Acceleration of Preconditioned Krylov Solvers for Bubbly Flow Problems

Jok Man Tang and Kees Vuik

Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft Institute of Applied Mathematics,
Mekelweg 4, 2628 CD Delft, The Netherlands
j.m.tang, c.vuik@tudelft.nl

Abstract. We consider the linear system arising from discretization of the pressure Poisson equation with Neumann boundary conditions, derived from bubbly flow problems. In the literature, preconditioned Krylov iterative solvers are proposed, but they often suffer from slow convergence for relatively large and complex problems. We extend these traditional solvers with the so-called deflation technique, that accelerates the convergence substantially and has favorable parallel properties. Several numerical aspects are considered, such as the singularity of the coefficient matrix and the varying density field at each time step. We demonstrate theoretically that the resulting deflation method accelerates the convergence of the iterative process. Thereafter, this is also demonstrated numerically for 3-D bubbly flow applications, both with respect to the number of iterations and the computing time.

Keywords: deflation, conjugate gradient method, preconditioning, symmetric positive semi-definite matrices, bubbly flow problems.

1 Introduction

Recently, moving boundary problems have received much attention in literature, due to their applicative relevance in many physical processes. One of the most popular moving boundary problems is modelling bubbly flows, see e.g. [12]. These bubbly flows can be simulated, by solving the well-known Navier-Stokes equations for incompressible flow:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u \cdot \nabla u + \dfrac{1}{\rho}\nabla p = \dfrac{1}{\rho}\nabla \cdot \mu \left(\nabla u + \nabla u^T\right) + g; \\ \nabla \cdot u = 0, \end{cases} \tag{1}$$

where g represents the gravity and surface tension force, and ρ, p, μ are the density, pressure and viscosity, respectively. Eqs. (1) can be solved using, for instance, the pressure correction method [7]. The most time-consuming part of this method is solving the symmetric and positive semi-definite (SPSD) linear system on each time step, which comes from a second-order finite-difference

R. Wyrzykowski et al. (Eds.): PPAM 2007, LNCS 4967, pp. 1323–1332, 2008.

discretization of the Poisson equation with possibly discontinuous coefficients and Neumann boundary conditions:

$$\begin{cases} \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = f_1, & \mathbf{x} \in \Omega, \\ \frac{\partial}{\partial \mathbf{n}} p = f_2, & \mathbf{x} \in \partial\Omega, \end{cases} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{n}$ denote the spatial coordinates and the unit normal vector to the boundary $\partial\Omega$, respectively. In the 3-D case, domain Ω is chosen to be a unit cube. Furthermore, we consider two-phase bubbly flows, so that ρ is piecewise constant with a relatively high contrast:

$$\rho = \begin{cases} \rho_0 = 1, & \mathbf{x} \in \Lambda_0, \\ \rho_1 = 10^{-3}, & \mathbf{x} \in \Lambda_1, \end{cases} \tag{3}$$

where Λ_0 is water, the main fluid of the flow around the air bubbles, and Λ_1 is the region inside the bubbles.

The resulting linear system which has to be solved is

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \tag{4}$$

where the singular coefficient matrix A is SPSD and $b \in \text{range}(A)$. In practice, the preconditioned Conjugate Gradient (CG) method [4] is widely used to solve (4), see also References [1, 2, 3, 5].

In this paper, we will restrict ourselves to the Incomplete Cholesky (IC) decomposition [8] as preconditioner, and the resulting method will be denoted as ICCG. In this method,

$$M^{-1}Ax = M^{-1}b, \quad M \text{ is the IC preconditioner},$$

is solved using CG. ICCG shows good performance for relatively small and easy problems. For complex bubbly flows or for problems with large jumps in the density, this method shows slow convergence, due to the presence of small eigenvalues in the spectrum of $M^{-1}A$, see also [13]. This phenomenon also holds if we use other preconditioners instead of the IC preconditioner.

To remedy the bad convergence of ICCG, the deflation technique has been proposed, originally from Nicolaides [11]. The idea of deflation is to project the extremely small eigenvalues of $M^{-1}A$ to zero. This leads to a faster convergence of the iterative process, due to the fact that CG can handle matrices with zero-eigenvalues [6] and the effective condition number becomes more favorable. The resulting method is called Deflated ICCG or shortly DICCG, following [19], and it will be further explained in the next section.

DICCG is a typical two-level Krylov projection method, where a combination of traditional and projection-type preconditioners is used to get rid of the effect of both small and large eigenvalues of the coefficient matrix. In the literature, there are more related projection methods known, coming from the fields of domain decomposition (such as balancing Neumann-Neumann) and multigrid (such as additive coarse-grid correction). At first glance, these methods seem to be different. However, from an abstract point of view, it can be shown that they are closely related to each other and some of them are even equivalent. We refer to [18] for a theoretical and numerical comparison of these methods.

2 Deflation Method

In DICCG, we solve

$$M^{-1}PA\tilde{x} = M^{-1}Pb, \quad P \text{ is the deflation matrix,}$$

using CG, where

$$P := I - AZE^{-1}Z^T, \quad E := Z^TAZ, \quad Z \in \mathbb{R}^{n \times r}, \quad r \ll n. \tag{5}$$

Piecewise-constant deflation vectors are used to approximate the eigenmodes corresponding to the components which caused the slow convergence of ICCG. More technically, deflation subspace matrix $Z = [z_1 \; z_2 \; \cdots \; z_r]$ consists of deflation vectors, z_j, with

$$z_j(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Omega \setminus \bar{\Omega}_j; \\ 1, & \mathbf{x} \in \Omega_j, \end{cases}$$

where the domain Ω is divided into nonoverlapping subdomains Ω_j, which are chosen to be cubes, assuming that the number of grid points in each spatial direction is the same. This approach is strongly related to methods known in domain decomposition. Note that, due to the construction of the sparse matrix Z, matrices AZ and E are sparse as well, so that the extra computations with the deflation matrix, P, are relatively cheap.

Moreover, since the piecewise-constant deflation vectors correspond to nonoverlapping subdomains, the deflation technique has excellent parallel properties. This is in contrast to the IC preconditioner. Therefore, for parallel computations, one should combine the deflation technique with for instance the block-IC preconditioner in the CG method. For more details, one can consult [20].

3 Application to Bubbly Flows

The deflation technique works well for invertible systems and when the deflation vectors are based on the geometry of the problem, see also References [9, 10]. However, in our bubbly flow applications, we have systems with singular matrices and deflation vectors that are chosen independently of the geometry of the density field. Hence, main questions in this paper are:

– is the deflation method also applicable to linear systems with singular matrices?
– is the deflation method with fixed deflation vectors also applicable to problems, where the position and radius of the bubbles change in every time step?

The answers will be given in this section.

3.1 Deflation and Singular Matrices

First, we show that DICCG can be used for singular matrices. Due to the construction of matrix Z and the singularity of A, the coarse matrix $E := Z^T AZ$ is also singular. In this case, E^{-1} does not exist. We propose several new variants of deflation matrices P:

(i) invertibility of A is forced resulting in a deflation matrix P_1, i.e., we adapt the last element of A such that the new matrix, denoted as $\widetilde{A}$, is invertible;
(ii) a column of Z is deleted resulting in a deflation matrix P_2, i.e., instead of Z we take $[z_1 \; z_2 \; \cdots \; z_{r-1}]$ as the deflation subspace matrix;
(iii) systems with a singular E are solved iteratively resulting in a deflation matrix P_3, i.e., matrix E^{-1} as given in Eq. (5) is considered to be a pseudo-inverse.

As a result, Variant (i) and (ii) give a nonsingular matrix E, whereas the real inverse of E is not required anymore in Variant (iii). Moreover, note that Variant (iii) is basically identical to the original DICCG for invertible systems, see e.g. [9,19], since the original coefficient matrix and all r deflation vectors are used in this variant.

Subsequently, we can prove that the three DICCG variants are identical in exact arithmetic, see Theorem 1.

Theorem 1. $P_1\widetilde{A} = P_2A = P_3A.$

Proof. The proof can be found in [15,16].

We observe that the deflated systems of all variants are identical. From this result, it is easy to show that the preconditioned deflated systems are also the same, that is

$$M^{-1}P_1\widetilde{A} = M^{-1}P_2A = M^{-1}P_3A.$$

Since the variants are equal, any of them can be chosen in the numerical experiments. In the next section, we will apply the first variant for convenience, and the results and efficiency of this variant will be demonstrated numerically.

3.2 Deflation and Varying Density Fields

The number of smallest eigenvalues of $M^{-1}A$, that are of order 10^{-3}, are related to the number of bubbles in Ω, see [17, Sect. 3]. To show that DICCG works in cases with varying density fields, we have to show that the deflation vectors approximate the eigenvectors corresponding to the smallest eigenvalues. Only in this case, the deflation technique eliminates those eigenvalues that causes the slow convergence of ICCG. Proposition 1 can be found in [17]:

Proposition 1. *Eigenvectors v_i of $M^{-1}A$ corresponding to small eigenvalues λ_i associated with bubbles remain good approximations if*

– *one or more elements of v_i corresponding to Λ_0 are perturbed arbitrarily;*

– *elements of v_i corresponding to a whole bubble of Λ_1 are perturbed with a constant.*

Proposition 1 implies that the column space of Z can indeed approximate v_i as long as each subdomain contains at most (a part of) one bubble. Therefore, as long as r is chosen sufficiently large, the deflation method works appropriately, since the spectrum of $M^{-1}PA$ does not consist of eigenvalues of $\mathcal{O}(10^{-3})$. In addition, it appears that the subdomain deflation vectors even approximate other small eigenvalues of $\mathcal{O}(1)$, because the associated eigenvectors are slow-varying modes that allow small perturbations. For more details, we refer to [17]. We conclude that DICCG can be effective using subdomain deflation vectors, where Z can be constructed independently of the geometry of the density field.

In the next section, numerical experiments will be presented to show the success of the method for time-dependent bubbly flow problems.

4 Numerical Experiments

We test the efficiency of the DICCG method for two kinds of test problems.

4.1 Test Case 1: Stationary Problem

First, we take a 3-D bubbly flow application with eight air-bubbles in a domain of water, see Figure 1 for the geometry. We apply finite differences on a uniform Cartesian grid with $n = 100^3$, resulting in a very large but sparse linear system $Ax = b$ with SPSD matrix A.

Then, the results of ICCG and DICCG can be found in Table 1, where ϕ denotes the final relative exact residual and DICCG$-r$ denotes DICCG with r deflation vectors. Moreover, we terminate the iterative process, when the relative update residuals are smaller than the stopping tolerance, $\epsilon = 10^{-8}$.

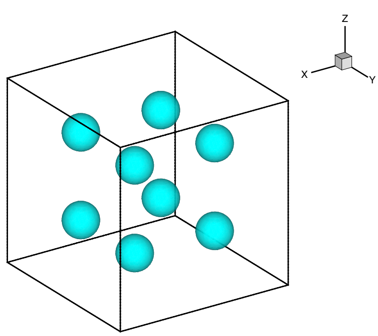


Fig. 1. An example of a bubbly flow problem: eight air-bubbles in a unit domain filled with water

Table 1. Convergence results of ICCG and DICCG$-r$ solving $Ax = b$ with $n = 100^3$, for the test problem as given in Figure 1

Method	# Iterations	CPU Time (s)	ϕ ($\times 10^{-9}$)
ICCG	291	43.0	1.1
DICCG-2^3	160	29.1	1.1
DICCG-5^3	72	14.2	1.2
DICCG-10^3	**36**	**8.2**	**0.7**
DICCG-20^3	22	27.2	0.9

From Table 1, one observes that the larger the number of deflation vectors, the less iterations DICCG requires. With respect to the CPU time, there is an optimum, namely for $r = 10^3$. Hence, in the optimal case, DICCG is more than five times faster compared to the original ICCG method, while the accuracy of both methods are comparable!

Similar results also hold for other related test cases. Results of ICCG and DICCG for the problem with 27 bubbles can be found in Table 2. In addition, it appears that the benefit of the deflation method is larger when we increase the number of grid points, n, in the test cases, see also [16].

Table 2. Convergence results of ICCG and DICCG$-r$ solving $Ax = b$ with $n = 100^3$, for the test case with 27 bubbles

Method	# Iterations	CPU Time (sec)	ϕ ($\times 10^{-9}$)
ICCG	310	46.0	1.3
DICCG-2^3	275	50.4	1.3
DICCG-5^3	97	19.0	1.2
DICCG-10^3	**60**	**13.0**	**1.2**
DICCG-20^3	31	29.3	1.2

Finally, for the test case with 27 bubbles, the plots of the residuals during the iterative process of both ICCG and DICCG can be found in Figure 2. Notice that the behavior of the residuals of ICCG are somewhat irregular due to the presence of the bubbles. For DICCG, we conclude that the larger r, the more linear the residual plot is, so the faster the convergence of the iterative process. Apparently, the eigenvectors associated to the small eigenvalues of $M^{-1}A$ have been well-approximated by the deflation vectors and $M^{-1}PA$ is better conditioned, if r is sufficiently large.

4.2 Test Case 2: Time-Dependent Problem

Next, we present some results for the 3-D simulation of a rising air bubble in water, in order to show that the deflation method is also applicable to real-life problems with varying density fields. We adopt the pressure-correction method

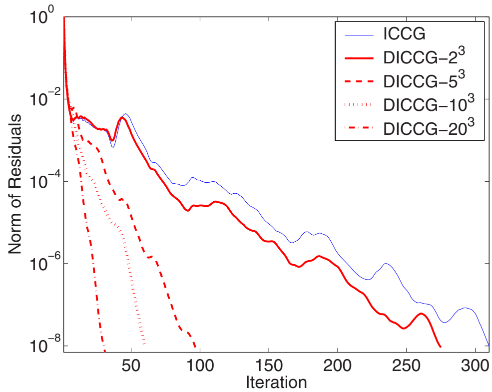


Fig. 2. Residual plots of ICCG and DICCG−r, for the test problem with 27 bubbles and various number of deflation vectors r

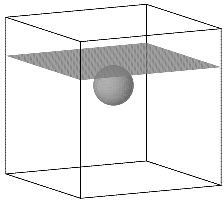
for the simulations, but it could be replaced by any operator-splitting method, in general. For more details, see [13]. At each time step, a pressure Poisson equation has to be solved, which is the most time-consuming part of the whole simulation. Therefore, during this section we only concentrate on this part at each time step. We investigate whether DICCG is efficient for all those time steps.

We consider a test problem with a rising air bubble in water without surface tension. The exact material constants and other relevant information can be found in [13, Sect. 8.3.2]. The starting position of the bubble in the domain and the evolution of the movement during the 250 time steps are given in Figure 3.

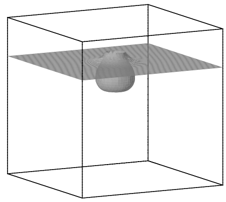
In [13], the Poisson solver is based on ICCG. Here, we will compare this method to DICCG with $r = 10^3$ deflation vectors, in the case of $n = 100^3$. The results are presented in Figure 4.

From Subfigure 4(a), we observe that the number of iterations is strongly reduced by the deflation method. DICCG requires approximately 60 iterations, while ICCG converges between 200 and 300 iterations at most time steps. Moreover, we observe the erratic behavior of ICCG, whereas DICCG seems to be less sensitive to the geometries during the evolution of the simulation. Also with respect of the CPU time, DICCG shows very good performance, see Subfigure 4(b). At most time steps, ICCG requires 25–45 seconds to converge, whereas DICCG only needs around 11–14 seconds. Moreover, in Figure 4(c), one can find the gain factors, considering both the ratios of the iterations and the CPU time between ICCG and DICCG. From this figure, it can be seen that DICCG needs approximately 4–8 times less iterations, depending on the time step. More importantly, DICCG converges approximately 2–4 times faster to the solution compared to ICCG, at all time steps.

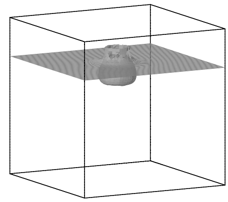
In general, we see that, compared to ICCG, DICCG decreases significantly the number of iterations and the computational time as well, which are required for solving the pressure Poisson equation with discontinuous coefficients, in applications of 3-D bubbly flows.



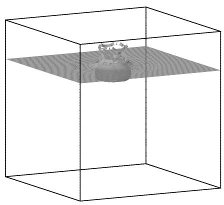
(a) $t = 0$.



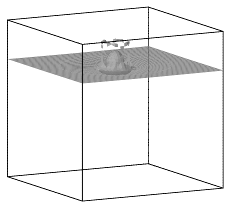
(b) $t = 50$.



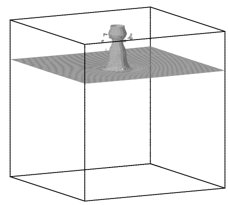
(c) $t = 100$.



(d) $t = 150$.

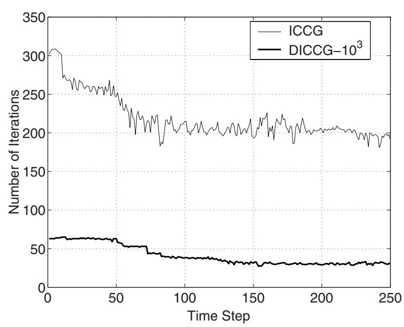


(e) $t = 200$.

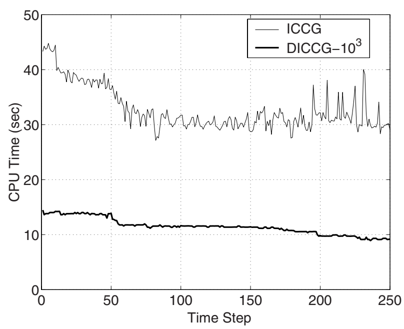


(f) $t = 250$.

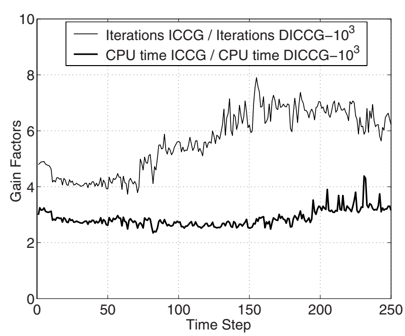
Fig. 3. Evolution of the rising bubble in water without surface tension in the first 250 time steps



(a) Number of iterations versus time step.



(b) CPU time versus time step.



(c) Gain factors with respect to ICCG and DICCG.

Fig. 4. Results of ICCG and DICCG with $r = 10^3$, for the simulation with a rising air bubble in water

5 Conclusions

A deflation technique has been proposed to accelerate the convergence of standard preconditioned Krylov methods, for solving bubbly flow problems. In the literature, this deflation method has already been proven to be efficient, for linear systems with invertible coefficient matrix and not-varying density fields in time. However, in our bubbly flow applications, we deal with linear systems with a singular matrix and varying density fields. In this paper, we have shown, both theoretically and numerically, that the deflation method with fixed subdomain deflation vectors can also be applied to this kind of problems. The method appears to be robust and very efficient in various numerical experiments, with respect to both the number of iterations and the computational time.

References

1. Benzi, M.: Preconditioning techniques for large linear systems: A survey. J. Comp. Phys. 182, 418–477 (2002)
2. Gravvanis, G.A.: Explicit Approximate Inverse Preconditioning Techniques. Arch. Comput. Meth. Eng. 9, 371–402 (2002)
3. Grote, M.J., Huckle, T.: Parallel preconditioning with sparse approximate inverses. SIAM J. Sci. Comput. 18, 838–853 (1997)
4. Hestenes, M.R., Stiefel, E.: Methods of Conjugate Gradients for Solving Linear Systems. J. Res. Nat. Bur. Stand. 49, 409–436 (1952)
5. Huckle, T.: Approximate sparsity patterns for the inverse of a matrix and preconditioning. Appl. Num. Math. 30, 291–303 (1999)
6. Kaasschieter, E.F.: Preconditioned Conjugate Gradients for solving singular systems. J. Comp. Appl. Math. 24, 265–275 (1988)
7. van Kan, J.J.I.M.: A second-order accurate pressure correction method for viscous incompressible flow. SIAM J. Sci. Stat. Comp. 7, 870–891 (1986)
8. Meijerink, J.A., van der Vorst, H.A.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. Comp. 31, 148–162 (1977)
9. Nabben, R., Vuik, C.: A comparison of Deflation and Coarse Grid Correction applied to porous media flow. SIAM J. Numer. Anal. 42, 1631–1647 (2004)
10. Nabben, R., Vuik, C.: A Comparison of Deflation and the Balancing Preconditioner. SIAM J. Sci. Comput. 27, 1742–1759 (2006)
11. Nicolaides, R.A.: Deflation of Conjugate Gradients with applications to boundary value problems. SIAM J. Matrix Anal. Appl. 24, 355–365 (1987)
12. Van der Pijl, S.P., Segal, A., Vuik, C., Wesseling, P.: A mass-conserving Level-Set method for modelling of multi-phase flows. Int. J. Num. Meth. in Fluids 47(4), 339–361 (2005)
13. Van der Pijl, S.P.: Computation of bubbly flows with a mass-conserving level-set method, PhD thesis, Delft University of Technology, Delft (2005)
14. Sousa, F.S., Mangiavacchi, N., Nonato, L.G., Castelo, A., Tome, M.F., Ferreira, V.G., Cuminato, J.A., McKee, S.: A Front-Tracking / Front-Capturing Method for the Simulation of 3D Multi-Fluid Flows with Free Surfaces. J. Comp. Physics 198, 469–499 (2004)
15. Tang, J.M., Vuik, C.: On Deflation and Singular Symmetric Positive Semi-Definite Matrices. J. Comp. Appl. Math. 206, 603–614 (2007)

16. Tang, J.M., Vuik, C.: Efficient Deflation Methods applied to 3-D Bubbly Flow Problems. Elec. Trans. Numer. Anal. 26, 330–349 (2007)
17. Tang, J.M., Vuik, C.: New Variants of Deflation Techniques for Bubbly Flow Problems. J. Numer. Anal. Indust. Appl. Math. 2, 227–249 (2007)
18. Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A.: Theoretical and numerical comparison of various projection methods derived from deflation, domain decomposition and multigrid methods, Delft University of Technology, Department of Applied Mathematical Analysis, Report 07-04, ISSN 1389-6520 (2007)
19. Vuik, C., Segal, A., Meijerink, J.A.: An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. J. Comp. Phys. 152, 385–403 (1999)
20. Frank, J., Vuik, C.: On the construction of deflation-based preconditioners. SIAM J. Sci. Comp. 23, 442–462 (2001)