A GENERALIZED PROJECTED CG METHOD WITH APPLICATIONS TO BUBBLY FLOW PROBLEMS

JOK M. TANG †

Abstract. For various applications, it is well-known that a two-level preconditioned CG method is an efficient method for solving large and sparse linear systems whose coefficient matrix is SPD. 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. The resulting methods are called projection methods. In the literature, various projection methods are known, coming from the fields of deflation, domain decomposition and multigrid. At first glance, these methods seem to be different. However, from an abstract point of view, they are closely related to each other theoretically. We show that a generalized projected CG method can be formulated based on these methods. The various variants are also compared numerically in order to test their robustness, where our main applications are bubbly flows. The computational cost for the simulations of bubbly flows is often dominated by the solution of a linear system corresponding to a pressure Poisson equation with discontinuous coefficients. We will explore the efficient solution of these linear systems using projection methods. Some of these methods turn out to be very fast and robust, resulting in efficient calculations for bubbly flows on highly resolved grids. We conclude with a suggestion of a two-level preconditioner for the CG method, that is as robust as the abstract balancing preconditioner and nearly as cheap and fast as the deflation preconditioner.

Key words. deflation, domain decomposition, multigrid, conjugate gradients, two-grid schemes, preconditioning, projection methods, bubbly flows.

1. Introduction. The Conjugate Gradient (CG) method [10] is a very popular iterative method for solving $Ax = b$, whose coefficient matrix, $A \in \mathbb{R}^{n\times n}$, is large, sparse and symmetric positive definite (SPD). It is well-known that the convergence rate of CG is bounded in terms of the condition number of A. If this condition number is large, it is more efficient to solve a preconditioned system, $M^{-1}Ax = M^{-1}b$. Traditional preconditioners such as incomplete factorization preconditioners are widely used for M, but appear to be less effective for applications with highly refined grids and flows with high coefficient ratios in the original PDEs.

In addition to the traditional preconditioner, a second kind of preconditioner can be incorporated to improve the conditioning of the coefficient matrix. This is also known as 'two-level preconditioning', and the resulting iterative method is called a 'projection(-preconditioning) method'. Several projection methods are known in the fields of deflation, multigrid (MG) and domain decomposition (DDM). The idea of deflation is to eliminate eigenvectors associated with unfavorable eigenvalues of $M^{-1}A$, so that they are no longer present during the iterations, see [13, 22, 26, 42]. Projection methods based on MG rely on a two-grid preconditioning with relaxation and coarse-grid correction, where a second-level problem is derived from the systematic coarsening of the underlying fine grid, see [2, 4, 24, 25, 28, 39, 40, 43]. Projection methods known in DDM aim at solving the linear sytem by solves on subdomains and improving the overall convergence by using a coarse-grid correction, see [14–16, 28, 35].

At first glance, the projection methods from deflation, DDM and MG seem to be different. For example, in deflation, eigenvector approximations associated with the unfavorable eigenvalues are often used as projection vectors. In contrast, MG or DDM use special projection vectors, which represent interpolations between the fine-grid and coarse-grid subspace. Surprisingly, from an algebraic and abstract point of view, the projection methods are comparable or even equivalent, which will be shown in this paper. We will introduce a general framework for deflation, MG and DDM, resulting in a unified theory. The projection methods will be compared theoretically by investigating their spectral properties and the

†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@tudelft.nl).
Part of the work has been funded by the Dutch BSIK/BRICKS project. This is a joint work with Kees Vuik (Delft University of Technology) and Reinhard Nabben (Technische Universität Berlin).

equivalences between them. Thereafter, the focus will be on numerical experiments, where the projection methods will be tested for their convergence properties and robustness.

The numerical experiments will be based on bubbly flow problems. Computation of bubbly flows is a very active research topic in computational fluid dynamics, see the recent papers [5, 11, 17, 27, 29, 37, 38]. Bubbly flows are complicated to simulate, because the internal geometry of the problem typically varies with time and the fluids involved can have very different material properties. Mathematically, bubbly flows are modelled using the Navier-Stokes equations, which can be approximated numerically using operator-splitting techniques. In these schemes, equations for the velocity and pressure are solved sequentially at each time step. In many popular operator-splitting methods, the pressure-correction is formulated implicitly, requiring the solution of a linear system at each time step. This system with a symmetric positive semi-definite (SPSD) coefficient matrix takes the form of a pressure Poisson equation with discontinuous coefficients. Solving these systems typically consumes the bulk of the computing time and has long been recognized as a computational bottleneck in fluid flow simulation [3, 11]. In the numerical experiments, the projection methods will be adopted to solve the pressure Poisson equation efficiently.

The remainder of this paper is organized as follows. In Section 2, we introduce the projection methods. Section 3 is devoted to the theoretical comparison of these methods, followed by the numerical comparison in Section 4. Finally, conclusions are given in Section 5. This paper is based on [30,32], where more details can be found about this work.

2. Projection Methods. In this section, projection methods and their algorithms will be defined. A review of their origins from the different fields can be found in [30].

DEFINITION 2.1. *Suppose that an SPD matrix, $A \in \mathbb{R}^{n\times n}$, a right-hand side, $b \in \mathbb{R}^n$, an SPD preconditioner, $M \in \mathbb{R}^{n\times n}$, and a deflation-subspace matrix, $Z \in \mathbb{R}^{n\times k}$, with full rank and $k < n$ are given. Then, we define the invertible matrix $E \in \mathbb{R}^{k\times k}$, the matrix $Q \in \mathbb{R}^{n\times n}$, and the deflation matrix, $P \in \mathbb{R}^{n\times n}$, as follows:*

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^TAZ,$$

where I is the $n \times n$ identity matrix.

Note that E is SPD for any full-rank Z, since A is SPD. If $k \ll n$ holds, then E is a matrix with small dimensions, so that it can be easily computed and factorized. In each field, the matrices as given in Definition 2.1 have different interpretations, see [30]. Nevertheless, from our point of view, matrices M and Z are arbitrary but the same for each projection method. In this way, the abstract setting allows us to compare the methods in terms of operators, although they have their roots in different fields.

The general linear system, that will be the basis for projection methods, is $\mathcal{P}Ax = \mathcal{P}b$, where the operator $\mathcal{P} \in \mathbb{R}^{n\times n}$ can be interpreted as a projector. Table 2.1 presents a list of the preconditioned CG and projection methods.

Name	**Method**	**Operator $\mathcal{P}$**	**References**
PREC	Traditional Preconditioned CG	M^{-1}	[8, 18]
AD	Additive Coarse-Grid Correction	$M^{-1} + Q$	[1, 28, 35]
DEF1	Deflation Variant 1	$M^{-1}P$	[42]
DEF2	Deflation Variant 2	P^TM^{-1}	[13, 22, 26]
A-DEF1	Adapted Deflation Variant 1	$M^{-1}P + Q$	[28, 36, 43]
A-DEF2	Adapted Deflation Variant 2	$P^TM^{-1} + Q$	[28, 36, 43]
BNN	Abstract Balancing	$P^TM^{-1}P + Q$	[14]
R-BNN1	Reduced Balancing Variant 1	$P^TM^{-1}P$	–
R-BNN2	Reduced Balancing Variant 2	P^TM^{-1}	[14, 35]

TABLE 2.1

List of methods which will be compared. Where possible, references to the methods and their implementations are given in the last column.

The general implementation of the projection methods given in Table 2.1 can be found in Algorithm 1. For each method, the corresponding matrices, $\mathcal{M}_i$, and vectors, $\mathcal{V}_{\text{start}}$ and $\mathcal{V}_{\text{end}}$, are presented in Table 2.2.

Algorithm 1 Generalized Projected CG Method for solving $Ax = b$.

Select arbitrary $\bar{x}$ and $\mathcal{V}_{\text{start}}, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{V}_{\text{end}}$ from Table 2.2
$x_0 := \mathcal{V}_{\text{start}},\ r_0 := b - Ax_0,\ y_0 := \mathcal{M}_1 r_0,\ p_0 := \mathcal{M}_2 y_0$
for $j := 0, 1, \ldots,$ until convergence **do**
  $w_j := \mathcal{M}_3 A p_j$
  $\alpha_j := (r_j, y_j)/(p_j, w_j)$
  $x_{j+1} := x_j + \alpha_j p_j$
  $r_{j+1} := r_j - \alpha_j w_j$
  $y_{j+1} := \mathcal{M}_1 r_{j+1}$
  $\beta_j := (r_{j+1}, y_{j+1})/(r_j, y_j)$
  $p_{j+1} := \mathcal{M}_2 y_{j+1} + \beta_j p_j$
end for
$x_{\text{it}} := \mathcal{V}_{\text{end}}$

Method	$\mathcal{V}_{\text{start}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{V}_{\text{end}}$
PREC	$\bar{x}$	M^{-1}	I	I	x_{j+1}
AD	$\bar{x}$	$M^{-1} + Q$	I	I	x_{j+1}
DEF1	$\bar{x}$	M^{-1}	I	P	$Qb + P^T x_{j+1}$
DEF2	$Qb + P^T \bar{x}$	M^{-1}	P^T	I	x_{j+1}
A-DEF1	$\bar{x}$	$M^{-1}P + Q$	I	I	x_{j+1}
A-DEF2	$Qb + P^T \bar{x}$	$P^T M^{-1} + Q$	I	I	x_{j+1}
BNN	$\bar{x}$	$P^T M^{-1} P + Q$	I	I	x_{j+1}
R-BNN1	$Qb + P^T \bar{x}$	$P^T M^{-1} P$	I	I	x_{j+1}
R-BNN2	$Qb + P^T \bar{x}$	$P^T M^{-1}$	I	I	x_{j+1}

TABLE 2.2
Choices of $\mathcal{V}_{start}, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ and $\mathcal{V}_{end}$ for each method, as used in Algorithm 1.

3. Theoretical Comparison. In this section, a comparison of eigenvalue distributions of the operators of the projection methods will be made and some equivalence relations between the methods will be derived. The proofs of the theoretical results will be omitted here, but can be found in [30]. The ratio of the largest and the smallest nonzero eigenvalue of a matrix, B, is called the condition number of B and will be denoted as $\kappa(B)$.

3.1. Spectral Analysis of the Projection Methods. It is known that the eigenvalue distribution of the system corresponding to PREC is always worse than those of the projection methods, see [41]. In [19, 20], it has further been shown that the condition number of DEF1 is not worse than that of both AD and BNN, i.e.,

$$\kappa\left(M^{-1}PA\right) \leq \kappa\left(M^{-1}A + QA\right) \quad \text{and} \quad \kappa\left(M^{-1}PA\right) \leq \kappa\left(P^T M^{-1} PA + QA\right),$$

for A, Z and M^{-1}, while such results between AD and BNN do not hold in general [21]. In addition to the comparisons of PREC, AD, DEF1 and BNN, more relations between the eigenvalue distribution of these and other projection methods are given below.

THEOREM 3.1. *The following two statements hold for the spectra, σ:*

- $\sigma\left(M^{-1}PA\right) = \sigma\left(P^T M^{-1} A\right) = \sigma\left(P^T M^{-1} PA\right)$;
- $\sigma\left((P^T M^{-1} P + Q)A\right) = \sigma\left((M^{-1}P + Q)A\right) = \sigma\left((P^T M^{-1} + Q)A\right)$.

As a consequence of Theorem 3.1, DEF1, DEF2, R-BNN1 and R-BNN2 can be interpreted as one class of projection methods with the same spectral properties, whereas BNN, A-DEF1 and A-DEF2 lead to another class of projection methods.

THEOREM 3.2. *Let the spectra of DEF1 and BNN be given by* $\sigma(M^{-1}PA) = \{\lambda_1, \ldots, \lambda_n\}$ *and* $\sigma(P^T M^{-1} PA + QA) = \{\mu_1, \ldots, \mu_n\}$ *respectively. Then, the eigenvalues within these spectra can be ordered such that the following statements hold:*

- $\lambda_i = 0$ *and* $\mu_i = 1$, *for* $i = 1, \ldots, k$;
- $\lambda_i = \mu_i$, *for* $i = k+1, \ldots, n$.

Due to Theorem 3.2, both DEF1 and BNN provide almost the same spectra with the same clustering. If $1 \in [\mu_{k+1}, \mu_n]$ then the condition numbers of BNN and DEF1 are identical. On the other hand, if $1 \notin [\mu_{k+1}, \mu_n]$, then DEF1 has a more favorable condition number compared to BNN, see also [20].

THEOREM 3.3. *Let the spectrum of DEF1, DEF2, R-BNN1 or R-BNN2 be given by* $\{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\}$, *satisfying* $\lambda_{k+1} \leq \lambda_{k+2} \leq \ldots \leq \lambda_n$. *Let the spectrum of BNN, A-DEF1 or A-DEF2 be* $\{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$, *with* $\mu_{k+1} \leq \mu_{k+2} \leq \ldots \leq \mu_n$. *Then,* $\lambda_i = \mu_i$ *for all* $i = k+1, \ldots, n$.

Theorem 3.3 provides a strong connection between the two classes as given in Theorem 3.1. It follows that all projection methods have almost the same clustering of eigenvalues. Therefore, we expect that the convergence of all methods will be similar, see Section 4.2 for the verification. Moreover, the zeros in the spectrum of the first class (DEF1, DEF2, R-BNN1 and R-BNN2) might become nearly zero, due to roundoff errors or the approximate solution of $Ey = v$ in the operator. This gives an unfavorable spectrum, resulting in slow convergence of the method. This phenomenon does not appear in the case of BNN, A-DEF1 or A-DEF2. Small perturbations in those projection methods lead to small changes in their spectra and condition numbers. This will be further illustrated in Section 4.3 and 4.4.

3.2. Equivalences between the Projection Methods. It can be shown that some steps in the BNN implementation can be reduced if a special starting vector is used, so that BNN can be transformed into a cheaper operator. Moreover, it can be proven that DEF1 and BNN basically produce the same iterates. These results are collected in the following theorem.

THEOREM 3.4. *Let* $\bar{x} \in \mathbb{R}^n$ *be an arbitrary vector. The following methods produce exactly the same iterates,* x_{j+1}, *in exact arithmetic:*

- *BNN with* $\mathcal{V}_{start} = Qb + P^T \bar{x}$;
- *DEF2, A-DEF2, R-BNN1 and R-BNN2 (with* $\mathcal{V}_{start} = Qb + P^T \bar{x}$*);*
- *DEF1 (with* $\mathcal{V}_{start} = \bar{x}$*) whose iterates are based on* $Qb + P^T x_{j+1}$.

As a result of Theorem 3.4, BNN is equivalent to R-BNN1, R-BNN2, A-DEF2 DEF2, if $\mathcal{V}_{\text{start}} = Qb + P^T \bar{x}$ is used. They even produce the same iterates as DEF1, if its iterates, x_{j+1}, are transformed into $Qb + P^T x_{j+1}$. In Section 4.2, it will be shown numerically that these methods indeed lead to almost identical results.

Another consequence of Theorem 3.4 is that the corresponding operators of DEF2, A-DEF2, R-BNN1 and R-BNN2 are all appropriate if $\mathcal{V}_{\text{start}} = Qb + P^T \bar{x}$ is used, although these operators are not symmetric. This also holds for AD and DEF1, see [42]. A-DEF1 is the only method which cannot be based on an SPD operator, and which can also not be decomposed or transformed into an SPD operator. Hence, it cannot be guaranteed that A-DEF1 always works. Moreover, note that the results as presented in Theorem 3.4 might not be valid anymore in the computations, if the rounding errors are too large. Therefore, although BNN, DEF2, A-DEF2, R-BNN1 and R-BNN2 give exactly the same iterates, all involved projection methods except BNN may lead to inaccurate solutions and may suffer from a lack of robustness in numerical experiments, see Section 4.3.

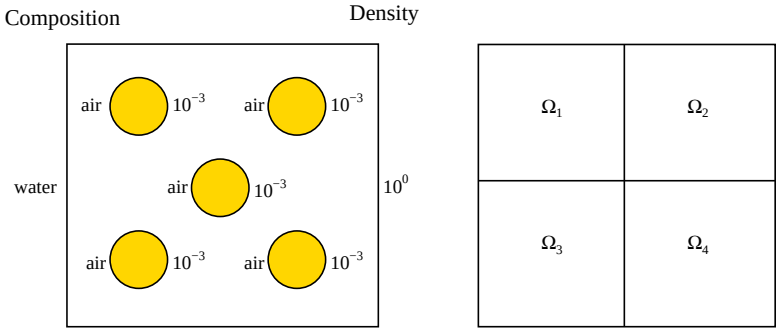
4. Numerical Comparison. In [30], an extensive numerical comparison of the projection methods are presented for the Laplace problem, porous media flows and bubbly flows. In this paper, we restrict ourselves to the latter application. Recently, we have applied DEF1 successfully to bubbly flows [31, 34], where theory has been developed to deal with the singularity of A and the varying density fields. Stationary and time-dependent bubbly flows on highly refined grids are considered to show the effectiveness of DEF1. For complex geometries of the density field and highly resolved grids, it is significantly faster

than PREC. In this section, we will investigate the performance of DEF1 and the other projection methods with respect to the convergence behavior. We remark that the choice of Z, M and the solver for $Ey = v$, are chosen the same for each projection method, which allows us to compare these methods fairly. However, in practice, the projection methods come from different fields, where typical optimal choices associated with deflation, MG or DDM are made for these parameters, see Section 5.

4.1. Setup of the Experiments. We consider the 2-D pressure Poisson equation with a discontinuous coefficient,

$$-\nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla p(\mathbf{x}) \right) = \mathbf{0}, \quad \mathbf{x} = (x, y) \in \Omega = (0, 1)^2, \tag{4.1}$$

where p denotes the pressure, and density ρ is a piecewise-constant function. After applying a second-order finite-difference scheme to (4.1) on a uniform Cartesian grid, this results in our main linear system, $Ax = b$, with $A \in \mathbb{R}^{n \times n}$. This system is usually the most difficult part to solve for the computations of a bubbly flow simulation. We consider circular air bubbles in Ω filled with water, see Figure 4.1(a). The Neumann boundary conditions are such that mass is conserved, which leads to a singular but compatible linear system. The singularity of A does not lead to any difficulties, neither in the theory (results in Section 3 can be generalized using [31–33]), nor in the experiments (see [30, 31]).



(a) Geometry of the piecewise-constant density.

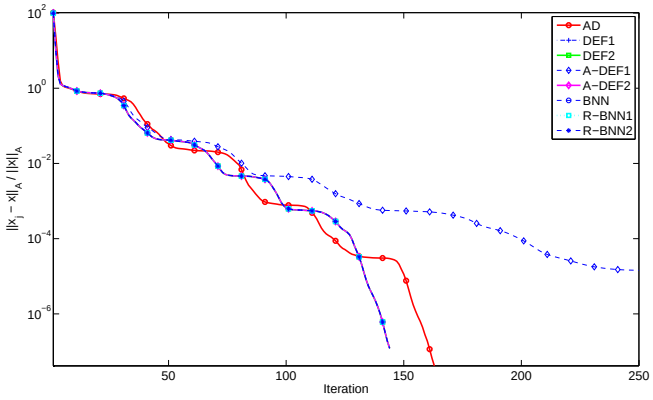
(b) Geometry of subdomains Ω_j with $k = 3$.

FIG. 4.1. *Settings for the bubbly flow.*

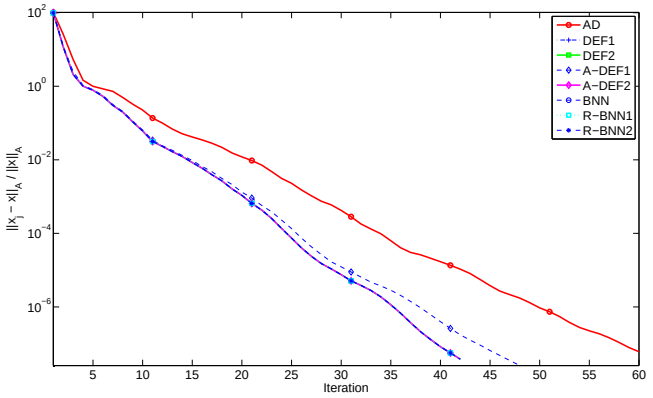
Deflation vectors are used that are based on domain decomposition, where each axis of Ω is divided into k equi-sized 'super-grid points'. We take $k + 1$ identical square subdomains, Ω_j, using the tensor-product mesh generated by these intervals, see Figure 4.1(b). This means that the subdomains are taken independently of the bubbles and a subdomain might cover two different values of the densities. With each subdomain, we associate a single deflation vector, z_j, which takes the value 1 at each grid point included in Ω_j and value 0 at all other grid points. With this choice for the deflation vectors, the deflation matrix is defined as $Z := [z_1 \; z_2 \; \cdots \; z_k]$ in order to preserve an invertible E. We can prove that this choice leads to exactly the same deflation matrix if Z is extended with z_{k+1}, see [31, 34]. Moreover, note that Z is sparse and consists of orthogonal disjoint piecewise-constant vectors, so that it can be stored efficiently in memory and computations with Z can be carried out at low cost [31]. In addition, we can show that the corresponding projection vectors approximate slow-varying eigenvectors corresponding to small eigenvalues [33]. This is even the case for bubbly flow problems with a more complex geometry, provided that a sufficient number of subdomains is taken. This means that our choice of z_j is typical for both DDM and deflation. Typical MG projection vectors could also be used instead, see [6]. Furthermore, we choose as preconditioner, M^{-1}, the Incomplete Cholesky decomposition without fill-in [18], but it seems that other traditional SPD preconditioners could also be used instead, leading to similar results, see [42].

We will start with a numerical experiment using standard parameters, followed by the experiment with inexact E^{-1}. More experiments with severe termination tolerances and perturbed starting vectors can be found in [30], giving similar results as will be shown in Section 4.3. Finally, a 3-D bubbly flow example will be considered to test the best methods for large grids. In addition, the CPU times will be taken into account, due to efficient implementation of the methods. For each test case, each iterative process will be terminated if $||r_{j+1}||_2/||r_0||_2$ falls below $\epsilon = 10^{-10}$.

4.2. Experiment with Normal Parameters. The results of the first experiment can be found in Figure 4.2, where the number of deflation vectors, k, is varied.



(a) $k = 4^2$.



(b) $k = 8^2$.

FIG. 4.2. *Relative errors during the iterative process for the bubbly flow problem problem with* $\epsilon = 10^{-3}, n = 64^2$ *and standard parameters. PREC requires 137 iterations.*

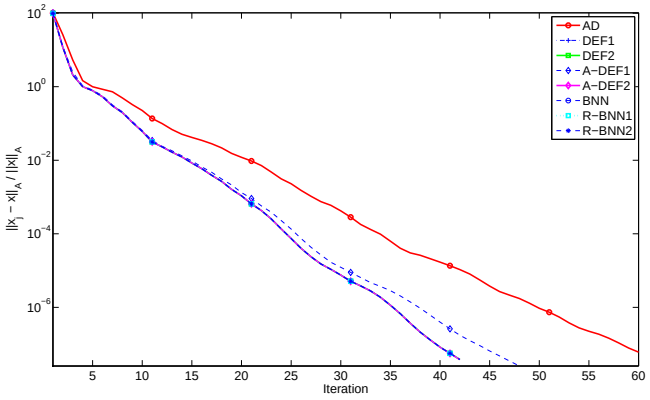
As expected from Section 3, all projection methods perform the same, except for PREC, AD and A-DEF1. A-DEF1 has difficulties to converge for small k. This is not surprising, since it cannot be shown that it is an appropriate preconditioner, as discussed in Section 3.2.

4.3. Experiment with Inaccurate E^{-1}. For problems with a relatively large number of projection vectors, it might be expensive to find an accurate solution of $Ey = v$ at each iteration. Instead, only an approximate solution, $\tilde{y}$, can be determined, using approximate solvers based on SSOR or IC preconditioners, recursive MG methods or nested iterations with a low accuracy. In this case, $\tilde{y}$ can be interpreted as $\widetilde{E}^{-1}v$, where $\widetilde{E}$ is an

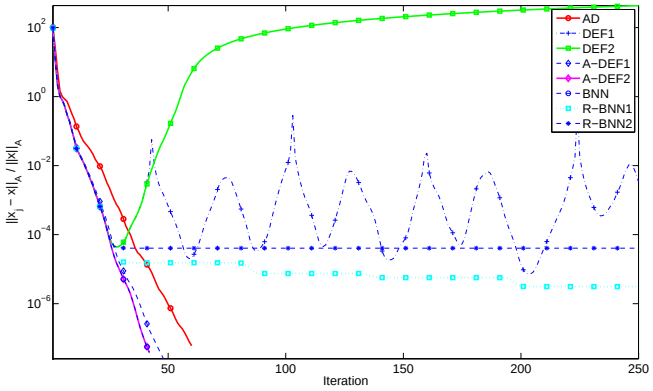
approximation of E. This motivates our next experiment, using $\widetilde{E}^{-1}$ defined as

$$\widetilde{E}^{-1} := (I + \psi R)E^{-1}(I + \psi R), \quad \psi > 0, \tag{4.2}$$

where $R \in \mathbb{R}^{k\times k}$ is a symmetric random matrix with elements from the interval $[-0.5, 0.5]$, see also [19]. Note that the theory, as derived in Section 3.2, does not hold for any $\psi > 0$, but we will see that some of those theoretical results are still valid for relatively large ψ. The results of this experiment can be found in Figure 4.3.



(a) $\psi = 10^{-12}$.



(b) $\psi = 10^{-8}$.

FIG. 4.3. *Relative errors with* $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ *and inaccurate* E^{-1}. *PREC requires 137 iterations.*

From Figure 4.3, we observe that all methods are robust as long as $\psi \leq 10^{-12}$. Obviously, DEF1, DEF2 and also R-BNN1 and R-BNN2 converge badly for $\psi \geq 10^{-8}$, while the most robust methods appear to be AD, BNN, A-DEF1 and A-DEF2.

4.4. More Realistic Experiment with 3-D Bubbly Flows. In this subsection, we perform a more realistic experiment for PREC, DEF1 and A-DEF2, by considering large 3-D bubbly flows (see Figure 4.4) and solving $Ey = v$ iteratively. Eq. (4.2) does not reflect the way that inexact E^{-1} typically enter projection methods; it only provides us with good insights into approximate solves applied to these methods. Since each $Ey = v$ is now solved iteratively, we deal with an inner-outer iteration process. Let us denote the relative stopping tolerance of the outer iteration process by ϵ_{outer} and the one corresponding to $Ey = v$ by ϵ_{inner}. Then, $\epsilon_{\text{inner}} \leq 10^{-2}\epsilon_{\text{outer}}$ turns out to be a safe choice for DEF1, see [31]. Indeed,

$Ey = v$ should then be solved very accurately. Following the discussion of Section 3.2, the expectation is that a larger ϵ_{inner} can be taken in A-DEF2. We will verify this below, see Table 4.1. We observe that it is possible to use the original CG method with inexact preconditioning, since the convergence rate of the outer CG process can be maintained up to a certain accuracy for the inner iterations [7, 9].

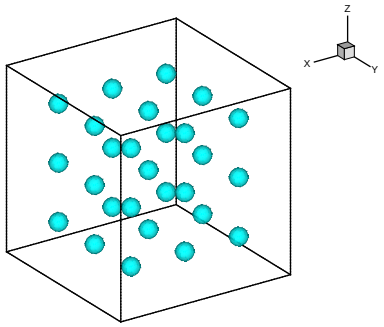


FIG. 4.4. *Geometry of a 3-D bubbly flow test problem with 27 identical air spherical bubbles in water.*

		$k = 15^3$		$k = 25^3$		$k = 50^3$	
Method	ϵ_{inner}	Iter.	CPU	Iter.	CPU	Iter.	CPU
DEF1	10^{-10}	53	24.1	44	25.1	24	82.1
	10^{-8}	N/C	–	N/C	–	N/C	–
A-DEF2	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	–

TABLE 4.1

Results for DEF1 and A-DEF2 to solve $Ax = b$ with $n = 150^3$ and $\epsilon_{outer} = 10^{-8}$, corresponding to the 3-D bubbly flow problem. PREC requires 543 iterations and 177.6 seconds.

DEF1 and A-DEF2 require approximately the same number of iterations for fixed k and they are both faster than PREC, which confirm the theory in Section 3. Furthermore, we expect that A-DEF2 is more robust than DEF1 due to Theorem 3.2. This is indeed the case: when $\epsilon_{\text{inner}} \leq 10^{-8}$, DEF1 did not converge anymore, while A-DEF2 still shows convergence for $\epsilon_{\text{inner}} \leq 10^{-4}$. Apparently, DEF1 can deal with nearly zero eigenvalues as long as they are very small, so that they are treated as zero eigenvalues by the method. We notice that the benefit of using a larger ϵ_{inner} in A-DEF2 can be huge for large k. Additionally, there is an optimum considering the computing time for specific k and corresponding ϵ_{inner}. For DEF1, this is achieved for $k = 20^3$ and $\epsilon_{\text{inner}} = 10^{-10}$, whereas $k = 20^3$ and $\epsilon_{\text{inner}} = 10^{-4}$ are the optimal values in the case of A-DEF2. However, the differences in computing time between the optimal A-DEF2 and DEF1 are small. A-DEF2 could be further improved using multi-level ideas [6].

The explanation that A-DEF2 does not work for $\epsilon_{\text{inner}} \leq 10^{-4}$ is twofold. On the one hand, the CG algorithm can not 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 [23] instead of CG, but it appears that this does not lead to better results [32]. On the other hand, due to large perturbations of E^{-1}, nearly-zero eigenvalues in the spectrum might appear, so that the effectiveness of the projector can be lost.

5. Conclusions and Current Research. Several abstract projection methods have been considered, coming from the fields of deflation, domain decomposition and multigrid. A comparison of these methods has been carried out by investigating their theoretical and numerical aspects. Based on these methods, a generalized projected CG method could be formulated.

It has been shown that all projection methods, except for PREC and AD, have comparable eigenvalue distributions. Two strongly connected classes of projection methods could be distinguished, each having the same spectral properties. The first class consists of DEF1, DEF2, R-BNN1 and R-BNN2, and the second class includes BNN, A-DEF1 and A-DEF2. In the numerical experiments, it has been verified that these projection methods show approximately the same convergence behavior. It has been shown that, for certain starting vectors, the expensive operator of BNN can be transformed into simpler and cheaper operators, which are used in DEF2, A-DEF2, R-BNN1 and R-BNN2. Hence, some projection methods of the two spectral classes are mathematically equivalent in exact arithmetic. However, these reduced variants including DEF1 do not appear to be robust, except for A-DEF2. By examining all theoretical and numerical aspects, we conclude that A-DEF2 is the fastest and most robust abstract projection method. Moreover, DEF1 and A-DEF2 are further compared using efficient implementations and applied to realistic bubbly flow problems. A-DEF2 is more robust, but it is currently more expensive than DEF1. The experiments emphasize that both methods are very effective solvers for the pressure equation derived from bubbly flow problems.

This paper has focused on the abstract versions of projection methods. Presently, we are also comparing a wide range of projection methods with typical parameters used in deflation, DDM and MG [1]. In addition to DEF1 and A-DEF2 with their typical parameters, some multigrid methods, such as Dendy's Blackbox Multigrid [12], appear to be competitive solvers for a series of huge stationary problems and time-dependent simulations in three dimensions.

REFERENCES

[1] J. H. Bramble, J. E. Pasciak, and A.H. Schatz. The construction of preconditioners for elliptic problems by substructuring. I. *Math. Comp.*, 47(175):103–134, 1986.

[2] A. Brandt. Multi–level adaptive solutions to boundary–value problems. *Math. Comp.*, 31:333–390, 1977.

[3] B. Bunner and G. Tryggvason. Dynamics of homogeneous bubbly flows. part 1. rise velocity and microstructure of the bubbles. *J. Fluid Mech.*, 466:17–52, 2002.

[4] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S.F. McCormick, G.N. Miranda, and J.W. Ruge. Robustness and scalability of algebraic multigrid. *SIAM J. Sci. Comp.*, 21:1886–1908, 2000.

[5] H. Ding, P. D. M. Spelt, and C. Shu. Diffuse interface model for incompressible two-phase flows with large density ratios. *J. Comput. Phys.*, 226(2):2078–2095, 2007.

[6] Y. A. Erlangga and R. Nabben. Multilevel projection-based nested Krylov iteration for boundary value problems. *SIAM J. Sci. Comput.*, 2007. To appear.

[7] G. H. Golub and M. L. Overton. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. *Numer. Math.*, 53(5):571–593, 1988.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[9] G. H. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comput.*, 21(4):1305–1320, 2000.

[10] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[11] J. Hua and J. Lou. Numerical simulation of bubble rising in viscous liquid. *J. Comput. Phys.*, 222(2):769–795, 2007.

[12] Jr. J. E. Dendy. Black box multigrid. *J. Comput. Phys.*, 48(3):366–386, 1982.

[13] L. Y. Kolotilina. Twofold deflation preconditioning of linear algebraic systems. I. Theory. *J. Math. Sci.*, 89:1652–1689, 1998.

[14] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[1] Joint work with Scott MacLachlan (Tufts University).

[15] J. Mandel. Hybrid domain decomposition with unstructured subdomains. In A. Quarteroni, editor, *Sixth Conference on Domain Decomposition Methods for Partial Differential Equations*, Providence, RI, USA, 1993. AMS. Como, Italy, June 15–19, 1992.

[16] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comput.*, 65(216):1387–1401, 1996.

[17] E. Marchandise, P. Geuzaine, N. Chevaugeon, and J. Remacle. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *J. Comput. Phys.*, 225(1):949–974, 2007.

[18] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31(137):148–162, 1977.

[19] R. Nabben and C. Vuik. A comparison of Deflation and Coarse Grid Correction applied to porous media flow. *SIAM J. Numer. Anal.*, 42:1631–1647, 2004.

[20] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.*, 27:1742–1759, 2006.

[21] R. Nabben and C. Vuik. A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners. *Num. Lin. Alg. Appl.*, 2007. To appear.

[22] R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987.

[23] Y. Notay. Flexible conjugate gradients. *SIAM J. Sci. Comput.*, 22(4):1444–1460, 2000.

[24] J. Mandel P. Vanek and M. Brezina. Algebraic multigrid based on smoothed aggregation for second- and fourth-order problems. *Comput.*, 56:179–196, 1996.

[25] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[26] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h. A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.*, 21(5):1909–1926, 2000.

[27] R. Singh and W. Shyy. Three-dimensional adaptive cartesian grid method with conservative interface restructuring and reconstruction. *J. Comput. Phys.*, 224(1):150–167, 2007.

[28] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, New York, NY, USA, 1996.

[29] M. Sussman, K. M. Smith, M. Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.*, 221(2):469–505, 2007.

[30] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Theoretical and numerical comparison of various projection methods derived from deflation, domain decomposition and multigrid methods. DIAM Report 07-04, Delft University of Technology, Delft, 2007. Submitted to SIAM J. Sci. Comput. (under revision).

[31] J. M. Tang and C. Vuik. Efficient deflation methods applied to 3-D bubbly flow problems. *Elec. Trans. Numer. Anal.*, 26:330–349, 2007.

[32] J. M. Tang and C. Vuik. Fast deflation methods with applications to two-phase flows. *Int. J. Mult. Comp. Eng.*, 2007. To appear.

[33] J. M. Tang and C. Vuik. New variants of deflation techniques for bubbly flow problems. *J. Numer. Anal. Indust. Appl. Math.*, 2(3-4):227–249, 2007.

[34] J. M. Tang and C. Vuik. On deflation and symmetric positive semi-definite matrices. *J. Comput. Appl. Math.*, 206(2):603–614, 2007.

[35] A. Toselli and O. B. Widlund. *Domain Decomposition: Algorithms and Theory*, volume 34 of *Comp. Math.* Springer, Berlin, 2005.

[36] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2001.

[37] G. Tryggvason, A. Esmaeeli, J. Lu, and S. Biswas. Direct numerical simulations of gas/liquid multiphase flows. *J. Comput. Phys.*, 38(9):660–681, 2006.

[38] S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving Level-Set method for modelling of multi-phase flows. *Int. J. Numer. Methods Fluids*, 47:339–361, 2005.

[39] P. Vanek. Acceleration of convergence of a two-level algorithm by smooth transfer operators. *Appl. Math.*, 37:265–274, 1992.

[40] P. Vanek. Fast multigrid solvers. *Appl. Math.*, 40:1–20, 1995.

[41] C. Vuik, R. Nabben, and J. M. Tang. Deflation acceleration for domain decomposition preconditioners. In P. Wesseling, C.W. Oosterlee, and P. Hemker, editors, *Proc. 8th European Multigrid Conference, September 27-30, 2005, The Hague, The Netherlands*, Delft, 2006. Delft University of Technology.

[42] C. Vuik, A. Segal, and J. A. Meijerink. 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.

[43] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992. Corrected Reprint. Philadelphia: R.T. Edwards, Inc., 2004.