INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 1141–1156 (1999)

PRECONDITIONED KRYLOV SUBSPACE METHODS USED IN SOLVING TWO-DIMENSIONAL TRANSIENT TWO-PHASE FLOWS

MAGNUS NORDSVEEN[a,*] AND RANDI MOE[b]

[a] *Institute for Energy Technology, PO Box 40, 2007 Kjeller, Norway*
[b] *Aker Engineering AS, Tjuvholmen, N-0250 Oslo, Norway*

SUMMARY

This paper investigates the performance of preconditioned Krylov subspace methods used in a previously presented two-fluid model developed for the simulation of separated and intermittent gas–liquid flows. The two-fluid model has momentum and mass balances for each phase. The equations comprising this model are solved numerically by applying a two-step semi-implicit time integration procedure. A finite difference numerical scheme with a staggered mesh is used. Previously, the resulting linear algebraic equations were solved by a Gaussian band solver. In this study, these algebraic equations are also solved using the generalized minimum residual (GMRES) and the biconjugate gradient stabilized (Bi-CGSTAB) Krylov subspace iterative methods preconditioned with incomplete LU factorization using the ILUT(p, τ) algorithm. The decrease in the computational time using the iterative solvers instead of the Gaussian band solver is shown to be considerable. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: two-phase flow; two-fluid model; GMRES; Bi-CGSTAB; ILUT

1. INTRODUCTION

The nuclear industry, the petroleum industry and the chemical process industry, to mention a few, are all users of multiphase/multicomponent flow technology. One example is the transport of oil and gas in pipelines, which can now be modelled by several one-dimensional, two/three-phase flow codes (OLGA [1], PLAC [2], PEPITE [3], etc.). In order to model these complex three-dimensional flow patterns with one-dimensional models, many simplifications are introduced, guided by extensive experimental investigations. Several general purpose codes (FLUENT [4], PHOENICS [5], FIDAP [6], etc.) have two- and three-dimensional models for at least two-phase/two-component flows. Industrial applications may comprise turbulent multiphase flow in complex geometries for which closure laws for these two- and three-dimensional models can be a weak point. To easily test new and improved closure laws, usually based on experimental investigations, there is a need for robust and fast numerical methods. Popular numerical methods used for multiphase flows, like IPSA [7], SIMPLE-2P [8] and PISO-2P [9], solve the equations sequentially with many iterations to obtain a converged solution. The difficulty with these methods for obtaining a converged solution is in favour of more coupled solution strategies. On the other hand, more coupled solution methods demand good algebraic

* Correspondence to: Studsvik Scandpower AS, PO Box 15, NO-2007 Kjeller, Norway.

CCC 0271–2091/99/231141–16$17.50

Received July 1998
Revised October 1998

equation solvers. In this work, we investigate the behaviour of preconditioned Krylov subspace methods for solving algebraic equations systems resulting from a coupled solution strategy implemented by Moe and Bendiksen [10] for the two-fluid model [11,12].

The two-fluid model comprises conservation equations for two 'continuous fluids' coexisting in each point in the computational domain. In this work, the two fluids are liquid and gas. The fluids can be separated (two continuous phases) and/or distributed (bubbles in liquid, droplets in gas). The model has been used [10,13] to simulate the broken dam problem, collapse of a liquid column, separation of mixed gas and liquid, propagation of a single Taylor bubble in horizontal and inclined channels and the initiation of slugs in upward inclined channels. By specifying a zero fraction of one of the fluids, single-phase flow cases have been simulated as well.

In the numerical solution procedure implemented by Moe and Bendiksen [10], a pressure equation is solved simultaneously with the momentum equations for the two phases. Mass equations for each phase are solved in a separate step. This two step procedure introduces small errors only. The method is aimed at transient flows and at each time step a source/sink term is added to the pressure equation to correct the errors introduced at the previous time step. With this correction, a robust non-iterative scheme is obtained. In previous works [10,13], the algebraic equation system resulting from the discretization of the pressure and momentum equations as well as for the mass equations, were solved with a Gaussian band solver. In the present work, the pressure–momentum algebraic equations are also solved with preconditioned Krylov subspace iterative algorithms [14]. The algebraic mass equations, which are considerably faster to solve, are still solved with the Gaussian band solver.

Krylov subspace methods are one of the most popular iterative methods for solving sparse linear systems. The most well known of these methods are probably the conjugate gradient (CG) algorithm due to Hestenes and Steifel [15] applicable to symmetric positive-definite matrices. However, the matrix resulting from the discretized pressure–momentum system is asymmetric and indefinite and we have used the GMRES algorithm of Saad and Schultz [16] and the Bi-CGSTAB algorithm of Van der Vorst [17]. To obtain a good convergence rate for Krylov subspace methods applied to computational fluid dynamics (CFD) problems, the equation systems require preconditioning, see, e.g. Rusten and Winther [18], Elman and Sylvester [19], Ajmani and Ng [20], and Liang and Lan [21], who all applied preconditioned Krylov subspace methods. In the present work the incomplete LU factorization algorithm ILUT(p, τ) [22] is used as a preconditioner. Here, p is the maximum fill-in allowed and τ is a threshold value used to reduce the fill-in. The implementations of the ILUT(p,τ), GMRES and Bi-CGSTAB algorithms used were the ones found in the SPARSKIT library due to Saad [23].

2. PHYSICAL TWO-FLUID MODEL

Local instantaneous conservation equations for each phase with transfer rates of mass, momentum and energy at all interfaces can, in principle, be written for a two-phase flow system. It is only possible to solve this system for the simplest cases. However, in many practical applications average quantities are of primary interest. One approach is thus to time/ensemble and space-average the equations to obtain the simpler two-fluid model for the average fields (see, e.g. Ishii [12] or Banerjee and Chan [24]). In the averaging process new 'unknown' quantities appears. To close the equation system these new terms must be modelled. The closure of the two-fluid model used in this work was outlined by Moe and Bendiksen [10]. Here we only state the main assumptions and present the resulting equations. Two dimensional

Cartesian and axisymmetric problems are regarded where y denotes the vertical or radial direction and x denotes the horizontal or axial direction.

2.1. Mass and momentum conservation equations

The fluids are assumed Newtonian and compressible subject to a constant gravitational body force component. Surface tension between the two phases is neglected and a common pressure field is applied. Constant temperature is assumed. Mass transfer between the phases is neglected and the interactions between the two phases are restricted to skin friction only. Since we have no knowledge of where the interfaces are located, or of their direction, the skin friction force will have a component in both directions and will be applied to the whole computational domain. The mass and momentum balances for each phase k ($k = l, g$) are expressed as follows:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\alpha_k \rho_k \mathbf{u}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k \mathbf{u}_k) = -\alpha_k \nabla p + \alpha_k \rho_k \mathbf{g} + \nabla \cdot \left[\alpha_k \mu_{k,\mathrm{eff}} \left(\nabla \mathbf{u}_k + \nabla \mathbf{u}_k^T - \frac{2}{3} \nabla \cdot \mathbf{u}_k I \right) \right] + \mathbf{F}_{ki} |\mathbf{u}_r| \mathbf{u}_r, \tag{2}$$

where $\mathbf{u}_k$ denotes the phase velocity, $\mathbf{u}_r$ the relative velocity between the two phases, p the pressure, ρ_k the density, $\mu_{k,\mathrm{eff}}$ the effective dynamic viscosity, $\mathbf{g}$ the acceleration due to gravity (vertical direction), α_k the volumetric fraction of phase k, $(\ )^T$ the transpose and I denotes the identity matrix. The conservation of volume is expressed as $\alpha_g + \alpha_l = 1$. $\mathbf{F}_{ki}$ is the interfacial drag coefficient given by

$$\mathbf{F}_{gi} = -\mathbf{F}_{li} = \frac{1}{2} \rho_g \frac{S_i}{A} (\lambda_i^x, \lambda_i^y), \tag{3}$$

where S_i and A are the interfacial perimeter and the volume of the discrete control volume respectively. Different models for the interfacial friction factors λ_i^x and λ_i^y were tested by Moe and Bendiksen [10]. We have used the constant values $\lambda_i^y = 10^{-3}$ and $\lambda_i^x = 10^6$, entailing a weak coupling in the vertical direction and a strong coupling (no-slip) in the horizontal direction. This model gave reasonable results for the different cases tested by Moe and Bendisken [10] and Moe [13]. The effective viscosity is given by $\mu_{k,\mathrm{eff}} = \mu_k + \mu_k^T$, where μ_k and μ_k^T are the molecular and eddy viscosity respectively. The eddy viscosity is modelled by the modified Prandtl mixing length formula [10]

$$\mu_k^T = \rho_k l_{\mathrm{m}}^2 \left| \frac{\partial v_k}{\partial x} + \frac{\partial}{\partial y}(u_k \alpha_k) \right|, \tag{4}$$

where the formula of Nikuradse [25] for single-phase flow is used for the mixing length l_{m}. v_k and u_k denote the velocity components in the y- and x-directions respectively.

2.2. Equation of state and pressure equation

Compressible flow is assumed with the density given by an equation of state $\rho_k = \rho_k(p, T)$. With the assumption of a constant temperature, an equation for the pressure is obtained by expanding the transient terms in the mass conservation equations (using the equation of state), dividing by the densities and adding the resulting equations. The pressure equation reads:

$$\left(\frac{\alpha_g}{\rho_g}\frac{\partial \rho_g}{\partial p}+\frac{\alpha_l}{\rho_l}\frac{\partial \rho_l}{\partial p}\right)\frac{\partial p}{\partial t}=-\frac{1}{\rho_g}\nabla\cdot(\alpha_g\rho_g\mathbf{u}_g)-\frac{1}{\rho_l}\nabla\cdot(\alpha_l\rho_l\mathbf{u}_l). \tag{5}$$

2.3. Boundary and initial conditions

At the inlet of a pipe or a channel, the velocities and the phase fractions of each phase are specified. At the outlet of a pipe, a hydrostatic pressure profile is used. In the case of closed inlet or outlet boundaries, zero-velocity type boundary conditions are imposed. Boundary conditions for velocity components parallel to the walls are either no-slip or free-slip boundary conditions. The normal-to-wall velocity components are set to zero. Initial conditions have to be specified for phase fractions and velocities. The initial pressure field is then calculated according to hydrostatic conditions.

3. NUMERICAL METHOD

Equations (1), (2) and (5) together with the constitutive relations described above yield a set of time-dependent coupled non-linear partial differential equations (PDEs). These equations are solved using a backward Euler semi-implicit finite difference scheme as outlined in detail by Moe and Bendiksen [10]. Here we will first give a brief overview of the overall method before presenting in more detail the solution of the algebraic equations with preconditioned Krylov subspace methods. As indicated in Figure 1, a staggered spatial grid with scalars stored in the centre of the control volumes and the velocity component stored on the boundaries is applied.

Using implicit time integration, the tight coupling between the different flow field variables requires, in general, a simultaneous or iterative solution of these equations. In the numerical method used in this work each time step is split into two parts. First, the momentum equations and the pressure equation are solved simultaneously using old masses, and second, the mass equations are solved using the most recent velocity field. This decoupling is a part of the linearization of the equation system and generally the non-linear full equation set is not satisfied after these two steps. Especially, it is observed that this two-step procedure gives a mismatch between conservation of specific mass and the volume fractions. This is because the density is calculated from the pressure and volume fraction times density equals specific mass. One way of remedying this error, as well as other errors due to the linearization of the

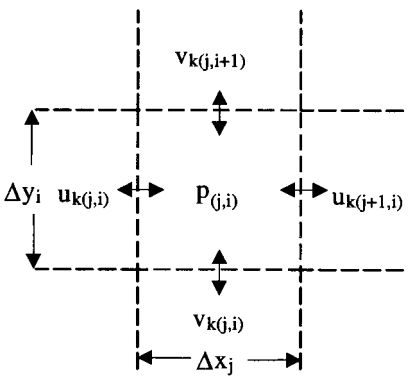

Figure 1. Finite difference staggered grid scheme.

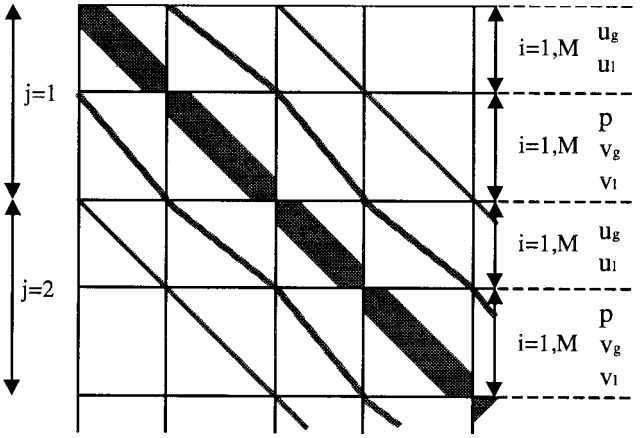

Figure 2. Pressure–velocity matrix structure.

equation set, would be to iterate by repeating the two above-described steps until convergence. This is not performed, however, instead a source/sink term based on the mass/volume fraction errors is added to the pressure equation at the next time point. With this correction the method has been experienced to give a stable solution with small mass/volume fraction errors only. The time step is limited by the Courant–Friedrich–Levi (CFL) criterion: $\Delta t < \min(\Delta x_j/u_k, \Delta y_i/v_k)$, where u_k and v_k are the phase velocities and Δx_j and Δy_i are the spatial control volume lengths in the x- and y-directions respectively.

For each of the two steps described above, an algebraic equation system $Ax = b$ has to be solved. The matrices A are banded and sparse. With N nodes in the x-direction and M nodes in the y-direction, the bandwidth and length of the pressure–velocity matrix become $10M - 3$ and $N(5M - 2)$ respectively. In Figure 2, a sketch of the structure of this matrix as well as the ordering of the five unknowns are shown. Beginning on the left of the domain, the y-direction index runs first over the momentum equations in the x-direction followed by the pressure equation and the momentum equations in the y-direction, before advancing one step in the x-direction and repeating the procedure.

In the present work we have used both a Gaussian elimination band solver as well as Krylov subspace iterative algorithms [14] to solve the discretized pressure–momentum equations. The mass equation systems are solved by Gaussian elimination.

A Krylov subspace is given by $K_m(A, v) = \text{Span}\{v, Av, \ldots, A^{m-1}v\}$ and a Krylov subspace method can be regarded as a projections method onto a subspace $K = x_0 + K_m(A, r_0)$ in which an approximation x_m to the solution of the equations system lies. Here x_0 and r_0 are the initial guess and the resulting residue respectively and m is the number of iterations. The projection is found by imposing that the residual vector $r = b - Ax$ is orthogonal to m independent vectors that span a subspace L. The most well known of these methods are probably the conjugate gradient (CG) method applicable to symmetric positive definite matrices. However, the matrix resulting from the discretized pressure–velocity system is asymmetric and indefinite and we have used the more general applicable algorithms; restarted GMRES [16] and Bi-CGSTAB [17].

The GMRES algorithm is based on creating an orthonormal basis for the Krylov subspace using the Arnoldi orthogonalization method [14], and minimalization of the residue norm over all vectors in K. The subspace L is then equal to AK_m. However, there is a potential storage

(and computational) problem with the Arnoldi orthogonalization method for large m since all bases vectors for the Krylov subspace must be saved and used in the calculations. This is circumvented by restarting the algorithm after M iterations and using x_m as the new initial guess x_0. In this work, M was taken as 15. The Bi-CGSTAB algorithm on the other hand can be regarded as based on Lanczos biorthogonalization [14], where non-orthogonal bases for the two Krylov subspaces $K_m(A, v_1)$ and $K_m(A^T, w_1)$ are built such that the bases vectors v_i and w_j for the two subspaces are biorthogonal, i.e. $(v_i, w_j) = \delta_{ij}$. In this procedure there is no need for storing the bases vectors and a simple recursion results. Note that the actual calculations with the transpose matrix A^T are avoided in the Bi-CGSTAB algorithm.

It has already been mentioned that matrices resulting from CFD problems are poorly conditioned and there is a need for preconditioning the matrices to obtain reasonable convergence rates for Krylov subspace methods. This is also the case for our pressure–velocity matrix. We apply a right preconditioner and the matrix system reads

$$AM^{-1}u = b, \qquad M^{-1}u = x.$$

The solution of the new unknown u is sought in the right preconditioned subspace $K = u_0 + K_m(AM^{-1}, r_0)$. For the preconditioner we have used the ILUT(p, τ) algorithm [22], which is an incomplete LU factorization of A. Our matrix A is a banded sparse matrix with many zeros within the band as indicated in Figure 2. In an LU factorization these zeros will typically be replaced by non-zero values. In an incomplete factorization one seeks to retain some (or all) of the sparseness of the original matrix A by applying a dropping strategy to each non-zero coefficient of the L and U factors. Three rules have been followed [22]. First, the locations with non-zero values in the original matrix A have been retained. Second, a coefficient in row i is dropped if its absolute value is less than the relative tolerance τ_i obtained by multiplying the given value τ by the original norm of the ith row. Third, only the p largest coefficients in row i in L and the p largest coefficients in row i in U are kept (in addition to the coefficient kept obeying the first rule).

4. PRESENTATION AND DISCUSSION OF RESULTS

The model has been developed for separated gas–liquid flows. However, single-phase flow is a simplified case where the gas or liquid volume fraction is set to zero. Three different cases have been simulated; single-phase turbulent channel flow, propagation of an elongated bubble in a channel and the broken dam problem. We are interested in the performance of the equation solvers and refer to the previous works [10,13] for a more detailed presentation of the flow field for these cases.

4.1. Single-phase turbulent pipe flow

Moe and Bendiksen [10] showed that the presented model and numerical method reproduced the analytical laminar solution for single-phase flow in a horizontal channel and in a pipe. They also simulated single-phase turbulent pipe flow, the case we have chosen to investigate here. Three different grid resolutions have been used. In the axial flow direction (x-direction) the number of nodes was 20, 30 and 40, corresponding to 6, 9 and 12 nodes respectively in the radial direction (y-direction). From experience, a relative error tolerance of $1.0\mathrm{E}-7$ for stopping the iterations of the iterative solver is sufficient to obtain a solution that agreed with the one obtained with the Gaussian band solver. If the number of iterations for a time step exceeded 300 the simulation was terminated. The threshold value for ILUT was $1.0\mathrm{E}-6$.

Table I. Single-phase turbulent pipe flow simulations: comparisons of used CPU time per time step for the different equation solvers

Grid	Gauss (CPU s)	P-GMRES (CPU s)	P-Bi-CGSTAB (CPU s)
20×6	0.49	0.11	0.11
30×9	2.77	0.50	0.52
40×12	10.4	1.51	1.55

In Table I, CPU time per time step for solving the pressure–momentum equation system for the different equations solvers are shown. It can be seen that the ILUT preconditioned GMRES (P-GMRES) and Bi-CGSTAB (P-BICGSTAB) equations solvers results in similar CPU times. For the course grid these iterative solvers are about 4.5 times faster than the direct Gaussian band solver, while for the finest grid, the speed-up has increased to about 6.8 times. In Figure 3, used CPU time per time step for the preconditioned iterative solvers are shown as function of fill-in. For the 20×6 grid, a fill-in of 9 was necessary to reach the error tolerance in less than 300 iterations per time step. For the 30×9 and 40×12 grids the minimum fill-in were increased to 15 and 20 respectively. It is seen that there are very small differences between the two different Krylov subspace methods. In Figure 4 the CPU times for the preconditioned GMRES algorithm are split into the parts used by the ILUT preconditioner and the GMRES equation solver. It is seen that initially the preconditioner cost increases while the GMRES cost decreases with increasing fill-in until constant values are reached. The decrease in the cost for the GMRES algorithm with increasing fill-in is caused by a reduced number of iterations, as shown in Figure 5. Note that the number of iterations for the three different grid resolutions become the same with increasing fill-in.

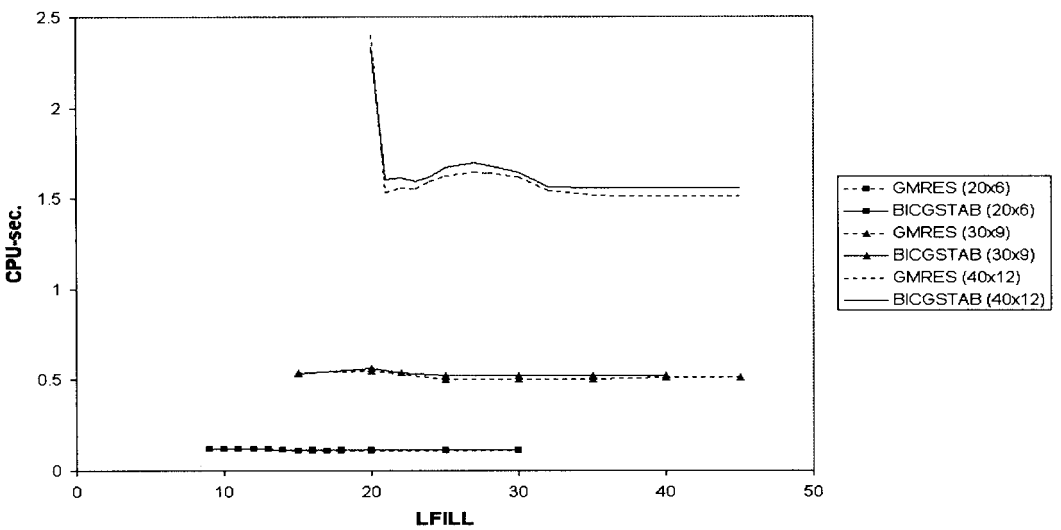

Figure 3. Average CPU time for a time step as function of fill-in for the preconditioned GMRES and BICGSTAB algorithms.

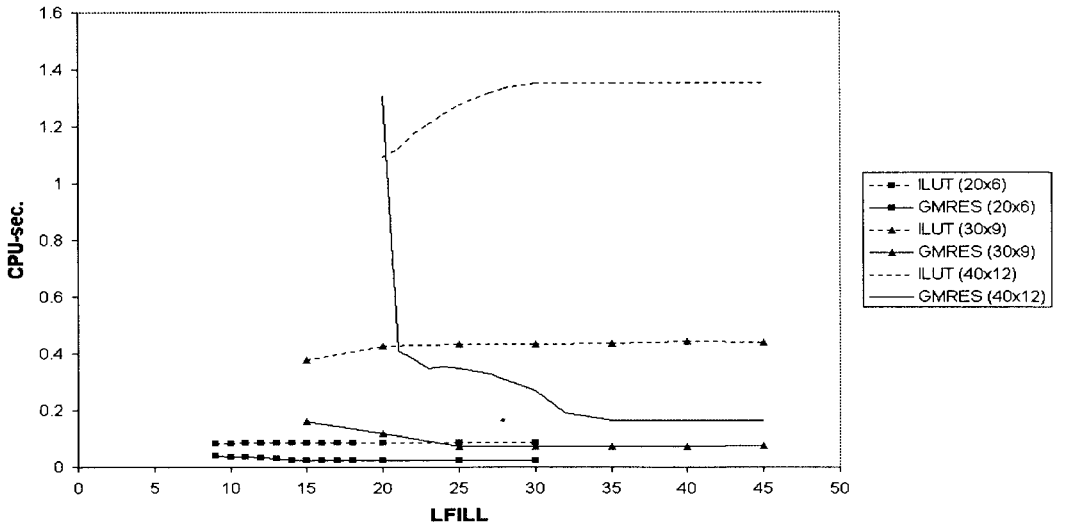

Figure 4. Average CPU time for a time step as function of fill-in for the ILUT preconditioner and the GMRES equation solver.

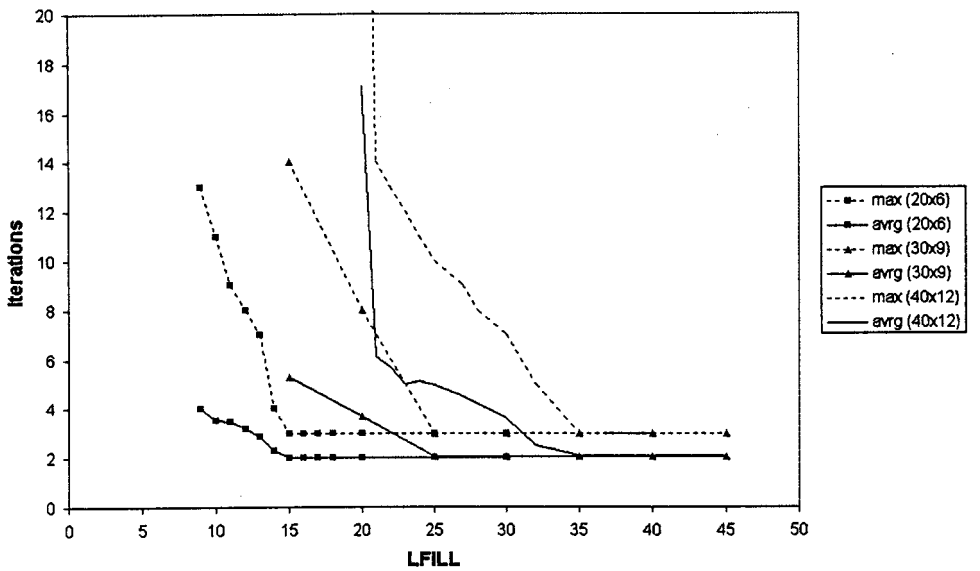

Figure 5. Maximum and average number of iterations for a time step using the P-GMRES algorithm.

4.2. Propagation of a single Taylor bubble in a horizontal channel

The propagation of an elongated bubble in a channel of height 0.7 m initially filled with liquid, closed at one end and open to the atmosphere at the other, has been simulated. The fluid properties used were those of water and air. The interface is assumed to be located where the volumetric fraction is 0.5. Using this approach, the calculated bubble shape and velocity field in the liquid for a given time point are shown in Figure 6.

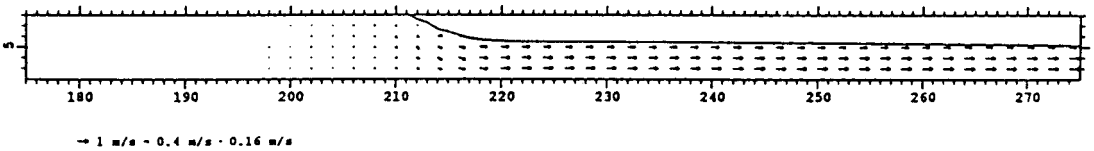

Figure 6. Calculated bubble shape and liquid velocity field.

The same three grid resolutions are used for this case as for the single-phase flow case. In Table II used CPU time per time step for solving the pressure–momentum equation system for the different equations solvers are shown. Also for this case it is seen the preconditioned Krylov subspace solvers use very similar CPU time. For the course grid, the iterative solvers are about 2.1 times faster than the Gaussian band solver, while for the finest grid, the speed-up has increased to about four times. Comparing with the single-phase flow case (see Table I) it is seen that the Gaussian solver uses the same CPU time for both cases, while the used CPU time for the iterative methods are about twice as large for the two-phase flow case. This difference for the iterative solvers performance for single-phase to two-phase flow is probably mainly caused by the reduction in the number of dependent variables. For single-phase flow cases, the off-diagonal elements in the rows corresponding to the missing phase are zero. There will thus be zero fill-in for these rows. This reduction of the system is not utilized by the Gaussian band solver. Note also that for a single-phase flow case there will initially only be five non-zero coefficients per row, while for two-phase flow cases there is nine non-zero coefficients per row.

In Figure 7 the used CPU time per time step for the preconditioned iterative solvers are shown as function of fill-in. For the 20×6 grid a fill-in of 11 was necessary to reach the error tolerance in less than 300 iterations per time step. For the 30×9 and 40×12 grids, the minimum fill-ins were increased to 16 and 23 respectively. It is seen that there are small differences between the two different Krylov subspace methods. In Figure 8 the CPU times for the preconditioned GMRES algorithm are split into the parts used by the ILUT preconditioner and the GMRES equation solver. It is seen that initially the preconditioner cost increases while the GMRES cost decreases with increasing fill-in until constant values are reached. In Figure 9, the effect of the threshold value is shown by plotting the total CPU time for the preconditioned GMRES and Bi-CGSTAB algorithms as function of fill-in. Of the three threshold values tested, $1.0\text{E}-6$ gives the most effective solvers. In Figure 10 the corresponding iterations for the preconditioned GMRES algorithm for the three different threshold values are shown. The threshold value $1.0\text{E}-6$ gives the lowest number of iterations.

Table II. Propagating Taylor bubble simulations: comparisons of used CPU time per time step for the different equation solvers

Grid	Gauss (CPU s)	P-GMRES (CPU s)	P-Bi-CGSTAB (CPU s)	Gauss/P-GMRES
20×6	0.51	0.24	0.23	2.1
30×9	2.72	0.91	0.9	3
40×12	10.3	2.56	2.54	4

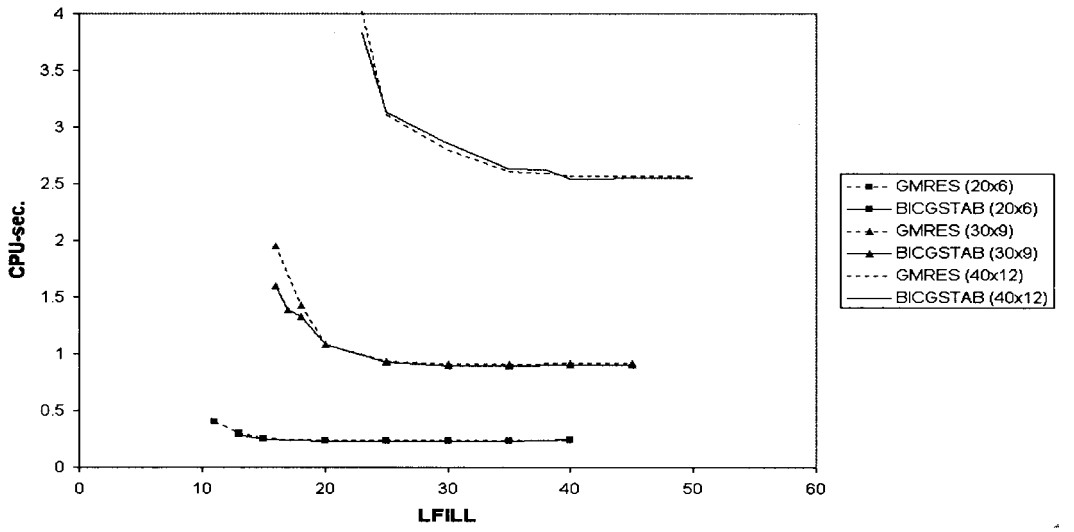

Figure 7. Average CPU time used per time step as function of fill-in for the P-GMRES and Bi-CGSTAB algorithms. The threshold value is setequal to 1.0E − 6.

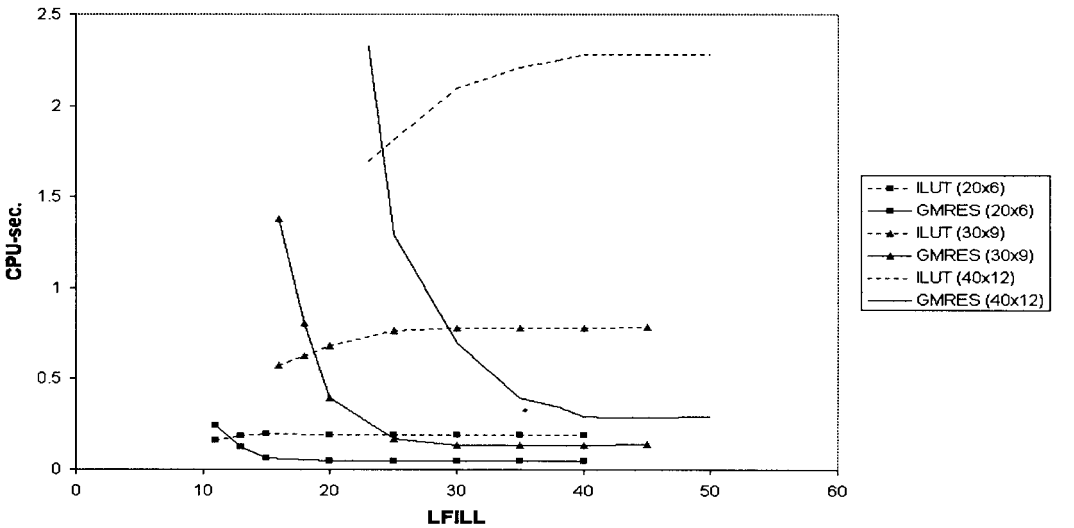

Figure 8. Average CPU time used per time step as function of fill-in for the ILUT preconditioner and the GMRES equation solver. The threshold value is set equal to 1.0E − 6.

4.3. Broken dam problem

A rectangular column of water in hydrostatic equilibrium is confined between two vertical walls and a horizontal bottom. The right wall is suddenly removed, and the water column starts to collapse under the influence of gravity. The initial column is 1.0 m wide and 2.0 m high. Test fluids are water and air at atmospheric conditions. In Figure 11 contour lines of the void fraction equal to 0.5 is shown for six different time points. Both results using the Gaussian band solver and the GMRES iterative solver are presented. The two equation solvers give almost identical results and the contour lines can scarcely be separated in this plot. This

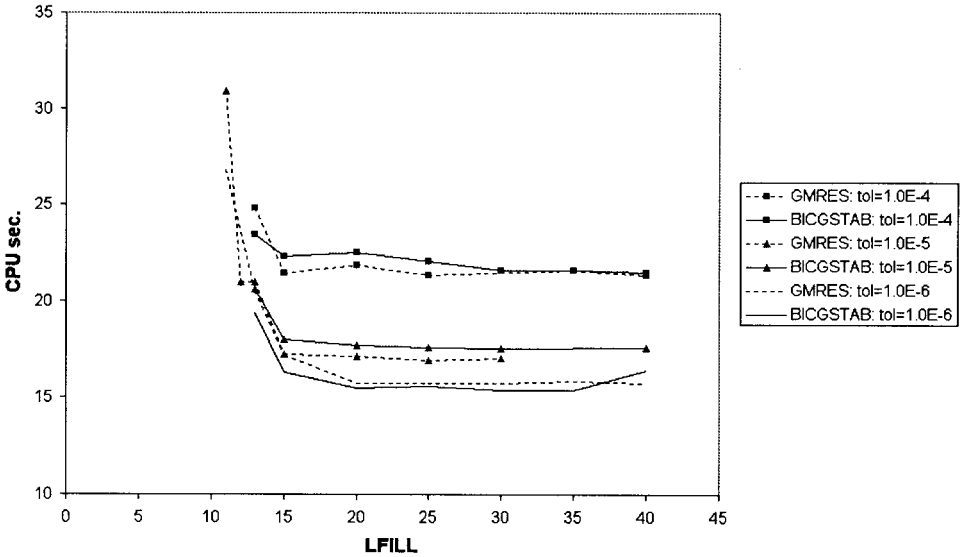

Figure 9. Total used CPU time as function of fill-in for the P-GMRES and Bi-CGSTAB equation solvers are shown for three different threshold values. The grid is 20×6 and 67 time steps have been performed.

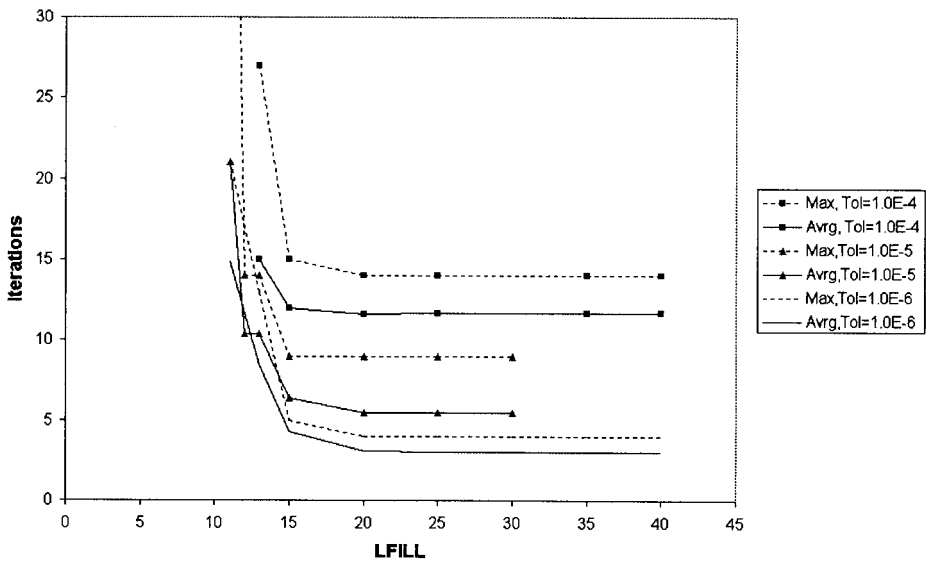

Figure 10. Maximum and average iterations per time step for the GMRES equation solver are shown for three different threshold values. The grid is 20×6 and 67 time steps have been performed.

is taken as an indication for that the error tolerance $1.0\mathrm{E}-7$ for terminating the iterations of the GMRES algorithm was sufficient low, remembering that Gaussian elimination gives the exact solution. A threshold value of $1.0\mathrm{E}-4$ was used.

In Table III used CPU time per time step for solving the pressure–momentum equation system for the different equations solvers are shown. Also for this case it is seen the preconditioned Krylov subspace solvers use very similar CPU time. For the course grid, the

Table III. Broken dam simulations: comparisons of used CPU time for the different equation solvers

Grid	Gauss (CPU s)	GMRES (CPU s)	Bi-CGSTAB (CPU s)	Gauss/GMRES
20×10	152	45	48	3.4
30×15	1343	247	253	5.4
40×20	5300	790	820	6.5

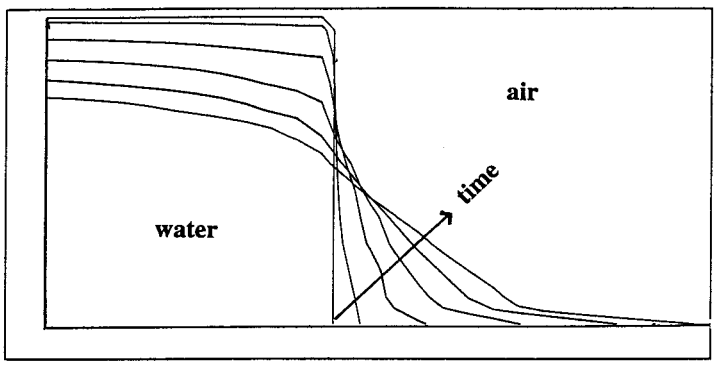

Figure 11. Free surface evolution in the broken dam case. Contour lines for the volumetric fraction 0.5 are shown for six different times. Results of both the Gaussian band solver and the GMRES iterative solver are plotted, but can scarcely be separated.

iterative solvers are about 3.4 times faster than the Gaussian band solver, while for the finest grid the speed-up has increased to about 6.5 times. Note that the grids are more refined for this case than the two previous ones, explaining the improved speed-up compared with the Taylor bubble case. In Figure 12, used CPU time per time step for the preconditioned iterative solvers are shown as a function of fill-in. For the 20×10 grid, a fill-in of 18 was necessary to reach the error tolerance in less than 300 iterations per time step. For the 30×15 and 40×20 grids, the minimum fill-ins were increased to 28 and 40 respectively. It is seen that there are small differences between the two Krylov subspace methods. In Figure 13, the CPU times for the preconditioned GMRES algorithm are split into the parts used by the ILUT preconditioner and the GMRES equation solver. It is seen that initially the preconditioner cost increases while the GMRES cost decreases with increasing fill-in until constant values are reached. The decrease in the cost for the GMRES algorithm with increasing fill-in is caused by a reduced number of iterations, as shown in Figure 14.

4.4. Fill-in of the lower (L) and upper (U) matrices

We have seen that the necessary numbers of fill-in to obtain 'reasonable' convergence rates (less than 300 iterations in one time step) seem to increase with the size of the A matrix. It turns out that the relative fill-in, defined as the number of fill-ins of the L and U matrices divided by number of initially zero coefficients within the band of the lower and upper part of the A matrix, is fairly constant. The bandwidth of the A matrix is equal to $10M - 3$, where M is the number of nodes in the y-direction. From this number, we subtract 9, which is the non-zero coefficients in the band and divide by 2 to obtain the number of zeroes in the lower

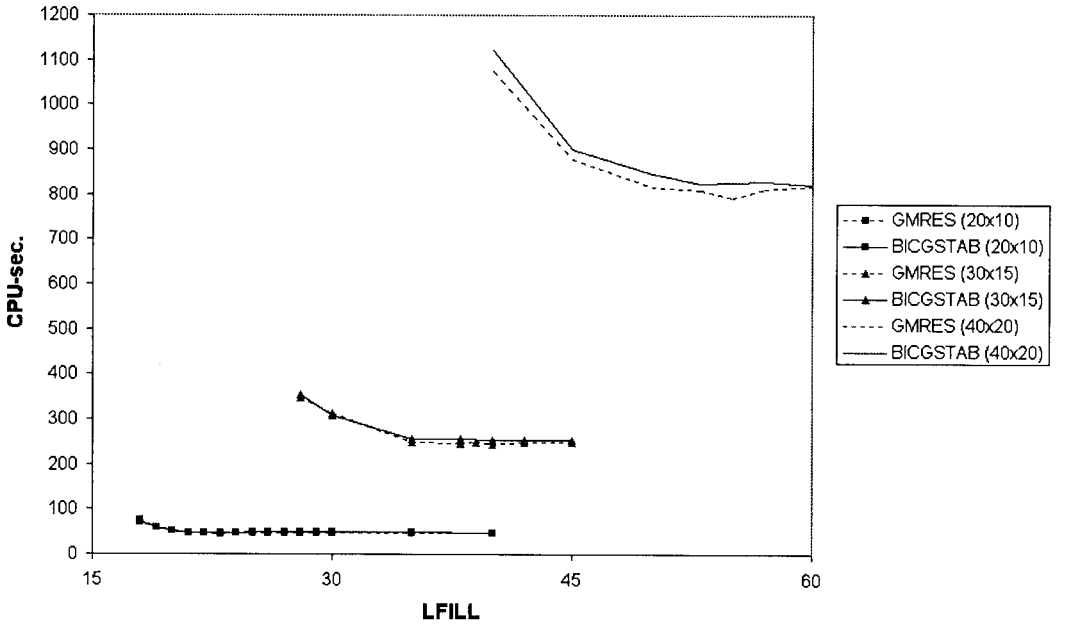

Figure 12. Total used CPU time for the P-GMRES and Bi-CGSTAB algorithms.

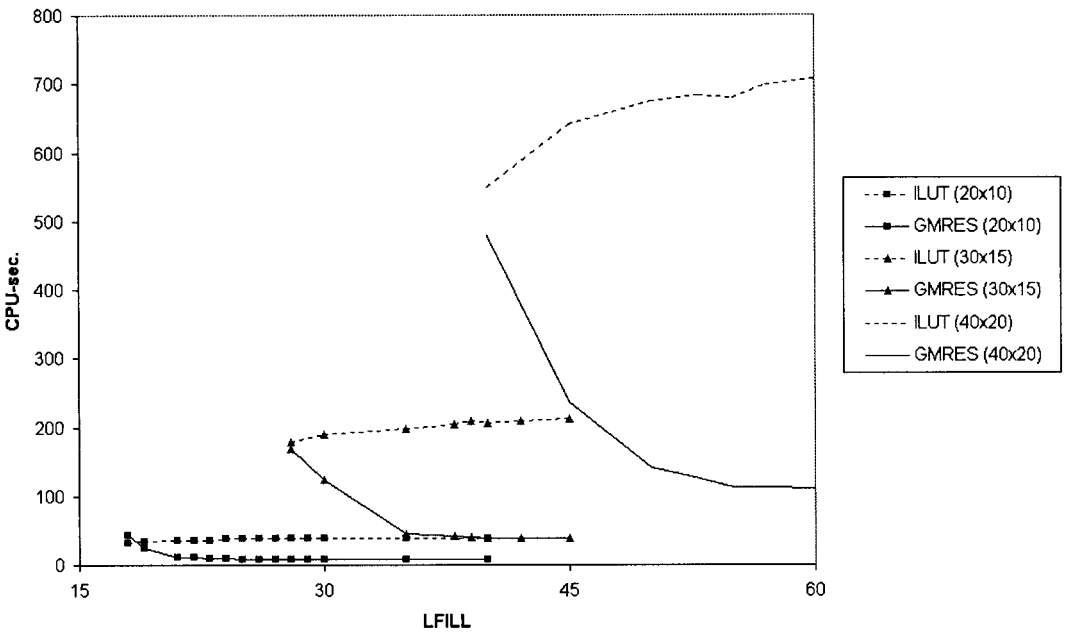

Figure 13. CPU time used as function of fill-in for the ILUT preconditioner and the GMRES equation solver.

and upper parts. In Table IV we have shown the minimum and relative fill-in for the different cases and grid resolutions. TB denotes the Taylor bubble case and BD denotes the broken dam case. It is seen that the relative fill-in is fairly constant with an average of about 43%.

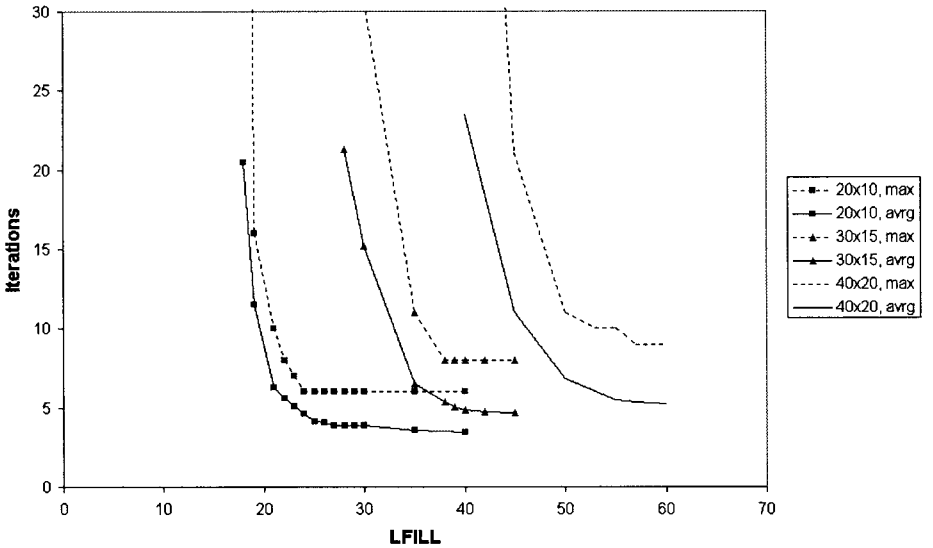

Figure 14. Maximum and average number of iterations per time step using the preconditioned GMRES algorithm.

Table IV. Relative fill-in of the lower (L) and upper (U) matrices

Case	Grid	Minimum fill-in	% Relative fill-in
TB	20×6	11	46
TB	30×9	16	41
BD	20×10	18	41
TB	40×12	23	43
BD	30×15	28	41
BD	40×20	40	43

We have disregarded the single-phase flow case in this analysis since the A matrix is not optimally stored for single-phase flow problems.

5. CONCLUSIONS

Preconditioned GMRES [16] and the Bi-CGSTAB [17] algorithms have been used in solving a two-fluid model developed for simulation of separated and intermittent gas–liquid flows. The preconditioner was the ILUT(p, τ) algorithm [22]. The two-fluid model has one pressure equation and momentum and mass balances for each phase. The numerical method applies a two-step semi-implicit time integration procedure. First, a pressure equation is solved simultaneously with the momentum equations for the two phases, then the mass equations for each phase are solved. For the first step, both the two iterative algorithms, as well as a Gaussian band direct solver, were used, while only the Gaussian solver was applied for the second, much faster, step. There were only small differences between the performance of the preconditioned GMRES and the Bi-CGSTAB algorithms. The computational time using these iterative methods was from 2 to 6 times shorter than when using the Gaussian band solver. The speed-up in computational time increased with the number of unknowns. Only relatively small

cases up to 800 nodes were tested, since the computational time using the Gaussian solver then started to become large. It was seen that the preconditioner was more time-demanding than the iterative solvers. The fill-in parameter for the preconditioner increased with the number of unknowns. With more unknowns, the bandwidth of the matrix increased as well and we observed that the fill-in was about 43% of the bandwidth. This is not completely satisfactory and since the iterative solvers GMRES and Bi-CGSTAB are known to be two of the most robust Krylov subspace methods, improvements should probably be sought in the preconditioning step. The final conclusion is that the efficiency of the presented robust numerical method has been improved considerably, but that there is possibly room for more improvements on the preconditioning of the equation system.

ACKNOWLEDGMENTS

The authors are grateful for the financial support from the Strategic Institute Program 'Fundamental Studies in Multiphase Flow' under the Norwegian Research Council.

REFERENCES

1. K. Bendiksen, D. Malnes, R. Moe and S. Nuland, 'The dynamic two-fluid model OLGA: theory and applications', *SPE Production Engineers*, 1991, pp. 171–180.
2. M.T. Philbin, 'The simulation of transient phenomena in multiphase production systems', *IBC Multiphase Operations Offshore Conference*, 1991, pp. 1–29.
3. G. Ferschneider, M. Lagiere, T. Bourgeois and M. Fitreman, 'How to calculate two-phase flow of gas and oil in pipe lines', *Pipe Line Industry*, **63**, 33–39 (1985).
4. Fluent Incorporated, *FLUENT Users Guide*, Version 4.2, Fluent Incorporated, Lebanon, USA, 1993.
5. H.I. Roston and D.B. Spalding, The PHOENICS Reference Manual, *CHAM TR/200*, 1987.
6. *FIDAP Users Manual*, Fluid Dynamic International, Inc., 1993.
7. D.B. Spalding, 'The calculation of free-convection phenomena in gas–liquid mixtures', Imperial College Mechanical Engineering Department, *Report No. HTS/76/11*, 1976.
8. R.I. Issa and A.D. Gosman, 'The computation of three-dimensional turbulent two-phase flows in mixture vessels', *Proceedings of the 2nd International Conference on Numerical Methods in Laminar Turbulent Flow*, Pineridge Press, Swansea, 1981, pp. 827–839.
9. M.K. Looney, R.I. Issa, A.D. Gosman and S. Politis, 'Modelling of the turbulent flow of solid/liquid suspensions in stirred vessels', *Proc. 5th Int. Conf. on Mathematical Modelling*, Berkeley, CA, 1985.
10. R. Moe and K.H. Bendiksen, 'Transient simulation of 2–3D stratified and intermittent two-phase flow. Part I: Theory', *Int. J. Numer. Methods Fluids*, **16**, 461–487 (1993).
11. P.H. Vernier and J.M. Delhaye, 'General two-phase flow equations applied to the thermodynamics of boiling water nuclear reactors', *Energie Primaire*, **4**, 5–46 (1968).
12. M. Ishii, *Thermo-fluid Dynamic Theory of Two-phase Flow*, Eyrolles, Paris, 1975, pp. 45–115.
13. R. Moe, 'Transient simulation of 2–3D stratified and intermittent two-phase flows. Part II: Applications', *Int. J. Numer. Methods Fluids*, **16**, 967–988 (1993).
14. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
15. M.R. Hestenes and E. Steifel, 'Methods of conjugate gradients for solving linear systems', *J. Res. Nat. Bur. Stand.*, **49**, 409–436 (1952).
16. Y. Saad and M.H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
17. H.A. Van der Vorst, 'Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **13**, 631–644 (1992).
18. T. Rusten and R. Winther, 'A preconditioned iterative method for saddlepoint problems', *SIAM J. Matrix Anal. Appl.*, **13**, 887–904 (1992).
19. H. Elman and D. Sylvester, 'Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations', *Comput. Sci. Tech. Rep. Ser.*, **3283** (1994).
20. K. Ajmani and W.F. Ng, 'Preconditioned conjugate gradient methods for the Navier–Stokes equations', *J. Comput. Phys.*, **110**, 68–81 (1994).
21. M.C. Liang and C.W. Lan, 'A finite volume/Newton method for a two-phase heat flow problem using primitive variables and collocated grids', *J. Comput. Phys.*, **127**, 330–345 (1996).
22. Y. Saad, 'ILUT, a dual threshold incomplete ILU factorization', *Numer. Linear Algebra Appl.*, **1**, 387–402 (1994).

23. Y. Saad, 'SPARSKIT, a basic tool-kit for sparse matrix computations', http://www.cs.umn.edu/Research/arpa/SPARSKIT/sparskit.html.
24. S. Banerjee and A.M.C. Chan, 'Separated flow models—I. Analysis of the averaged and local instantaneous formulations', *Int. J. Multiphase Flow*, **6**, 1–24 (1980).
25. B.E. Launder and D.B. Spalding, *Mathematical Models of Turbulence*, Academic Press, London, 1972.

