International Journal for Multiscale Computational Engineering, 6(1)13–24(2008)

Fast Deflation Methods with Applications to Two-Phase Flows

J. M. Tang & C. Vuik

Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, J. M. Burgerscentrum, Mekelweg 4, 2628 CD Delft, The Netherlands

ABSTRACT

Traditional Krylov iterative solvers, such as the preconditioned conjugate gradient method, can be accelerated by incorporating a second-level preconditioner. We use deflation as a second-level preconditioner, which is very efficient in many applications. In this paper, we give some theoretical results for the general deflation method applied to singular matrices, which provides us more insight into the properties and the behavior of the method. Moreover, we discuss stability issues of the deflation method and consider some ideas for a more stable method. In the numerical experiments, we apply the deflation method and its stabilized variant to singular linear systems derived from two-phase bubbly flow problems. Because of the appearance of bubbles, those linear systems are ill-conditioned, and therefore, they are usually hard to solve using traditional preconditioned Krylov iterative methods. We show that our deflation methods can be very efficient to solve the linear systems. Finally, we also investigate numerically the stability of these methods by examining the corresponding inner-outer iterations in more detail.

KEYWORDS

deflation, inner-outer iterations, bubbly flows, poisson equation, discontinuous coefficients

*Address all correspondence to J.M.Tang@tudelft.nl

1. INTRODUCTION

Computation of two-phase flows, and in particular bubbly flows, is currently a very active research topic in computational fluid dynamics (CFD); see, for instance, [1–6]. Understanding the dynamics and interaction of bubbles and droplets in a large variety of industrial processes is crucial for economically and ecologically optimized design. Two-phase flows are complicated to simulate because the geometry of the problem typically varies with time and the fluids involved can have very different material properties. A simple example is that of air bubbles in water, where the densities vary by a factor of about 1000.

Mathematically, bubbly flows are governed by the Navier-Stokes equations, which can be solved numerically using an operator-splitting method, where the velocity and pressure are solved sequentially for each time step. In many of the popular operator-splitting methods, a linear system for the pressure correction has to be solved for every time step, which is a discretized version of a Poisson equation with discontinuous coefficients. This consumes the bulk of the computing time, although the coefficient matrix of the linear system is elliptic. This is caused by the occurrence of strongly varying coefficients, arising from large jumps of the density. The appearance of bubbles in the domain has a negative influence on the condition of the coefficient matrix.

The pressure correction equation can be solved in different ways. A Krylov iterative solver can be adopted, such as the preconditioned conjugate gradient (PCG) method. However, PCG with traditional preconditioners such as Jacobi, incomplete Cholesky, and geometric multigrid preconditioners often suffers from slow convergence because the coefficient matrix remains ill-conditioned after preconditioning. As an alternative for PCG, a deflated variant of PCG, called DPCG, can be applied (see, e.g., [7,8]). The incorporated deflation technique removes the components of the eigenmodes that cause the slow convergence of PCG. In many applications, DPCG has been proven to be an efficient method, such as in porous media flows [9] and ground water flows [10]. Recently, DPCG has also been applied successfully to bubbly flows in [11,12].

The DPCG method can be seen as a two-grid method because an additional coarse system has to be solved in each iteration. If these coarse systems are solved recursively, then we would obtain a method that is very close to multigrid methods [13,14]. In this paper, we do not consider this class of methods; we restrict ourselves to coarse systems that are solved either in a direct or iterative way.

Although the DPCG method can be applied effectively to various applications, there is a lack of theory in the literature. For nonsingular linear systems, some analysis can be found in [10,15,16], but for singular linear systems, theoretical results are lacking. Only for special cases, the deflation method has been proven to be efficient (see, e.g., [11,17]). The aim of this paper is to investigate theoretical properties of the general DPCG method applied to singular systems, including some insight into the stability and the difficulties in two-grid iteration processes. This will provide us a better understanding of the efficiency of the deflation method. Moreover, to illustrate the theoretical results, we present some numerical experiments, where the DPCG method is applied to ill-conditioned linear systems derived from two-phase bubbly flows.

In the two-phase flow applications we consider, the multiphysics and multiscale aspects play an important role. The linear systems are hard to solve because (small) bubbles or droplets of one phase move in a background of the other phase and relatively high jumps appear in the density field.

This paper is organized as follows. We give the deflation method in Section 2. Subsequently, the theoretical properties and results considering the deflation method are presented in Section 3. Section 4 is devoted to the description of the bubbly flow problem. In Section 5, the numerical experiments considering the bubbly flows are carried out. Then, conclusions are drawn.

2. DEFLATION METHOD

Suppose that the following SPSD linear system has to be solved:

$$Ax = b, \quad A \in \mathbb{R}^{n \times n} \tag{1}$$

where A is singular and $b \in \text{Col}(A)$ holds. A popular iterative method to solve (1) is PCG. In this method, the resulting linear system to be solved is

$$M^{-1}Ax = M^{-1}b$$

where M denotes an SPD preconditioner. As mentioned in the introduction, the spectrum of $M^{-1}A$, denoted by $\sigma\left(M^{-1}A\right)$, often consists of small eigenvalues, implying relatively slow convergence of PCG. To accelerate the convergence of the iterative process, the deflation method can be used, which is described below.

We define the deflation matrix P as

$$P := I - AQ, \quad Q = ZE^{+}Z^{T}, \quad E := Z^{T}AZ \quad (2)$$

In Eq. (2), $Z \in \mathbb{R}^{n\times k}$ with $k \ll n$ is the so-called deflation subspace matrix whose k columns are the deflation vectors, which remain unspecified for the moment. In this paper, the following assumption is always fullfilled.

Assumption 1. *Z consists of linearly independent columns, i.e., rank $Z = k$.*

It depends on the choice of Z whether or not the coarse matrix $E \in \mathbb{R}^{k\times k}$ is singular. If E is singular, then the pseudoinverse is used, denoted as E^{+} in (2). Of course, if E is nonsingular, E^{+} is equal to E^{-1}.

Now, the following linear system has to be solved in the resulting DPCG method:

$$M^{-1}PA\tilde{x} = M^{-1}Pb \quad (3)$$

Note that PA is again SPSD. To find solution x, it can be proven that $P^{T}\tilde{x} = P^{T}x$. This implies $x = Qb + P^{T}\tilde{x}$ (see, e.g., [8,10] for more details).

3. THEORETICAL PROPERTIES

In this section, we present some theoretical results for the DPCG method. In contrast to the case with a singular system (1), a lot is known about deflation applied to a linear system with an invertible coefficient matrix (see, e.g., [10,16]). We aim at generalizing some theoretical results given in [10,16] so that they can be applied to general singular matrices as well. Note that a few results given in this section have also been proven in [11], but that is only valid for special choices of Z. We start the analysis for a nonsingular matrix E, and thereafter, the case with a singular E is considered. Subsequently, we analyze the coarse matrix E in more detail and consider the stability properties of DPCG.

Throughout this section, we denote by $\lambda_i(C)$ the ith eigenvalue of a symmetric matrix C, where the sequence of $\lambda_i(C)$ is ordered increasingly. The condition number is given by $\kappa(C) := \lambda_{\max}(C)/\lambda_{\min}(C)$, where $\lambda_{\max}(C)$ and $\lambda_{\min}(C)$ are the largest and smallest eigenvalue of C, respectively. Moreover, the effective condition number is denoted by $\kappa_{\text{eff}}(C) := \lambda_{\max}(C)/\bar{\lambda}_{\min}(C)$, where $\bar{\lambda}_{\min}(C)$ is the smallest *nonzero* eigenvalue of C. Finally, we define $\mathbf{1}_p$ and $\mathbf{0}_p$ as the all-one and all-zero vector with p elements, respectively.

3.1 Deflation Properties with Nonsingular Coarse Matrix

The following assumption holds throughout this subsection.

Assumption 2. *Suppose that $A \in \mathbb{R}^{n\times n}$ is a singular SPSD matrix and $M \in \mathbb{R}^{n\times n}$ is a nonsingular SPD matrix. Z is chosen in such a way that E is an invertible SPD matrix.*

Note that, from Assumption 2, it does not necessarily follow that AZ has full rank, which is in contrast to the case that A is invertible. Next, we will use the following notation throughout this subsection:

$$\begin{aligned} P_i &:= I - AQ_i, \quad Q_i := Z_i E_i^{-1} Z_i^{T} \\ E_i &:= Z_i^{T} A Z_i, \quad Z_i \in \mathbb{R}^{n\times k_i} \end{aligned}$$

for $i = 1, 2$, where Assumption 2 holds. The following results can now be proven.

Theorem 1. *Let $k_1 = k_2 = k$. If Col(Z_1) = Col(Z_2), then $M^{-1}P_1A = M^{-1}P_2A$, and in particular, $Q_1 = Q_2$.*

Proof. The proof is identical to the proof of [10] (Lemma 2.9).

Theorem 2. *Suppose that $M^{-1}P_1A$ has $k_1 + 1$ zero-eigenvalues and $M^{-1}P_2A$ has $k_2 + 1$ zero-eigenvalues. If Col(Z_1) $\subseteq$ Col(Z_2), then*

$$\kappa_{\text{eff}}(M^{-1}P_2A) \le \kappa_{\text{eff}}(M^{-1}P_1A) \quad (4)$$

Proof. It is sufficient to show that $\kappa_{\text{eff}}(P_1A) \le \kappa_{\text{eff}}(P_2A)$ because the proof of (4) is analogous to it. In other words, we have to show that

$$\begin{aligned} \lambda_n(P_1A) &\geq \lambda_n(P_2A) \\ \lambda_{k_1+2}(P_1A) &\leq \lambda_{k_2+2}(P_2A) \end{aligned} \tag{5}$$

Note first that $(P_1 - P_2)A$ is PSD, which can be easily proved, by applying the same procedure as in the proof of [9] (Lemma 2.8). By combining this fact with [10] (Lemma 2.2), the first inequality of (5) can be obtained. The proof of the second inequality of (5) is exactly the same as for the case that A is invertible, see the proof of [10] (Theorem 2.10).

Theorem 3. *The following inequality hold:*

$$\kappa_{\text{eff}}(M^{-1}PA) < \kappa_{\text{eff}}(M^{-1}A) \tag{6}$$

Proof. Note first that the columns of Z are no elements of the null space of A, otherwise Assumption 2 cannot be satisfied. Therefore, $z_i \in \mathbb{R}^n \setminus \text{Null}(A)$ holds, where z_i is a vector of Z. Now, the proof of (6) is similar to the proof of [22] (Theorem 2.2), which uses the unpreconditioned variant of (6)

$$\kappa_{\text{eff}}(PA) < \kappa_{\text{eff}}(A) \tag{7}$$

Inequality (7) can be shown similarly to the proof of [16] (Theorem 2.1). One only needs to replace [16] (Theorem 2.7, which is [10] Theorem 2.10) by Theorem 2, and it has to be encountered that rank $Azz^TA = 1$ also holds if A is singular, for an arbitrary vector $z \in \mathbb{R}^n \setminus \text{Null}(A)$.

Theorem 1 shows that P is determined by the space spanned by the columns of Z rather than the actual columns. This has consequences for constructing the deflation vectors (see Section 5.1). Theorem 2 shows that the effective condition number of $M^{-1}PA$ becomes more favorable by an increasing number of vectors in Z; hence, a better convergence of the iterative process is expected. Finally, Theorem 3 shows that $M^{-1}PA$ is always better conditioned than $M^{-1}A$; therefore, the convergence of DPCG is expected to be faster than the original PCG method.

3.2 Deflation Properties with Singular Coarse Matrix

Under certain circumstances, the results given in the previous subsection also hold for the case with singular matrix E. We start with Assumption 3 that is satisfied throughout this subsection.

Assumption 3. *Suppose that $Z = [Z_{k-1}\ z_k]$ holds, where Z_{k-1} is an $n \times (k-1)$ matrix and z_k is a vector. Let $E := Z^TAZ$ be a $k \times k$ singular SPSD matrix, whose pseudoinverse E^+ satisfies*

$$E^+ = \begin{bmatrix} E_{k-1}^{-1} & 0_{k-1} \\ 0_{k-1}^T & 0 \end{bmatrix} \tag{8}$$

where $E_{k-1} := Z_{k-1}^TAZ_{k-1}$ is an invertible SPD matrix.

Note that E_{k-1} of Assumption 3 can be interpreted as the nonsingular matrix E from the previous subsection. Now, the next lemma can be easily shown.

Lemma 1. (Theorem 4.3 of [11]) *The equality $ZE^+Z^T = Z_{k-1}E_{k-1}^{-1}Z_{k-1}^T$ holds.*

If we define $Q_{k-1} := Z_{k-1}E_{k-1}^{-1}Z_{k-1}^T$ and $P_{k-1} := I - AQ_{k-1}$, then Lemma 1 implies $Q = Q_{k-1}$; and therefore, this yields $M^{-1}PA = M^{-1}P_{k-1}A$. In other words, the deflation method has two identical variants in exact arithmetic: one variant with the invertible coarse matrix and the other variant with the singular coarse matrix. Moreover, another consequence of Lemma 1 is that all results given in the previous subsection can be generalized for the case with a singular E. Before we give these results, we need the following assumption.

Assumption 4. *Suppose that $A \in \mathbb{R}^{n\times n}$ is a singular SPSD matrix and M is a nonsingular SPD matrix. Z is chosen in such a way that E is a singular SPD matrix satisfying Assumption 3.*

Throughout this subsection, we will use

$$\begin{aligned} P_i &:= I - AQ_i, \quad Q_i := Z_iE_i^+Z_i^T \\ E_i &:= Z_i^TAZ_i, \quad Z_i \in \mathbb{R}^{n\times k_i} \end{aligned} \tag{9}$$

for $i = 3, 4$, which should satisfy Assumption 4. Now, the generalization of the results given in the previous subsection can be found below.

Theorem 4. *Let $k_3 = k_4 = k$. If Col(Z_3) = Col(Z_4), then $M^{-1}P_3A = M^{-1}P_4A$, and in particular, $Q_3 = Q_4$.*

Theorem 5. *Suppose that P_3A has $k_3 + 1$ zero-eigenvalues and P_4A has $k_4 + 1$ zero-eigenvalues.*

If Col(Z_3) $\subseteq$ Col(Z_4), then $\kappa_{\text{eff}}(M^{-1}P_4A) \leq \kappa_{\text{eff}}(M^{-1}P_3A)$.

Theorem 6. *The following inequality holds:* $\kappa_{\text{eff}}(M^{-1}P_3A) < \kappa_{\text{eff}}(M^{-1}A)$.

If Eq. (8) does not hold, then the above theorems cannot proved in general. However, it turns out that they are still satisfied in certain cases (see Section 5.1).

3.3 Comparison of Nonsingular and Singular Coarse Matrices

As concluded in the previous subsection, one can use either the singular matrix E or nonsingular matrix E_{k-1} in the deflation method. If one solves the corresponding coarse systems in a direct way, then extra care is needed in the case of E, by generating a solution up to the null space of E. The corresponding coarse systems can also be solved iteratively. In this subsection, we investigate the spectra and the condition numbers of E or E_{k-1}, so that it can be determined which case is expected to give the fastest convergence.

If Assumption 3 is fullfilled, then it follows immediately that the nonzero eigenvalue distributions of E and E_{k-1} are identical. In this case, the associated expected convergence behaviors are the same.

Assume now that Assumption 3 is satisfied with

$$E = \begin{bmatrix} E_{k-1} & \times \\ \times & \times \end{bmatrix}$$

instead of Eq. (8), where $\times$ means elements not all equal to zero. Then, we can show that the eigenvalues of E_k and E_{k-1} interlace, see Theorem 7.

Theorem 7. *The following inequalities holds:*

$$\begin{aligned} 0 &= \lambda_1(E) \leq \lambda_1(E_{k-1}) \leq \lambda_2(E) \leq \ldots \\ &\leq \lambda_{k-1}(E) \leq \lambda_{k-1}(E_{k-1}) \leq \lambda_k(E) \end{aligned}$$

Proof. The theorem follows immediately from the interlacing property (see, e.g., [18], p. 396].

In contrast to the case that A is invertible, it can be noted that the spectrum of E_{k-1} is not in the range of the nonzero eigenvalues of E [i.e., $\kappa(E_{k-1})$ is not smaller than $\kappa_{\text{eff}}(E)$]. But, in practice, we often see that the largest eigenvalues of both matrices are almost the same, whereas the smallest nonzero eigenvalues differ significantly, [i.e., we have $\lambda_{k-1}(E_{k-1}) \to \lambda_k(E_k)$ and $\lambda_1(E_{k-1}) \ll \lambda_2(E_k)$ for large n or k]. Hence, this yields $\kappa_{\text{eff}}(E) \leq \kappa(E_{k-1})$. Thus, the conclusion is that one should iterate with E_k rather than E_{k-1}, to obtain the fastest convergence of the iterative coarse solves. This will be done in Section 5.

3.4 Stabilization of Deflation Method

In practice, DPCG can be unstable for problems with a large number of unknowns. This is caused by the fact that coarse sytems involving E or E_{k-1} might also become large and cannot be solved accurately. To stabilize the method, we can choose for the so-called adapted DPCG (ADPCG) method (see also, e.g., [19], where ADPCG is known as the A-DEF2 method). In this method, we have to solve

$$(P^TM^{-1} + Q)Ax = (P^TM^{-1} + Q)b \qquad (10)$$

with starting vector $x_0 = Qb + P^T\tilde{x}_0$ and an arbitrary vector $\tilde{x}_0$. Note that the operator $P^TM^{-1} + Q$ in Eq. (10) cannot be replaced by $M^{-1}P + Q$, because it has been shown in [19] that the resulting method suffers from instability in some cases. Furthermore, note that the solution x in (10) is the same as the solution given in (1). It has also been shown in [19] that ADPCG can be derived from the well-known balancing Neumann-Neumann method [20] and ADPCG has the same stability properties as this method. Finally, the reason that ADPCG is more stable than DPCG follows from the following theorem.

Theorem 8.(Theorem 3.1 of [19]) *Let the spectra of DPCG and ADPCG be given by*

$$\begin{aligned} \sigma(M^{-1}PA) &= \{\lambda_1, \ldots, \lambda_n\} \\ \sigma(P^TM^{-1}A + QA) &= \{\mu_1, \ldots, \mu_n\} \end{aligned}$$

respectively. Then, the numbering of the eigenvalues within these spectra can be such that the following statements hold:

$$\begin{cases} \lambda_i = 0,\ \mu_i = 1, & \text{for } i = 1, \ldots, k \\ \lambda_i = \mu_i, & \text{for } i = k+1, \ldots, n \end{cases}$$

The eigenvalues $\lambda_i = 0$ associated with DPCG become nearly zero, if coarse systems with E or E_{k-1}

are solved inaccurately. Therefore, DPCG leads to an unstable method. On the other hand, we do not have this phenomenon in the ADPCG method, because the corresponding eigenvalues of $M^{-1}A$ are projected to 1 instead of 0 (see [19] for more details). It follows that if coarse systems $Ev = w$ are solved iteratively, then this can be done with a lower accuracy for ADPCG, compared to DPCG. Let us denote the stopping tolerance of PCG by ϵ_{outer} and the stopping tolerance of the coarse systems $Ev = w$ by ϵ_{inner}. Suppose that

$$\epsilon_{\text{inner}} = \omega \epsilon_{\text{outer}}, \quad \omega > 0 \tag{11}$$

Then, $\omega \leq 10^{-2}$ is a safe choice in the DPCG method for many test problems (see [11], Section 4.1). Indeed, the coarse systems should be solved very accurately. Following the discussion above, the expectation is that a larger ω can be taken in the ADPCG method. In the numerical experiments (see Section 5), we will investigate the choice of ω for both DPCG and ADPCG in more detail.

We remark that if coarse systems $Ev = w$ are solved inaccurately, the resulting operator $P^T M^{-1} + Q$ is varying at each iteration, while a fixed operator is expected in the CG iteration. In other words, the step $y_2 = (P^T M^{-1} + Q)^{-1} y_1$ seen by the outer process turns out to be $y_2 = \mathcal{P}(y_1)$, where $\mathcal{P}$ is a nonlinear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. If the inner tolerance is too loose, then the optimal convergence property of CG can only be preserved if one performs a full orthogonalization of the search direction vectors that can be extended with truncation and restart strategies. This results in a GMRES-like method, namely the flexible CG method (see [21]). We will also consider this variant in Section 5. However, note that it is possible to use the original CG method with inexact preconditioning because it can be shown that the convergence rate of the outer CG process can be maintained up to a certain accuracy for the inner iterations (see, e.g., [22,23]).

4. BUBBLY FLOWS

Bubbly flows are governed by the incompressible Navier-Stokes equations which read

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) + \mathbf{f} \tag{12}$$

subjected to the incompressibility constraint

$$\nabla \cdot \mathbf{u} = 0 \tag{13}$$

where $\mathbf{u} = (u, v, w)^T$ is the velocity vector and ρ, p, μ, and $\mathbf{f}$ are the density, pressure, viscosity, and source vector (consisting of gravity and interface tension forces), respectively. We assume the density and viscosity to be constant within each fluid.

Equations (12) and (13) can be solved on a Cartesian grid in a rectangular domain by, e.g., the pressure-correction method [24]. These equations are discretized using finite differences on a staggered grid. In the pressure-correction method, first a tentative velocity vector $\mathbf{u}$ is computed by

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\nabla \cdot \mathbf{u}^n \mathbf{u}^n + \frac{1}{\rho} \nabla \cdot \mu \left[\nabla \mathbf{u}^* + (\nabla \mathbf{u}^n)^T \right] \tag{14}$$

where $\mathbf{u}^n$ denotes the velocity at time step n. The resulting system of equations with unknown vector $\mathbf{u}^*$ is solved by, for example, the PCG method. The velocities at the new time step $n+1$ are determined by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho} \mathcal{G} p + \mathbf{f}$$

under the constraint of (13). This yields

$$\begin{cases} \mathbf{u}^{n+1} &= \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} \mathcal{G} p + \mathbf{f} \right) \\ \mathcal{D} \mathbf{u}^{n+1} &= 0 \end{cases} \tag{15}$$

where $\mathcal{D}$ represents the discretization of the divergence and $\mathcal{G}$ is the discrete gradient operator. Finally, Eq. (15) implies

$$\mathcal{D} \frac{1}{\rho} \mathcal{G} p = \mathcal{D} \left(\frac{1}{\Delta t} \mathbf{u}^* + \mathbf{f} \right) \tag{16}$$

Equation (16) can again be solved using the PCG method. It can be noted that solving (16) takes significantly more computing time compared to solving (14) because the convergence of the iterative process suffers from the highly discontinuous behavior of the density ρ. The situation is even worse for relatively large problems: to find the solution of (16) requires the bulk of the computations in the whole pressure-correction method. Therefore, we aim at solving (16) efficiently in this paper. Further details about the pressure-correction method applied to bubbly flows can be found in, e.g., [5,6].

At the boundaries of the domain, we impose Dirichlet boundary conditions for the velocity. Because of the staggered grid, we do not have pressure points at the boundaries of the domain. They are also not required in the method because the velocity boundary conditions are included in the discrete divergent operator $\mathcal{D}$ in Eq. (16). It follows that pressure Neumann boundary conditions should hold implicitly. In this case, the pressure is a relative variable because the differences in pressure and not the absolute values are meaningful in the pressure-correction method.

Equation (16) can be written as a symmetric and positive semidefinite (SPSD) linear system [cf. Eq. (1)]

$$Ax = b, \quad A \in \mathbb{R}^{n\times n} \tag{17}$$

where A is singular. Assumption 5 will hold in our bubbly flow applications.

Assumption 5. *$A\mathbf{1}_n = \mathbf{0}_n$ and $b^T\mathbf{1}_n = 0$, and the algebraic multiplicity of the zero-eigenvalue of A is one.*

The solution x of Eq. (17) is determined up to a constant, i.e., if x_1 is a solution then $x_1 + c\mathbf{1}_n$ with an arbitrary $c \in \mathbb{R}$ is also a solution. This situation presents no real difficulty for the iterative solver, as long as $b \in \text{Col}(A)$.

We consider two-phase bubbly flows with, for instance, air (low-density phase) and water (high-density phase). In this case, ρ is piecewise constant with a contrast, $\delta > 0$, which is the ratio of the two densities, i.e.,

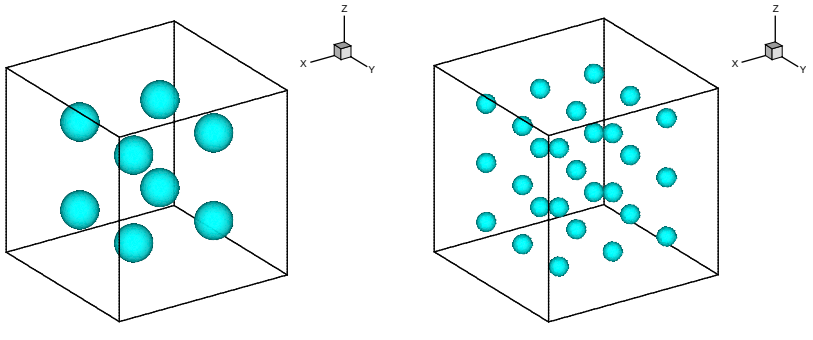
$$\rho = \begin{cases} \rho_0 = 1, & \mathbf{x} \in \Lambda_0 \\ \rho_1 = \delta, & \mathbf{x} \in \Lambda_1 \end{cases} \tag{18}$$

where Λ_1 is the low-density phase, namely, bubbles in Ω, and Λ_0 is the high-density phase, namely, the fluid domain around these bubbles.

5. NUMERICAL EXPERIMENTS

In this section, we present results for some numerical experiments with 3D bubbly flow problems, which will illustrate the theoretical results obtained in Section 3. The computations are performed on an Intel Core 2 Duo (2.66 GHz) computer with a memory capacity of 8 GB. Moreover, the code is compiled with FORTRAN g77 on LINUX.

We choose for a deflation subspace matrix Z consisting of subdomain deflation vectors, which are described in Section 5.1. The preconditioner M is taken to be an incomplete Cholesky factorization [25] based on A. For both DPCG and ADPCG, the singular coarse systems $Ev = w$ are solved using PCG with again the incomplete Cholesky factorization preconditioner, but without deflation. Section 5.2 is devoted to the bubbly flow problem with eight air bubbles in water, and subsequently, the problem with a lot of small air bubbles in water is investigated in Section 5.3. The corresponding geometries of these test problems can be found in Fig. 1. It is known (see, e.g., [17]) that more bubbles in the domain lead to a coefficient matrix that is more ill-conditioned, and therefore, the corresponding problem is more difficult to solve. Finally, we discuss the results in more detail in Section 5.4.



(a) (b)

FIGURE 1. Geometry of bubbly flow test problems. **a)** Test problem 1: 8 air bubbles in water and **b)** Test problem 2: 27 air bubbles in water

5.1 Construction of Z

As mentioned in Section 2, deflation subspace matrix Z remains to be specified. In the ideal case with respect to convergence, Z should consist of eigenvectors associated with the smallest eigenvalues of $M^{-1}A$, so that those unfavorable eigenvalues are eliminated from the spectrum. However, the computation of these eigenvectors can be very expensive, and in addition, these dense vectors might be inefficient in use because they require, relatively, much memory and lead to dense matrices AZ and E. Therefore, we will use sparse subdomain deflation vectors, that appear to be good approximations of those eigenvectors.

Let the open domain Ω be divided into subdomains $\Omega_j,\ j = 1, 2, \ldots, k$, such that $\overline{\Omega} = \cup_{j=1}^{k} \overline{\Omega}_j$ and $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j$. The discretized domain and subdomains are denoted by Ω_h and Ω_{h_j}, respectively. Then, for each Ω_{h_j} with $j = 1, 2, \ldots, k$, we introduce a subdomain deflation vector, z_j, as follows:

$$(z_j)_i := \begin{cases} 0, & x_i \in \Omega_h \setminus \Omega_{h_j} \\ 1, & x_i \in \Omega_{h_j} \end{cases}$$

where x_i is a grid point in the discretized domain, Ω_h. Then, for $k > 1$, we define the deflation subspace matrices Z_{k-1} and Z as follows:

$$\begin{aligned} Z_{k-1} &:= [z_1\ z_2\ \cdots\ z_{k-1}] \\ Z &:= [Z_{k-1}\ z_k] = [z_1\ z_2\ \cdots\ z_k] \end{aligned} \tag{19}$$

Obviously, Z consists of disjunct orthogonal piecewise-constant vectors. Therefore, Z is sparse, so that it can be stored efficiently in memory and computations with Z can be carried out at low cost (see [11] Section 3) for more details. Because of the construction of Z and Z_{k-1}, coarse matrix E is singular, while coarse matrix $E_{k-1} := Z_{k-1}^T A Z_{k-1}$ appears to be nonsingular.

If one replaces z_k by $\mathbf{1}_n$, then it can be proven that it leads to the same deflation matrix P (see [11] Section 4.2). Hence, Assumption 4 can be satisfied in our case, because $A\mathbf{1}_n = \mathbf{0}_n$ follows from Assumption 5. Then, it follows from Lemma 1 that the use of both deflation subspace matrices given in (19) results in the same P. Therefore, we do not lose generality in the numerical experiments, if we restrict ourselves to iterative solvers for coarse systems based on Z rather than Z_{k-1}. Moreover, Theorems 4–6 can now also be applied. For example, we know from Theorem 4 that the success of the deflation method is determined by the space spanned by subdomain deflation vectors instead of the separate vectors. Of course, the space should approximate the eigenvectors associated with small eigenvalues of $M^{-1}A$ in order to obtain an effective deflation method. For our bubbly flow applications, this can be done using the subdomain deflation vectors. It appears that the subdomains can be chosen independently of the density field, as long as the number of deflation vectors k is sufficiently large (see [17] Section 7).

5.2 Test Problem 1

We consider the problem setting as given in Section 4, where the test problem is taken with eight airbubbles in the unit domain filled with water. In this case, the ratio between the two phases is 10^3 [i.e., $\delta = 10^{-3}$ is chosen in Eq. (18)]. The bubbles are spheres with a constant radius of 0.10 [see Fig. 1a].

We examine PCG and the deflation methods DPCG and ADPCG for different parameters of ϵ_{inner} and number of deflation vectors k on a grid of $n = 100^3$. The stopping criterion is the norm of the relative residuals (i.e., $||r_j||_2/||r_0||_2$, should be below a tolerance $\epsilon_{\text{outer}} = 10^{-8}$). The results of this test problem can be found in Table 1.

From Table 1, we see in all test cases that DPCG and ADPCG are always faster and require fewer iterations compared to PCG, which confirms Theorem 6. Both deflation methods require approximately the same number of iterations for fixed k, which is as expected from Theorem 8. Moreover, it can be observed that increasing the number of deflation vectors k leads to a reduction of the number of iterations for both DPCG and ADPCG. This is in agreement with Theorem 5. Furthermore, we have expected that ADPCG is more stable than DPCG due to Theorem 8. This is indeed the case: when $\epsilon_{\text{inner}} \leq 10^{-8}$, DPCG did not converge anymore, while ADPCG still shows convergence, provided that $\epsilon_{\text{inner}} \leq 10^{-4}$. Finally, we note there is an optimum considering the computing time for specific k and corresponding ϵ_{inner}. For DPCG, this is $k = 20^3$ and $\epsilon_{\text{inner}} = 10^{-10}$, whereas $k = 20^3$ and $\epsilon_{\text{inner}} = 10^{-4}$ are the optimal values in the case of ADPCG. Hence, it can be seen that ADPCG can be faster than DPCG.

TABLE 1. Results for DPCG and ADPCG to solve the linear system $Ax = b$ with $n = 100^3$, corresponding to Test Problem 1. PCG requires 390 iterations and 37.0 s to converge. Iter = number of iterations of the outer process, CPU = the required computing time (in seconds) including the setup time of the methods, N/C = no convergence within 250 iterations

		$k = 5^3$		$k = 10^3$		$k = 20^3$		$k = 25^3$	
Method	ϵ_{inner}	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
DPCG	10^{-10}	151	17.9	66	8.5	32	5.8	28	6.5
	10^{-8}	N/C	–	N/C	–	N/C	–	N/C	–
ADPCG	10^{-10}	140	20.2	60	9.2	30	7.2	27	10.1
	10^{-8}	140	20.1	60	9.1	30	6.7	27	9.4
	10^{-6}	140	20.1	60	9.1	30	6.3	27	8.2
	10^{-4}	141	20.2	60	9.0	29	5.6	29	7.0
	10^{-2}	N/C	–	194	28.2	N/C	–	N/C	–

5.3 Test Problem 2

We repeat the same experiment as performed in the previous subsection, but we now take $n = 150^3$ and 27 airbubbles, which have a smaller constant radius of 0.05 (see Fig. 1b). The results of this experiment can be found in Table 2.

First, it can be observed in Table 2 that both DPCG and ADPCG require more iterations and computing time, compared to the results as given for Test Problem 1. This is due to a combination of the increased grid sizes n and an increased number of bubbles. It appears that if k is sufficiently large, then the number of iterations remains approximately the same for varying number of bubbles. For varying grid sizes, the number of iterations can only be kept constant, if k is chosen proportionally to n. More details about experiments with varying grid sizes and the number of bubbles can be found in [11,12].

From Table 2, we see that the conclusions drawn in the previous subsection also hold in this test problem. In fact, the results in Table 2 emphasize these conclusions. For example, both DPCG and ADPCG are faster than PCG as long as they converge. We note that the benefit of using a larger ϵ_{inner} in ADPCG can be huge for large k. In addition, the optimal choices are $k = 15^3$ and $\epsilon_{\text{inner}} = 10^{-10}$ for DPCG, and $k = 25^3$ and $\epsilon_{\text{inner}} = 10^{-4}$ for ADPCG. In contrast to Test Problem 1, it can be observed that the optimal DPCG is always faster than ADPCG in this test problem.

5.4 Discussion of the Results

From the above results of Test Problems 1 and 2, it can be observed that the optimal values are $\omega = 10^{-2}$ for DPCG and $\omega = 10^4$ for ADPCG with respect to Eq. (11). These still hold if we vary ϵ_{outer}. Apparently, DPCG can deal with nearly zero eigenvalues, as long as they are very small, so that they are treated as zero eigenvalues by the method. In addition, ADPCG is faster than DPCG in some cases, because a larger ϵ_{inner} can be taken while the number of outer iterations remain approximately the same. This is rather surprising, because no extra orthogonalization steps considering the search directions or residuals have been added to the iterative process, in order to preserve the known orthogonality properties of CG.

We investigate the inner-outer iterations in more detail. Note first that each outer iteration of ADPCG requires two inner solves, whereas DPCG only needs one [cf. Eq. (3) and (10)]. Therefore, ADPCG can only be more efficient, if each inner solve of this method can be performed at least twice as fast as DPCG, which is the case for sufficiently large ϵ_{inner}. This is illustrated in Fig. 2, which shows a typical convergence of the residuals of an inner solve, within an outer iteration of ADPCG for $k = 25^3$. It can be observed that ADPCG can only be faster than DPCG, if the inner solves would be reduced from 142 to at most 71. This means that, in theory, one has to perform the inner solves with an accuracy of approximately $\epsilon_{\text{inner}} \leq 10^{-5}$, which can indeed be achieved in ADPCG (see Tables 1 and 2). Moreover, we remark that if k becomes relatively large, then

TABLE 2. Results for DPCG and ADPCG to solve the linear system $Ax = b$ with $n = 150^3$, corresponding to Test Problem 2. PCG requires 543 iterations and 177.6 s to converge

		$k = 15^3$		$k = 25^3$		$k = 50^3$	
Method	ϵ_{inner}	Iter.	CPU	Iter.	CPU	Iter.	CPU
DPCG	10^{-10}	53	24.1	44	25.1	24	82.1
	10^{-8}	N/C	–	N/C	–	N/C	–
ADPCG	10^{-10}	50	27.6	41	32.5	22	130.4
	10^{-8}	50	27.2	41	30.7	22	116.0
	10^{-6}	50	26.7	42	29.3	22	86.2
	10^{-4}	52	27.4	43	27.0	24	58.2
	10^{-2}	N/C	–	N/C	–	N/C	–

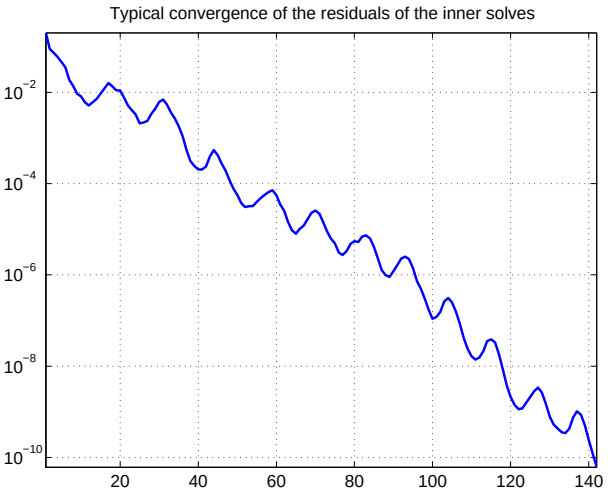


FIGURE 2. Convergence of the residuals during an inner solve at one iteration of ADPCG with $k = 25^3$ (Test Problem 2). The plots are similar for the other outer iterations of the same test case, since one applies the inaccurate solves on the same matrix E

matrix E would also be very large. Then, it is inevitable to use DPCG or ADPCG instead of PCG in order to solve $Ev = w$ efficiently. In this case, we would obtain an iterative method with a multilevel preconditioning.

Next, we examine the residuals of the outer iterations to see what happens if a method does not converge (see Fig. 3). From Fig. 3, we can observe that if DPCG shows no convergence, it even diverges. This is in contrast to ADPCG, whose residuals are still slowly decreasing. This is an extra advantage of ADPCG. Although it might not be the fastest method, it gives somewhat more stable residuals in case it converges slowly.

The reason that ADPCG does not work for $\epsilon_{\text{inner}} \leq 10^{-4}$ is twofold. On the one hand, solving $Ev = w$ with low accuracy can be interpreted as computing $v = E^+w$ with a strongly perturbed matrix E^+. As concluded in Section 3.4, the associated spectrum of ADPCG remains the same if E^+ is slightly perturbed. Large perturbations of E^+ can lead to the appearance of relatively small eigenvalues in the spectrum, which cause the slow convergence of the method. On the other hand, as mentioned in Section 3.4, the CG algorithm cannot deal with (strongly) varying preconditioners because orthogonal properties of the residuals and search directions are not guaranteed anymore. This problem might be solved by using flexible CG instead of CG, but experiments show that this does not lead to better results. In Fig. 4, the results can be found for one test case of Test Problem 1, where ADPCG is used with both original CG and flexible CG without restart or truncation strategies, denoted by original ADPCG and flexible ADPCG, respectively. It can be readily noted in Fig. 4 that the flexible variant might lead to a convergent method (see the case of $\epsilon_{\text{inner}} = 10^{-4}$), but it requires too many iterations to be an effective method. The situation will be even worse, if restart or truncation strategies would be added.

6. CONCLUSIONS

Some new theoretical results are obtained for the deflation method applied to singular systems. Moreover, some insights into stabilizing the deflation method and its corresponding inner-outer itera-

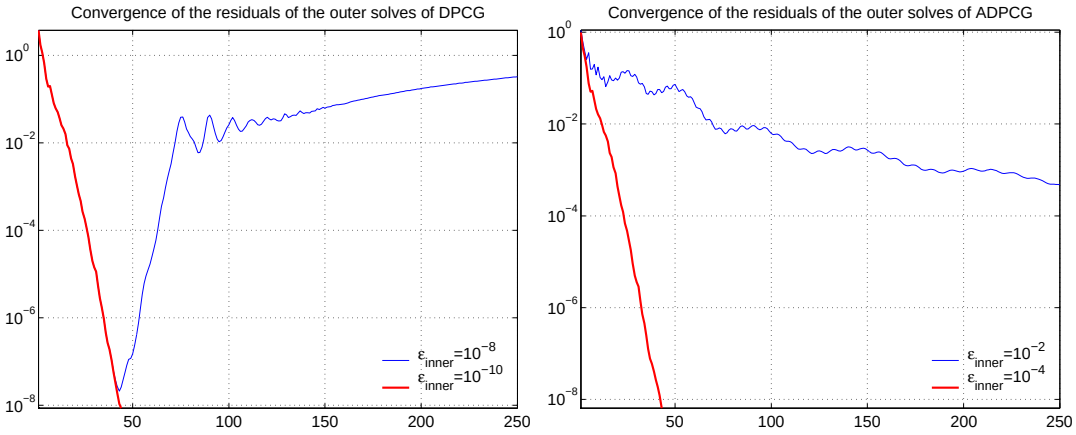


FIGURE 3. Convergence of the residuals of the outer iterations from DPCG and ADPCG with $k = 25^3$ (Test Problem 2). **a)** DPCG, **b)** ADPCG

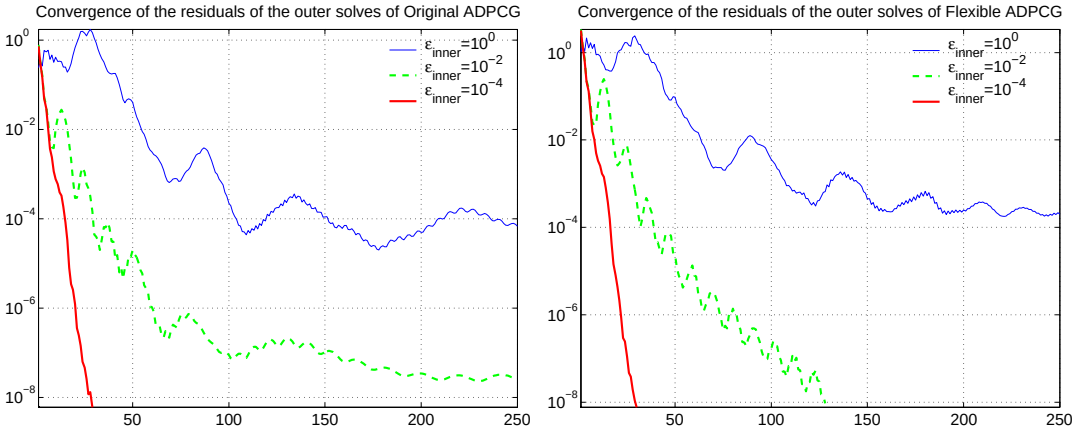


FIGURE 4. Convergence of the residuals of the outer iterations from ADPCG with $k = 25^3$ (Test Problem 1). **a)** Original ADPCG, **b)** Flexible ADPCG

tions are given. The numerical experiments confirm the theoretical results and show that the deflation method can indeed be stabilized without losing much efficiency. The experiments emphasize that both deflation methods are very effective and fast solvers for the pressure equation derived from two-phase bubbly flow problems.

ACKNOWLEDGMENTS

Part of this research has been funded by the Dutch BSIK/BRICKS project.

REFERENCES

1. Coyajee, E. R. A., A front-capturing method for the numerical simulation of dispersed twophase flow. Ph.D. thesis, Delft University of Technology, 2007.
2. de Sousa, F. S., Mangiavacchi, N., Nonato, L. G., Castelo, A., Tome, M. F., Ferreira, V. G., Cuminato, J. A., and McKee, S., A front-tracking/front-capturingmethod for the simulation of 3d multi-fluid flows with free surfaces. *J. Comput. Phys.* **198**:469-499, 2004.

3. Hua, J., and Lou, J., Numerical simulation of bubble rising in viscous liquid. *J. Comput. Phys.* **222**:769-795, 2007.
4. Sussman, M., Smith, K. M., Hussaini, M. Y., Ohta, M., and Zhi-Wei, R., A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.* **221**:469-505, 2007.
5. Van der Pijl, S. P., Segal, A., and Vuik, C., Modelling of three-dimensional multi-phase flows with a mass-conserving level-set method. Report 06-10, Delft University of Technology, Delft Institute of Applied Mathematics, Delft, 2006.
6. Van der Pijl, S. P., Segal, A., Vuik, C., and Wesseling, P., A mass-conserving level-set method for modelling of multi-phase flows. *Int. J. Numer. Methods Fluids.* **47**:339-361, 2005.
7. Nicolaides, R. A., Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.* **24**:355-365, 1987.
8. Vuik, C., Segal, A., and 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.
9. Vuik, C., Segal, A., Meijerink, J. A., and Wijma, G. T., The construction of projection vectors for a Deflated ICCG method applied to problems with extreme contrasts in the coefficients. *J. Comp. Phys.* **172**:426-450, 2001.
10. Nabben, R., and Vuik, C., A comparison of Deflation and Coarse Grid Correction applied to porous media flow. *SIAM J. Numer. Anal.* **42**:1631-1647, 2004.
11. Tang, J. M., and Vuik, C., Efficient deflation methods applied to 3-D bubbly flow problems. *Elec. Trans. Numer. Anal.* **26**:330-349, 2007.
12. Tang, J. M., and Vuik, C., On deflation and symmetric positive semi-definite matrices. *J. Comput. Appl. Math.* **206**:603-614, 2007.
13. Trottenberg, U., Oosterlee, C. W., and Schuller, A., Multigrid. Academic Press, London, 2000.
14. Wesseling, P., *An Introduction to Multigrid Methods*. Wiley, Chichester, 1992 corrected reprint, R.T. Edwards, Inc., Philadelphia, 2004.
15. Nabben, R., and Vuik, C., A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.* **27**:1742-1759, 2006.
16. Vuik, C., Nabben, R., and Tang, J. M., Deflation acceleration for domain decomposition preconditioners. In Wesseling, P., Oosterlee, C. W., and Hemker, P., ed., *Proc. of 8th European Multigrid Conference*, Sept. 27–30, 2005, Scheveningen, The Netherlands. TU Delft, 2006.
17. Tang, J. M., and Vuik, C., New variants of deflation techniques for bubbly flow problems. *J. Numer. Anal. Indust. Appl. Math.* **2**:227-249, 2007.
18. Golub, G. H., and van Loan, C. F., Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
19. Tang, J. M., Nabben, R., Vuik, C., and Erlangga, Y. A., Theoretical and numerical comparison of various projection methods derived from deflation, domain decomposition and multigrid methods. Report 07-04, Delft University of Technology, Delft Institute of Applied Mathematics, Delft, 2007.
20. Mandel, J., Balancing domain decomposition. *Comm. Numer. Meth. Eng.* **9**:233-241, 1993.
21. Notay, Y., Flexible conjugate gradients. *SIAM J. Sci. Comput.* **22**:1444-1460, 2000.
22. Golub, G. H., and Overton, M. L., The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems, *Numer. Math.* **53**:571-593, 1988.
23. Golub, G. H., and Ye, Q., Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comput.* **21**:1305-1320, 2000.
24. Van Kan, J., A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comput.* **7**:870-891, 1986.
25. Meijerink, J. A., and van der Vorst, H. A., An iterative solution method for linear equations systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.* **31**:148–162, 1977.