the convergence is slower compared to that obtained with the upper triangular preconditioner.

It should be noted that for many industrial problems an approximate solution of the linear system of equations is reasonable. In our case the results obtained with $\varepsilon = 1 \times 10^{-3}$ and $\varepsilon = 1 \times 10^{-8}$ in many cases show less than 10 percent difference.

Finally we show some results obtained using algebraic multigrid for inverting Λ (Table VI). Here we observe that the results with AMG are superior to the other results.

The results obtained with the test geometry have been confirmed with several industrial plastic parts (see an example on Figure 4). The speedup of the combination of RILU and AMG preconditioner lead to a speedup up to factor 4 for complicated geometries with more than 100.000 fluid cells.

6. Summary

The performance of different preconditioners and iterative methods was studied for a Venturi geometry, for fast and slow, Newtonian and non-Newtonian flow, on coarse and finer grids, and for $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-8}$, using different time steps.

The results for Newtonian and non-Newtonian flows are very close in our calculations. A reason for this might be that we always treat the viscous block (momentum equations) in a coupled way. A more detailed comparison with a fully

$\tilde{A}$ Preconditioner	$\tilde{\Lambda}$ Preconditioner	Iterative method	Iterations	Time [s]
ILU	ILU	CGS	182	25.6
ILU	ILU	BiCGstab	194	26.3
RILU 0.8	RILU 0.4	CGS	124	17.5
RILU 0.8	RILU 0.4	BiCGstab	103	14.0
RILU 0.8	Jac 2 RILU 0.4	CGS	105	17.7
RILU 0.8	Jac 2 RILU 0.4	BiCGstab	84	13.7
CGS 6 RILU 0.8	RILU 0.4	CGS	82	172.5
CGS 6 RILU 0.8	RILU 0.4	BiCGstab	68	144.8
RILU 0.8	AMG V	CGS	44	≈6.9
RILU 0.8	AMG V	BiCGstab	41	≈5.0

Table VI.
Number of iteration and CPU time for non-isothermal non-Newtonian flow on fine grid, $\varepsilon = 10^{-8}$

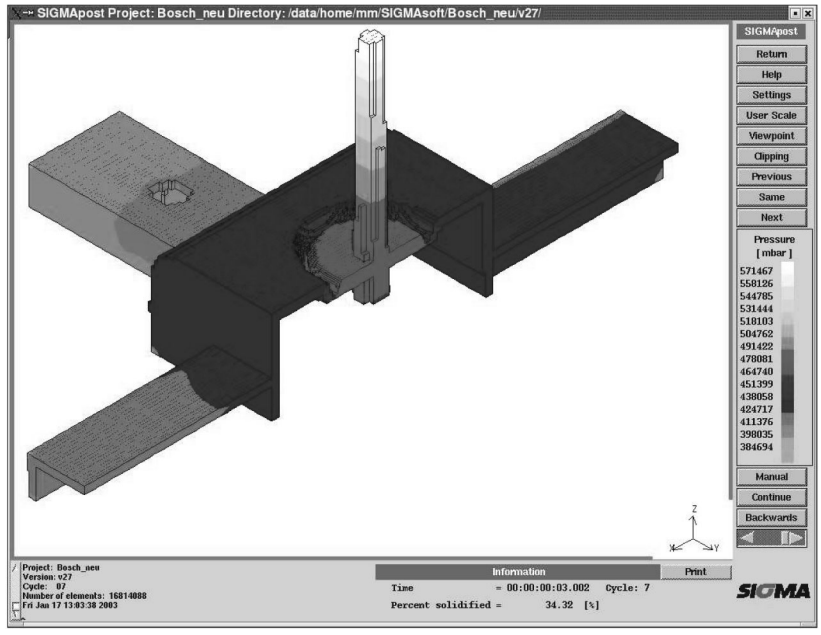

Figure 4.
Pressure during compaction stage in polymer solidification

segregated solution is in progress now and will be a subject of another paper. Another reason might be the relatively small time step used in our calculations.

We have simulated relatively slow flows, and we did not observe significant differences between our slower and faster regimes. The choice of the velocities in our case was motivated by some practical applications (packing phase of injection molding), for other applications the behavior of the solvers might be different.

BiCGStab has shown the most robust behavior. It was the fastest method in our tests. GMRES needs long sequences (e.g. 100) for a good convergence, which is inappropriate for 3D industrial applications.

A lower triangular preconditioner with inversion of the viscous block was the best choice for lower accuracy ($\varepsilon = 10^{-3},\ \varepsilon = 10^{-4}$).

An upper triangular preconditioner with inversion of the Laplacian by an algebraic multigrid solver was the best choice for high accuracy ($\varepsilon = 10^{-8}$).

The results presented here concern the performance of the linear solvers for a particular finite volume discretization and for a particular class of flows. Further studies are planned to better understand the performance of the solvers for other geometries, for other non-Newtonian flows and other discretizations.

References

Armfield, S. and Street, R. (2002), "An analysis and comparison of the time accuracy of fractional step methods for the Navier-Stokes equations on staggered grids", *International Journal of Numerical Methods in Fluids*, Vol. 38, pp. 255-82.

Axelsson, O. and Neytcheva, M. (2003), "Preconditioning methods for linear systems arising in constrained optimization problems", *Numerical Linear Algebra with Applications*, Vol. 10, pp. 1-26.

Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and Van der Vorst, V. (1994), *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA.

Benzi, M. (2004), "HSS preconditioners for the Oseen problems", in Neitaanmaki, P. *et al.* (Eds), *Proc. ECCOMAS*, available at: www.mit.jyu.fi/eccomas2004/proceedings/proceed.html

Brown, D., Cortez, R. and Minion, M. (2001), "Accurate projection methods for incompressible Navier-Stokes equations", *Journal of Computational Physics*, Vol. 168, pp. 464-99.

Churbanov, A. (2003), "A unified algorithm to predict both compressible and incompressible flows", *Lecture Notes in Comput. Science*, Springer, New York, NY, Vol. 2542, pp. 412-9.

Core, P. and Angot, Ph. (2002), "A multilevel FIC projection method for low Mach natural convection flows", in Herbin, R. and Kroener, D. (Eds), *Finite Volumes for Complex Applications III*, Hermes Penton Ltd, London.

Deng, G., Piquet, J., Vasseur, X. and Visonneau, M. (2001), "A new fully coupled method for computing turbulent flows", *Computers & Fluids*, Vol. 30, pp. 445-72.

Ferry, M. (2002), "New features of MIGAL solver", *PHOENICS Journal Computational Fluid Dynamics and its Applications*, Vol. 15, available at: www.cham.co.uk/phoenics/d_polis/d_enc/migal/enc_mig.htm

Ferziger, J.H. and Peric, M. (1999), *Computational Methods for Fluid Dynamics*, Springer, New York, NY.

Fletcher, C. (1991), *Computational Techniques for Fluid Dynamics*, Springer, New York, NY.

Gramann, P.J., Osswald, T.A. and Turng, L.S. (2001), *Injection Molding Handbook*, Hanser Gardner Publications, Cincinnati, OH.

Gresho, P. and Sani, R. (1998), *Incompressible Flow and Finite Element Method*, Wiley, New York, NY.

Hattel, J. (Ed.) (2005), *Fundamentals of Numerical Modelling of Casting Processes*, Polyteknisk Forlag, p. 540.

Minev, P. (2001), "A stabilized incremental projection scheme for the incompressible Navier-Stokes equations", *International Journal of Numerical Methods in Fluids*, Vol. 36, pp. 441-64.

Oosterlee, C.W. (1993), "Robust multigrid methods for steady and unsteady incompressible Navier-Stokes equations in general coordinates", PhD thesis, TU Delft, Delft.

Silvester, D., Elman, H., Kay, D. and Wathen, A. (2001), "Efficient preconditioning of the linearized Navier-Stokes equations", *Journal of Computational and Applied Mathematics*, Vol. 128, pp. 261-79.

Stuben, K. (2001), "A review of algebraic multigrid", *Journal of Computational and Applied Mathematics*, Vol. 128, pp. 281-309.

Turek, S. (1999), "Efficient solvers for incompressible flow problems", *An Algorithmic and Computational Approach*, Springer, New York, NY, p. 352.

Vabishchevich, P.N. and Samarskii, A.A. (2000), "Problems in compressible fluid dynamics with a variable viscosity", *Journal of Computational Mathematics and Mathematical Physics*, Vol. 40 No. 12, pp. 1741-50.

Further reading

Wesseling, P. (2001), *Principles of Computational Fluid Dynamics*, Springer, New York, NY.

Corresponding author

Oleg Iliev can be contacted at: iliev@itwm.fhg.de