DELFT UNIVERSITY OF TECHNOLOGY

REPORT 06-14

New Variants of Deflation Techniques for Bubbly Flow Problems

J.M. Tang, C. Vuik

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2006

NEW VARIANTS OF DEFLATION TECHNIQUES FOR BUBBLY FLOW PROBLEMS

J.M. TANG AND C. VUIK

Abstract. For various applications, it is well-known that deflated ICCG is an efficient method for solving linear systems with invertible and singular coefficient matrix. This deflated ICCG with subdomain deflation vectors is used by us to solve linear systems with singular coefficient matrix, arising from a discretization of the Poisson equation with Neumann boundary conditions and discontinuous coefficients. However, we have not explained the fact that the use of subdomain deflation vectors is effective. In this paper, we investigate this by doing some perturbed eigenvector analysis and we make it plausible that the use of subdomain deflation vectors is beneficial for the number of iterations and the amount of computational time.

Moreover, we introduce new variants of the deflation technique which can deal with bubbly flow problems and with two-phase flow problems in general. The most simple variant is the so-called levelset deflation technique which chooses deflation vectors corresponding to the bubbles in the domain. Another variant is to combine the subdomain and levelset deflation technique which makes profit of the advantages of the original subdomain deflation and the levelset deflation. Numerical experiments show the good performance of these new deflation variants.

Finally, the undeflated and deflated MICCG and RICCG methods are also investigated, which apply the modified and relaxed IC preconditioners instead of the original IC preconditioner, respectively. We show that the extension of the deflation technique to these methods is only beneficial for test problems with smooth coefficients. In our bubbly flow problems, the original ICCG and DICCG appear to be the most efficient methods.

Keywords. deflation, conjugate gradient method, preconditioning, Poisson equation, symmetric positive semi-definite matrices, bubbly flow problems.
AMS subject classifications. 65F10, 65F50, 65N22.

Date: October 20, 2006.

Contents

1. Introduction

Simulating two-phase flows and in particular bubbly flows is a very popular topic in CFD. These bubbly flows are governed by the Navier-Stokes equations. In many popular operator splitting formulations for these equations, solving the linear system coming from the Poisson equation with discontinuous coefficients takes the most computational time, despite of its elliptic nature. ICCG is widely used for this purpose, but for complex bubbly flows this method shows slow convergence. As alternative for ICCG, we apply a deflated variant of ICCG which is called DICCG. This method incorporates the eigenmodes corresponding to the components which caused the slow convergence of ICCG.

Figure 1. An example of a bubbly flow problem: domain of water with a lot of bubbles.

Recently, DICCG applied on bubbly flows has been studied by the authors [11, 12, 15]. In [11], a literature overview has been given with respect to bubbly flow problems and various deflation variants which may be applied on these problems. In [12], theoretical considerations of DICCG with subdomain deflation vectors are given with respect to the singularity of the linear system. Moreover, some 3-D numerical experiments have been performed in that paper, where ICCG and DICCG have been compared by considering the number of iterations required for convergence to the solution. Furthermore, in [15] some implementation issues of DICCG have been discussed and some more considerations about the singularity are made. In addition, efficient methods to solve coarse linear systems within DICCG have been given including some theoretical analysis in order to compare these methods. Finally, in the same paper it has been shown that the proposed DICCG approaches are very efficient by using numerical experiments. Compared to ICCG, DICCG decreased significantly the number of iterations and the computational time as well, which are required for solving Poisson equation in applications of 2-D and 3-D bubbly flows.

However, in [11, 12, 15] it has not been shown why the approach of DICCG with the subdomain deflation vectors is successful. We have assumed that the small eigenvalues of the preconditioned coefficient matrix have been projected to zero, leading to fast convergence of DICCG. In this paper, we show that this is indeed the case, by doing some spectral analysis with 1-D and 2-D numerical experiments.

Furthermore, the subdomain deflation vectors are chosen in such a way that they do not depend on the geometry of the bubbly flow. The advantage is that this approach is a black-box method. Independent of the structure of the bubbles, DICCG with the same deflation vectors can be used. The drawback is that relevant information which can improve DICCG significantly has been omitted. Therefore, in this paper we will give two new variants of DICCG, where the deflation vectors associate to the bubbles.

In addition, in previous works we have only considered DICCG with the standard IC preconditioner. There are also other IC-like preconditioners available like modified and relaxed IC preconditioners [1] with different spectral properties. In this paper we will investigate these preconditioners, also in combination with deflation.

The outline of this paper is as follows. Section 2 will be devoted to the problem setting of the bubbly flow problems and to the definition of the test problems. The deflation technique with general deflation vectors is introduced in Section 3. In Section 4, we give the existing eigenvector deflation method. Subsequently, in Section 5, we investigate perturbations in eigenvectors and their consequences to the eigenvector deflation, resulting in the perturbed eigenvector deflation. In Section 6 we introduce the so-called levelset deflation approach, where the choice of deflation vectors is strongly related to the bubbles. Related to the levelset deflation, we obtain the original subdomain deflation method as used in [11, 12, 15] which will shortly be discussed in Section 7. Next, Section 8 is devoted to the so-called levelset-subdomain deflation approach, which combines both the original subdomain and the new levelset deflation approaches. Subsequently, in Section 9 we deal with various IC-like preconditioners in combination with deflation. Some numerical experiments will be done in Section 10, where the various variants of deflation will be compared. Finally, the conclusions will be drawn in Section 11.

2. Definition of the Problem

The problem setting of the bubbly flows and some test problems will be defined in this section.

2.1. Problem Setting.

We consider the singular SPSD (symmetric and positive semi-definite) linear system

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

The linear system (1) is derived after a second-order finite-difference discretization of the 1-D, 2-D or 3-D Poisson equation with Neumann boundary conditions, which is

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

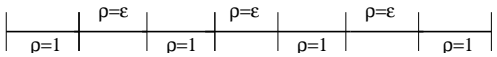
where $p, \rho, \mathbf{x}$ and $\mathbf{n}$ denote the pressure, density, spatial coordinates and the unit normal vector to the boundary $\partial\Omega$, respectively. In the 2-D case, domain Ω is chosen to be a unit square. We apply the computations on a uniform Cartesian grid, so that $n = n_x n_y$ in 2-D, where n_x and n_y are the grid sizes in each spatial direction. Furthermore, we consider two-phase bubbly flows with for instance air and water. In this case, ρ is piecewise constant with a relatively high contrast:

$$\rho = \begin{cases} \rho_0 = 1, & \mathbf{x} \in \Lambda_0, \\ \rho_1 = \epsilon \ll 1, & \mathbf{x} \in \Lambda_1. \end{cases} \tag{3}$$

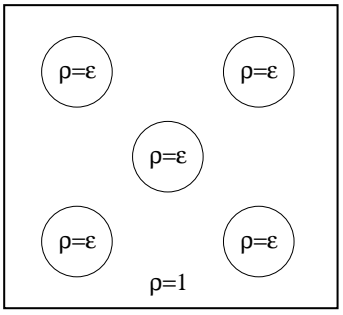
In this paper we use $\epsilon = 10^{-3}$ and $\epsilon = 10^{-6}$.

2.2. Test Problems.

In this paper we consider two test problems: one 1-D test problem with three bubbles (T1) and one 2-D test problem with five bubbles (T2). The geometry of the problems can be found in Figure 2. Moreover, we define P1 and P2 to be the same as T1 and T2 but without the bubbles. In other words, test problems P1 and P2 correspond to the Laplace problems, since the density is constant in the whole domain.



(a) Test problem T1: 1-D problem with 3 bubbles.



(b) Test problem T2: 2-D problem with 5 bubbles.

FIGURE 2. Geometry of the two test problems T1 and T2.

3. Deflation Technique

For a given deflation subspace matrix $Z \in \mathbb{R}^{n\times k}$, the deflation matrix P is defined as follows [6]:

$$P := I - AZE^{-1}Z^T \in \mathbb{R}^{n\times n}, \quad E := Z^TAZ \in \mathbb{R}^{k\times k}. \tag{4}$$

Now, in DICCG we have to solve the resulting linear system

$$M^{-1}PA\tilde{x} = M^{-1}Pb,$$

where M is the Incomplete Cholesky or another preconditioner. From $\tilde{x}$ we can find solution x using the following expression (see e.g. [5]):

$$x = ZE^{-1}Z^Tb + P^T\tilde{x}.$$

More details of DICCG can be found in e.g. [11, 12, 15].

If Z is full-ranked, then Theorem 1 ensures that $M^{-1}PA$ has a more favorable spectrum compared to $M^{-1}A$.

Theorem 1. *Let $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ be the eigenvalues of $M^{-1}A \in \mathbb{R}^{n\times n}$. Let $Z \in \mathbb{R}^{n\times k}$ be a full-ranked deflation subspace matrix. Then,*

$$\textit{spectrum}(M^{-1}PA) = \{0, \ldots, 0, \mu_{k+1}, \ldots, \mu_n\}, \tag{5}$$

with $\lambda_1 \le \mu_i \le \lambda_n$ for all $i > k$.

The theorem is an immediate consequence of Theorem 4.2 of [12]. Obviously, the success of DICCG depends on the choice of the deflation subspace matrix Z. Below, we discuss some choices for Z resulting in different deflation techniques.

4. Eigenvector Deflation

In our bubbly flow applications, we have some small eigenvalues of $\mathcal{O}(\epsilon)$ in the spectrum of $M^{-1}A$ due to the presence of bubbles. The most straightforward choice for the columns of Z is the set of eigenvectors associated to those small eigenvalues. In this case, the corresponding eigenvalues are projected out of the spectrum of $M^{-1}PA$, see Theorem 2.

Theorem 2 (Th. 2.2, [5])**.** *Let $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ be the eigenvalues of $M^{-1}A$. Let $Z = [v_1\ v_2\ \cdots\ v_k]$ be the deflation subspace matrix with v_i eigenvectors of $M^{-1}A$. Then,*

$$\textit{spectrum}(M^{-1}PA) = \{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\}. \tag{6}$$

From Theorem 2 we see that applying eigenvectors as deflation vectors is a good strategy to obtain a better condition number of $M^{-1}A$, which will improve the convergence of the iterative process. Comparing Theorem 2 to Theorem 1, we see that $\mu_i = \lambda_i$ for $i = k+1, \ldots, n$. Hence, $\lambda_{k+1} \leq \mu_i \leq \lambda_n$ for all $i > k$.

5. Perturbed Eigenvector Deflation

In practice it is too expensive or it is even not possible to find exact eigenvectors of $M^{-1}A$. Instead, we could use approximated or perturbed eigenvectors, denoted as $\bar{v}_i$. Of course, we would like to have $\bar{v}_i$ such that the corresponding non-zero eigenvalues $\bar{\lambda}_i$ of $M^{-1}PA$ satisfy

$$\lambda_{k+1} \leq \bar{\lambda}_i \leq \lambda_n \tag{7}$$

for all $i > k$, similar to the eigenvector deflation given in the previous section. However, there are no many strategies known in literature in which way v_i can be perturbed so that it indeed satisfies (7).

We first start with the following definition and assumption.

Definition 1. We define $P_{\tilde{v}_i} := I - A\tilde{v}_i(\tilde{v}_i^T A\tilde{v}_i)^{-1}\tilde{v}_i^T$ and $P_{v_i} = I - Av_i(v_i^T Av_i)^{-1}v_i^T$ where $\tilde{v}_i := v_i + \delta$ with $v_i, \delta \in \mathbb{R}^n$ and fixed $i = 1, 2, \ldots, n$.

Assumption 1. *Let $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of $M^{-1}A \in \mathbb{R}^{n\times n}$ with v_2 to be the eigenvector corresponding to the small eigenvalue λ_2 of $\mathcal{O}(\epsilon)$. Then, we assume that $P_{\tilde{v}_2}A = P_{v_2}A + Q$ with $|Q| = \mathcal{O}(\epsilon)$.*

From numerical experiments it appears that Assumption 1 is fullfilled in our bubbly flow applications. Then the next theorem can be easily proved.

Theorem 3. *If $\delta \in \mathbb{R}^n$ is chosen such that $A\delta = \mathcal{O}(\epsilon)$ is satisfied, then we have*

$$|\lambda_j(P_{\tilde{v}_2}A) - \lambda_j(P_{v_2}A)| \leq \gamma, \quad \gamma = \mathcal{O}(\epsilon). \tag{8}$$

Proof. Since $P_{\tilde{v}_2}A = P_{v_2}A + Q$ with $Q = \mathcal{O}(\epsilon)$ holds, the theorem follows immediately by applying Corollary 8.1.6 of [3]. □

As a consequence, since γ is relatively small, an 'appropriate' perturbation of the eigenvector v_2 does not significantly influence the spectrum of the deflated system which is a favorable result.

Unfortunately, the theorem can not be generalized to the preconditioned deflated systems. It appears from numerical experiments that Inequality (8) does not hold for $M^{-1}P_{\tilde{v}_2}A$ and $M^{-1}P_{v_2}A$ instead of $P_{\tilde{v}_2}A$ and $P_{v_2}A$. In order to show this theoretically, note that the eigenvalues of $M^{-1}P_{v_2}A$ and $M^{-1}P_{\tilde{v}_2}A$ are the same as the eigenvalues of $\widehat{P}_{w_2}\widehat{A}$ and $\widehat{P}_{\tilde{w}_2}\widehat{A}$, respectively, where we apply the transformations $\widehat{A} = M^{-1/2}AM^{-1/2}$, $\widehat{P}_{w_2} = I - \widehat{A}w_2(w_2^T\widehat{A}w_2)^{-1}w_2^T$, $w_2 = M^{1/2}v_2, \tilde{w}_2 = M^{1/2}v_2 = w_2 + \tilde{\delta}$ and $\tilde{\delta} = M^{1/2}\delta$. Then, if $\widetilde{A}\tilde{\delta} = \mathcal{O}(\epsilon)$, Theorem 8 can be generalized. However, after the transformation we have $M^{1/2}A\tilde{\delta} = \mathcal{O}(\epsilon)$, so in general $\widetilde{A}\tilde{\delta} \neq \mathcal{O}(\epsilon)$.

In the next subsection we will do some simple 1-D numerical experiments to get more insight into the consequences to the spectrum of $M^{-1}P_{\bar{v}_i}A$ with varying $\bar{v}_i$ and perturbations δ.

5.1. Perturbing Eigenvectors in Test Problem T1.

We consider Test Problem T1 in a domain with $n = 16$ grid points and we restrict ourselves to the diagonal preconditioner D, since the Incomplete Cholesky preconditioner M is equal to the exact Cholesky decomposition in the 1-D case. Now, we have two eigenvalues λ_2 and λ_3 of order ϵ. The geometry and the two eigenvectors v_2 and v_3 associated to the two smallest non-zero eigenvalues λ_2 and λ_3 can be found in Figure 3.

From the figure we see that the eigenvector and the solution are constant in the bubbles, i.e., in the high-permeable layers. For the eigenvectors, it also holds that the parts connected to the boundaries are also constant.

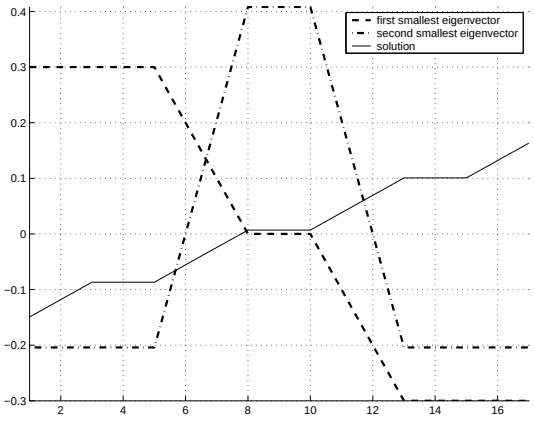


FIGURE 3. Eigenvectors v_2 and v_3 corresponding to the smallest non-zero eigenvalues and solution of $M^{-1}Ax = M^{-1}b$ in Test Problem T1.

5.1.1. *Single Perturbations of the Eigenvectors.* We perturb v_2 so that the perturbed eigenvector is $\bar{v}_2 = v_2 + \alpha e_i$, where e_i is the i-th column of the identity matrix and α is a constant. Then, we consider the second smallest non-zero eigenvalue of $M^{-1}P_{\bar{v}_2}A$. The results are given in Table 1. Similar results hold if we perturb v_3 instead of v_2.

permeability	i	$\alpha = 10^{-1}$	$\alpha = 10^{0}$	$\alpha = 10^{1}$
low	1	2.9×10^{-1}	1.0×10^{-1}	1.3×10^{-3}
	2	2.9×10^{-1}	5.8×10^{-2}	6.5×10^{-4}
high	3	1.6×10^{-5}	2.3×10^{-7}	1.8×10^{-7}
	4	1.1×10^{-5}	1.7×10^{-7}	2.3×10^{-7}
	5	1.6×10^{-5}	2.3×10^{-7}	1.8×10^{-7}
low	6	2.9×10^{-1}	5.9×10^{-2}	6.5×10^{-4}
	7	2.9×10^{-1}	5.9×10^{-2}	6.5×10^{-4}
high	8	1.9×10^{-5}	5.1×10^{-7}	2.1×10^{-7}
	9	1.5×10^{-5}	5.9×10^{-7}	2.9×10^{-7}
	10	1.9×10^{-5}	5.1×10^{-7}	2.1×10^{-7}
low	11	2.9×10^{-1}	5.7×10^{-2}	6.5×10^{-4}
	12	2.9×10^{-1}	1.0×10^{-1}	1.3×10^{-3}
high	13	3.2×10^{-5}	4.6×10^{-7}	2.5×10^{-7}
	14	2.3×10^{-5}	3.5×10^{-7}	2.8×10^{-7}
	15	3.2×10^{-5}	4.6×10^{-7}	2.5×10^{-7}
low	16	2.9×10^{-1}	9.8×10^{-2}	1.2×10^{-3}
	17	2.9×10^{-1}	1.5×10^{-1}	2.4×10^{-3}

TABLE 1. Second smallest non-zero eigenvalue of $M^{-1}P_{\bar{v}_2}A$ where $\bar{v}_2 = v_2 + \alpha e_i$ is the perturbed eigenvector. Second smallest non-zero eigenvalue λ_2 of the original deflation system $M^{-1}P_{v_2}A$ is 2.9×10^{-1}.

From Table 1 we can make the following important observations:

- the high permeable layers (the bubbles) are extremely sensitive for perturbations. A small perturbation in these layers leads immediately to a bad approximation of the eigenvector v_2, since it can be noticed that $\bar{\lambda}_2 \ll \lambda_2$;

- the interfaces between the phases should be treated as the interior of the high permeable layers. Those are also extremely sensitive for perturbations;
- instead, the low permeable layers (region around the bubbles) are relatively insensitive for perturbations, although these perturbations should not be too large ($\alpha \leq 10^0$).

Similar results are known in literature with applications in porous media flows [16]. The main conclusion is that the perturbed vectors can approximate v_2 very well, where only in the low permeable layers can be perturbed.

5.1.2. *Multiple Perturbations of the Eigenvectors.* We investigate whether the high permeable layers (the bubbles) are also extremely sensitive if we perturb the whole part of those layers. We first introduce $e_{i,j}$ which denotes the zero vector with ones from element i to j. For example, we have $e_{3,5} = (0, 0, 1, 1, 1, 0, \ldots, 0)^T$. Then, we perturb v_2 as follows: $\tilde{v}_2 = v_2 + \alpha e_{i,j}$ with various α. Results of the perturbation analysis can be found in Table 2, where the second smallest non-zero eigenvalue of $M^{-1}P_{\tilde{v}_2}A$ is given for a couple of test cases with a fixed left point i and a varying right point j.

$e_{i,j}$	permeability of j	$\alpha = 10^{-1}$	$\alpha = 10^{1}$	$\alpha = 10^{3}$
$e_{3,3}$	high	8.8×10^{-6}	2.7×10^{-7}	2.6×10^{-7}
$e_{3,4}$		1.0×10^{-5}	4.6×10^{-7}	4.4×10^{-7}
$e_{3,5}$	low	2.6×10^{-1}	6.8×10^{-2}	6.5×10^{-2}
$e_{3,6}$		2.9×10^{-1}	7.8×10^{-2}	7.5×10^{-2}
$e_{3,7}$		2.6×10^{-1}	6.8×10^{-2}	6.5×10^{-2}
$e_{3,8}$	high	1.1×10^{-5}	4.4×10^{-7}	4.2×10^{-7}
$e_{3,9}$		1.1×10^{-5}	4.4×10^{-7}	4.2×10^{-7}
$e_{3,10}$	low	2.6×10^{-1}	6.8×10^{-2}	6.5×10^{-2}
$e_{3,11}$		2.7×10^{-1}	7.8×10^{-2}	7.5×10^{-2}
$e_{3,12}$		2.6×10^{-1}	6.8×10^{-2}	6.5×10^{-2}
$e_{3,13}$	high	1.0×10^{-5}	4.6×10^{-7}	4.4×10^{-7}
$e_{3,14}$		8.8×10^{-6}	2.7×10^{-7}	2.7×10^{-7}
$e_{3,15}$	low	2.3×10^{-1}	1.0×10^{-4}	2.5×10^{-7}
$e_{3,16}$		2.5×10^{-7}	1.5×10^{-4}	2.5×10^{-7}

TABLE 2. Second smallest non-zero eigenvalue λ_2 of $M^{-1}P_{\tilde{v}_2}A$ where $\tilde{v}_2 = v_2 + \alpha e_{i,j}$.

From Table 2 we can make the following important observations:

- we have earlier seen that the high permeable layers are extremely sensitive for single perturbations. However, a perturbation in the whole layer (called a 'multiple perturbation') still lead to a good approximation of the eigenvector v_2 independent of the value of α, since it can be observed that $\tilde{\lambda}_2 \approx \lambda_2$. In other words, the eigenvector is insensitive for a shift of the whole high permeable layer;
- the range of the multiple perturbation can be made larger by taking also elements of the low permeable layers. In this case, we still have that the eigenvector v_2 is well-approximated independent of the value of α, whereas it depends on α for a single perturbation in the low permeable layers.
- The multiple perturbation can cover more than one high permeable layers as long as the layers are captured completely. Note that for small α the multiple perturbation can even cover all high permeable layers.

Similar results as in Table 2 hold if we take $i = 8, 13$ instead of $i = 3$ and also if we investigate v_3 instead of v_2.

In contrast to the results of Table 2, these results are not known in literature with applications in porous media flows. The main conclusion is that the deflation vectors in the perturbed eigenvector deflation can approximate the eigenvectors corresponding to the small eigenvalues, by simply covering one or more bubbles in the 'interior' of those deflation vectors. However, note that in general the perturbed eigenspace (the space of the corresponding deflation vectors) does not resemble the original eigenspace, i.e., $\text{span}\{\tilde{v}_2, \tilde{v}_3\} \not\approx \text{span}\{v_2, v_3\}$. In other words, we have $||M^{-1}Av_i|| = \mathcal{O}(\epsilon)$ whereas in general $||M^{-1}A\tilde{v}_i|| = \mathcal{O}(1)$ for $i = 2, 3$.

5.1.3. *Extended Multiple Perturbations of the Eigenvectors.* In Table 2 we can see that relatively large perturbation α with the end point j in the low permeable layers has a somewhat worse effect on the λ_2, although it is still significantly larger than ϵ. For example, consider the results for perturbation $e_{3,5}$. Then observe that if $\alpha = 10^{-1}$, then $\lambda_2 = \mathcal{O}(10^{-1})$, whereas for larger α we have $\lambda_2 > \mathcal{O}(10^{-1})$ This fact is possibly caused by the fact that the bubbles are larger than depicted in Figure 3. The interfaces are namely not in the grid points, but between the grid points in general. Now, an idea is to assume that the neighbour points are in the interior of the bubbles, in the sense that they should be treated equally as the interior points in the perturbation analysis. We repeat some parts of the experiment as done in the previous subsection. So, instead of investigating $e_{3,j}$, we consider $e_{2,j}$, see Table 3. In contrast to the previous results as mentioned in Table 3, we see that relatively

$e_{i,j}$	$\alpha = 10^{-1}$	$\alpha = 10^{1}$	$\alpha = 10^{3}$
$e_{2,3}$	8.8×10^{-6}	2.7×10^{-7}	2.6×10^{-7}
$e_{2,4}$	1.0×10^{-5}	4.6×10^{-7}	4.4×10^{-7}
$e_{2,5}$	2.9×10^{-1}	1.1×10^{-1}	1.2×10^{-1}
$e_{2,6}$	2.9×10^{-1}	1.5×10^{-1}	1.6×10^{-1}
$e_{2,7}$	2.9×10^{-1}	1.1×10^{-1}	1.3×10^{-1}
$e_{2,8}$	1.1×10^{-5}	4.4×10^{-7}	4.2×10^{-7}
$e_{2,9}$	1.1×10^{-5}	4.4×10^{-7}	4.2×10^{-7}

TABLE 3. Second smallest non-zero eigenvalue λ_2 of $M^{-1}P_{\tilde{v}_2}A$ where $\tilde{v}_2 = v_2 + \alpha e_{i,j}$.

large perturbation α in the high permeable layers has more or less no worse effect on the λ_2. In other words, when we apply multiple perturbations to the eigenvectors, then the neighbour points of the high pemeable layers should be taken into account.

5.2. **Consequences for the Perturbed Eigenvector Deflation.** After doing some simple 1-D experiments, we notice that

- in contrast to high permeable layers, the low permeable layers are insensitive for small single perturbations;
- the high (and also the low) permeable layers are insensitive for all kind of multiple perturbations as long as the perturbation acts on the whole bubble with their neighbour points.

It appears that similar observations also hold for other 1-D, 2-D and 3-D numerical experiments with diagonal preconditioners and with more deflation vectors. The consequences for the choice of the deflation vectors in the perturbed eigenvector deflation are the following:

- parts of the eigenvectors associated to the small eigenvalues lying in the low permeable layers can be chosen arbitrarily;
- whole parts of the eigenvectors associated to the small eigenvalues lying in each of the high permeable layers with each neighbour points can be moved.

It appears that the above consequences is equivalent with Conjecture 1.

Conjecture 1. *Let $\lambda_2, \ldots, \lambda_m = \mathcal{O}(\epsilon)$ correspond to eigenvectors $v_2, \ldots, v_m$ and, moreover let $\lambda_{m+1}, \ldots, \lambda_n = \mathcal{O}(1)$. Additionally, assume that $0 = \mu_1 = \mu_2 \leq \mu_3 \leq \ldots \leq \mu_n$ are the eigenvalues of $M^{-1}P_{\tilde{v}_i}A \in \mathbb{R}^{n\times n}$. Then, for all $i = 2, 3, \ldots, m$ we have*

$$\mu_{m+2} = \gamma, \tag{9}$$

where

$$\gamma = \begin{cases} \mathcal{O}(1), & \text{if } |A\delta| = \mathcal{O}(\epsilon); \\ \mathcal{O}(\epsilon), & \text{if } |A\delta| \neq \mathcal{O}(\epsilon), \end{cases} \tag{10}$$

and for the remaining eigenvalues μ_i it holds

$$\mu_3, \ldots, \mu_{m+1} = \mathcal{O}(\epsilon), \quad \mu_{m+3}, \ldots, \mu_n = \mathcal{O}(1). \tag{11}$$

Note that from the conjecture it follows that the strategy of choosing right deflation vectors for the system $M^{-1}A$ depends on $A\delta$ rather than on $M^{-1}A\delta$. Hence, as a rule of thumb in the perturbed eigenvector deflation technique, appropriate deflation vectors $\tilde{v}_i = v_i + \delta_i$ from eigenvectors v_i and perturbations δ_i with $i = 2, 3, \ldots, m$ can be obtained by choosing δ_i such that $A\delta_i = \mathcal{O}(\epsilon)$.

Finally, we refer to [10] for more literature which is related to this subject. In that paper two-level overlapping domain decomposition preconditioners with coarse spaces obtained have been studied by smoothed aggregation in iterative solvers for finite element discretizations of elliptic problems. Similar observations made in this section have been proven in [10] using functional analysis.

6. Levelset Deflation

To project the smallest eigenvalues out of the spectrum of $M^{-1}A$, we have seen that we do not need exact eigenvectors. In the approach of perturbed eigenvector deflation we use deflation vectors where parts of the eigenvectors associated to the small eigenvalues lying in the low permeable layers can be chosen arbitrarily. To obtain a sparse Z, it is convenient to choose them zero in these parts. Moreover, since whole parts of the eigenvectors associated to the small eigenvalues lying in each of the high permeable layers can be moved and they are constant in these layers, we can move them to exactly one. In this way, we can obtain a deflation subspace Z which is sparse and only consists of the values 0 and 1, while the eigenvectors associated to the small eigenvalues are still well approximated. The new deflation vectors are denoted by w_i. Hence, in this new deflation variant, the idea to obtain a sparse Z is to move the bubbles to exactly one, while the surroundings are taken to be zero. In this case we have a kind of deflation technique with subdomains. Note that in an efficient deflation method, such a subdomain deflation vector may cover more than one bubble, as long as it captures them completely. This is in contrast to the subdomain deflation vectors which do not have to cover all the bubbles. Each bubble which can be covered by w_i leads to a more favorable effective condition number of $M^{-1}PA$ as long as this w_i does not cover all the bubbles or parts of other bubbles. To illustrate this, we consider a 2-D application as drawn in Figure 4, where two choices of subdomains are depicted. In the left subplot, the middle bubble is divided into four subdomains, while in the right subplot each bubble is captured by a subdomain. As a result, all four small eigenvalues of $M^{-1}A$ remain in the spectrum of $M^{-1}PA$ in the left situation, confirming the results obtained in Table 2, whereas three of the four small eigenvalues are eliminated out of spectrum of $M^{-1}PA$ in the right subplot. Note that if there is only one bubble in the domain, then both situations lead to the same results.

If the density field is known, then the new deflation variant can be simply applied by locating the bubbles. In our applications, the levelset approach is adopted to describe the density fields, see e.g. [8, 9]. More information about the levelset methods can be found in for instance [4,7] . For each time step, the levelset function

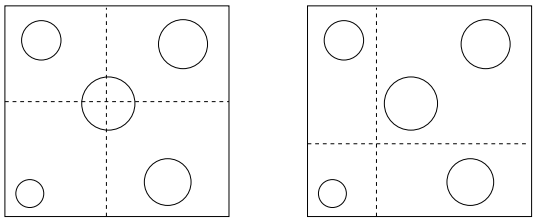


FIGURE 4. 2-D example of a bubbly flow problem where the choice of the subdomains is crucial for an efficient deflation method. Left: wrong choice of subdomains, since the middle bubble is not captured by one subdomain. Right: good choice of subdomains, since each bubble is in the interior of a subdomain.

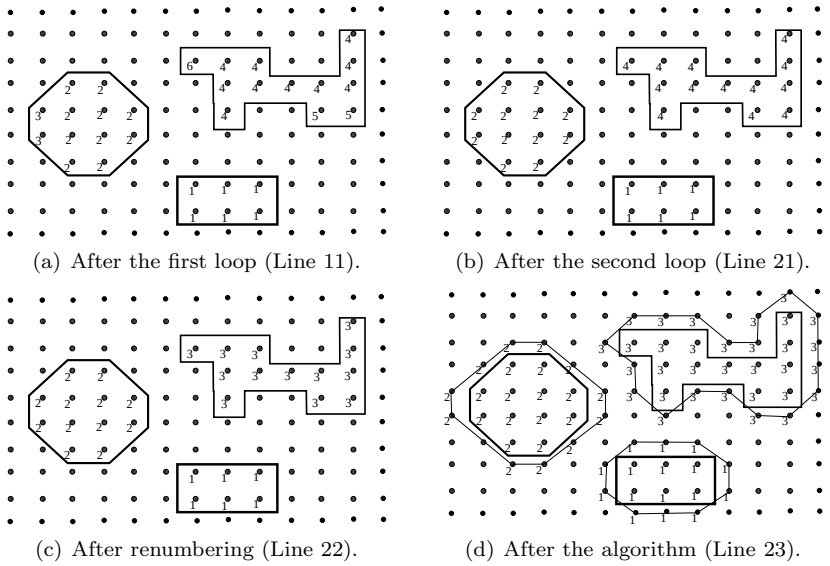
is known in the whole computational domain. Distinction of the bubbles from the levelset function can be done using for example Algorithm 1. Applying Algorithm 1 to a 2-D example with three bubbles can be seen in Figure 5.

Algorithm 1 Algorithm: separating bubbles from the levelset function

```
1: j = 1
2: for grid point 1 to n (from left to right and from bottom to top) do
3:    if grid point is in a bubble then
4:       if left and/or bottom neighbour grid point is not in a bubble then
5:          give j to the grid point
6:          j = j + 1
7:       else
8:          take lowest value of the neighbour(s)
9:       end if
10:   end if
11: end for
12: for grid point n to 1 (from right to the left and from top to bottom) do
13:   if grid point is in a bubble then
14:      if right and/or top neighbour grid point is not in a bubble then
15:         give j to the grid point
16:         j = j + 1
17:      else
18:         take lowest value of the neighbour(s)
19:      end if
20:   end if
21: end for
22: Renumber all grid points with number j
23: Include all neighbour grid points of each bubble j
```

In the algorithm, we need two loops through the domain to separate the bubbles, requiring $\mathcal{O}(n)$ flops. Note further that in the case of deciding whether or not a grid point of its neighbour is in a bubble, we simply look at the sign of the levelset function in that point. If the value is positive the grid point is in the interior of the bubble, if it is negative then it is outside the bubble and otherwise it is on the interface. In this way, it is straightforward to distinguish the bubbles from the levelset function and to obtain a code where each deflation vector corresponds to one separate bubble. This new method is called 'levelset deflation', since the deflation vectors are based on the levelset function.

In summary, each vector of the levelset deflation is sparse and corresponds to one separate bubble associated to the eigenvalues of order $\mathcal{O}(\epsilon)$. These eigenvalues



(a) After the first loop (Line 11). (b) After the second loop (Line 21).

(c) After renumbering (Line 22). (d) After the algorithm (Line 23).

FIGURE 5. 2-D example to illustrate Algorithm 1.

are captured by this deflation method so that we do not have to compute the real (perturbed) eigenvectors.

Finally, we note that if some bubbles in the domain are very close to each other, then some grid points can belong to several bubbles leading to overlap. In other words, the row sum of Z can be higher than one for some rows. This is no restriction of the method. The reverse is true: if one excludes overlap between the bubbles, then the bubbles appear to be approximated badly, leading to slower convergence of the iterative process, see also Subsection 10.4.

7. Subdomain Deflation

In applications where a lot of bubbles are in the domain or where the levelset function is unknown or too complex, we can choose for the original subdomain deflation. The principle is the same as levelset deflation, but now the subdomains are chosen fixed, independent of the locations of the bubbles. In [11,12,15] we used cubes as subdomains in 3-D applications. We can define Z in the following formal way.

Let the computational domain Ω be divided into open (equal) subdomains Ω_j, $j = 1, 2, \ldots, k$, such that $\Omega = \cup_{j=1}^{k} \overline{\Omega}_j$ and $\cap_{j=1}^{k} \Omega_j = \emptyset$. 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 deflation vector z_j as follows:

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

where x_i is a grid point in the discretized domain Ω_h. Then we define

$$Z := [z_1 \; z_2 \; \cdots \; z_k] \in \mathbb{R}^{n \times k}.$$

This implies immediately $Z\mathbf{1}_k = \mathbf{1}_n$, which does not hold for the levelset deflation vectors. Hence, Z consists of disjunct orthogonal piecewise-constant vectors. Note that Z from subdomain deflation is obviously less sparse than matrix Z from the levelset deflation. Moreover, $Z\mathbf{1}_k = \mathbf{1}_n$ is a good property with respect to the implementation and some proofs of theoretical results, but this can give difficulties to approximate the bubbles well leading to a worse effective condition number. This has already been noted in the previous section, where in the levelset deflation

method, some row sums of Z should be larger than one to approximate the bubbles optimal if they are very close to each other.

Subdomain deflation and levelset deflation are related to each other. In subdomain deflation, we can increase k until none of the subdomains consists of more than one part of the bubbles. In this case, the eigenvectors associated to the smallest eigenvalues are well approximated, as in levelset deflation. Moreover, the subdomain deflation vectors can also approximate other eigenvectors corresponding to small eigenvalues of order $\mathcal{O}(1)$ associated to slow eigenmodes. This means that subdomain deflation is more expensive in each iteration compared to levelset deflation due to a less sparse deflation subspace matrix, but it may increase the convergence behavior of the iterative process since more small eigenvalues of both $\mathcal{O}(\epsilon)$ and $\mathcal{O}(1)$ can be projected out of the spectrum.

8. Levelset Subdomain Deflation

For density fields with a complex geometry, it is difficult to capture all parts of the bubbles separately in subdomain deflation. In that case, the smallest eigenvalues can not be projected out of the spectrum. However, this can be realized using levelset deflation with the disadvantage that only the $\mathcal{O}(\epsilon)$ eigenvalues can be captured. In other words, it can be beneficial to combine both deflation technique leading to a new deflation method called 'levelset-subdomain' (LS) deflation. It will require more deflation vectors than the original methods separately, but in the new method we can make profit of the advantages of both deflation methods.

We define Z_s as the deflation subspace matrix corresponding to subdomain deflation and Z_l as the deflation subspace matrix corresponding to the levelset method. The new deflation subspace matrix is denoted as Z_{LS} and the corresponding deflation matrix is denoted as P_{LS}. Next, we define

$$\begin{cases} Z_1 & := & Z_s \cap (\mathbf{1} - \cup Z_l); \\ Z_2 & := & Z_l \cap Z_s. \end{cases} \tag{12}$$

In the above definition, $\cup$ acting on a matrix W means that a new vector is created with the maximum element of each row of W. Moreover, $\cap$ acting on two matrices/vectors V and W means that a new matrix/vector is created with columns equal to all possible non-zero combinations between the columns of V and W, where the elements are multiplied with each other componentwise. The next example will illustrate (12).

Example 1. *Let*

$$Z_s = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T, \quad Z_l = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^T.$$

Then this leads to

$$\cup Z_l = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^T, \quad \boldsymbol{1} - \cup Z_l = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T.$$

This gives us Z_1 and Z_2:

$$Z_1 := Z_s \cap (\boldsymbol{1} - \cup Z_l) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}^T,$$

and

$$Z_2 := Z_l \cap Z_s. = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^T.$$

Hence, Z_1 consists of all original subdomain vectors except for the parts corresponding to the bubbles, whereas Z_2 consists of columns corresponding to the bubbles divided by the original subdomains. Finally, the levelset subdomain deflation subspace matrix is

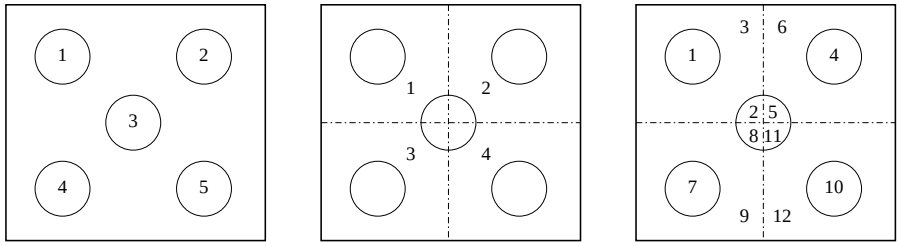
$$Z_{LS} = [Z_1, Z_2].$$

It is not difficult to show that

$$\begin{cases} \text{span } Z_s & \subseteq & \text{span } Z_{LS}; \\ \text{span } Z_l & \subseteq & \text{span } Z_{LS}. \end{cases}$$

Moreover, the row sum of Z_{LS} is equal to one if there is no overlap in the levelset deflation subspace matrix, for each row, i.e., $Z_{LS}\mathbf{1} = \mathbf{1}$, which can be favorable in the implementation of the method and in proving theoretical results as earlier mentioned. Note that if there is overlap in the levelset deflation subspace matrix, then $Z_{LS}\mathbf{1} \neq \mathbf{1}$.

Finally, the new method combines the advantages of both deflation approaches with the only drawback that a larger number of deflation vectors are required compared to one of these separate approaches, see also the example given in Figure 6.



(a) Levelset deflation. (b) Subdomain deflation. (c) Levelset subdomain deflation .

FIGURE 6. 2-D example to illustrate the levelset, subdomain and levelset subdomain deflation techniques.

9. Deflation Technique in combination with other IC-like Preconditioners

In DICCG the deflation method is used in combination with the standard Incomplete Cholesky (IC) preconditioner. In [11,12,15] we have seen that the small eigenvalues with associated slow eigenmodes corresponding to ICCG can be eliminated while the largest eigenvalues are of order 1 so that they should not be eliminated from the spectrum. In this section we investigate the deflation method in combination with other IC-like preconditioners, namely Modified IC (MIC) and Relaxed IC (RIC) resulting in the methods MICCG and RICCG, see also [1]. These methods have spectra of other type, so the question is whether deflation still succeed in combination with these methods.

After the definition of the preconditioners, we will perform some 2-D numerical experiments with both Test Problems P2 and T2 in the next section. In this 2-D case, matrix A consists of 5 diagonals. We assume that

$$A = \begin{bmatrix} a_1 & b_1 & \cdots & c_1 & & & & & \\ b_1 & a_2 & b_2 & & c_2 & & & & \\ \vdots & \ddots & \ddots & \ddots & & \ddots & & & \\ c_1 & & b_m & a_{m+1} & b_{m+1} & & c_{m+1} & & \\ & \ddots & & \ddots & \ddots & \ddots & & \ddots & \end{bmatrix} \in \ltimes \times \ltimes. \tag{13}$$

If A is SPD then the full Cholesky decomposition exists, i.e., we can find a lower triangular matrix L and diagonal matrix D such that $A = LD^{-1}L^T$. However, for large A this decomposition is expensive to compute. In addition, L will become dense due to fill-in, although A is sparse.

9.1. **IC-like Preconditioners.** In the next subsections, we will define some variants of the Cholesky decomposition which can be used as preconditioners. The main property of these preconditioners is that lower triangular matrix L have the same sparsity pattern as A, i.e., the element of L are given by

$$L = \begin{bmatrix} \tilde{d}_1 & & & & \\ \tilde{b}_1 & \tilde{d}_2 & & & \\ \vdots & \ddots & \ddots & & \\ \tilde{c}_1 & & \tilde{b}_m & \tilde{d}_{m+1} & \\ & \ddots & & \ddots & \ddots \end{bmatrix}. \tag{14}$$

9.1.1. *IC preconditioner.* In the standard IC preconditioner we use $M_{IC} = LD^{-1}L^T$ where $A = LD^{-1}L^T + R$ which satisfies the following rules:

- $l_{ij} = 0$ for all (i, j) where $a_{ij} = 0$ with $i > j$;
- $l_{ii} = d_{ii}$;
- $(LD^{-1}L^T)_{ij} = a_{ij}$ for all (i, j) where $a_{ij} \neq 0$ for $i \geq j$.

It appears that although A can be singular, M is always invertible [2]. It can be verified that the elements of L and D can be computed by

$$\left\{ \begin{array}{lcl} \tilde{d}_i & = & a_i - \frac{b_{i-1}^2}{d_{i-1}} - \frac{c_{i-m}^2}{d_{i-m}}; \\ \tilde{b}_i & = & b_i; \\ \tilde{c}_i & = & c_i, \end{array} \right. \tag{15}$$

for all $i = 1, \ldots, n$.

The resulting CG method has been denoted as ICCG and its deflated variant as DICCG.

9.1.2. *MIC preconditioner.* In the modified IC (MIC) preconditioner we use $M_{MIC} = LD^{-1}L^T$ where $A = LD^{-1}L^T + R$ which satisfies the following rules:

- $l_{ij} = 0$ for all (i, j) where $a_{ij} = 0$ with $i > j$;
- $l_{ii} = d_{ii}$;
- row sum $(LD^{-1}L^T)$ = row sum (A) for all rows;
- $(LD^{-1}L^T)_{ij} = a_{ij}$ for all (i, j) where $a_{ij} \neq 0$ for $i > j$.

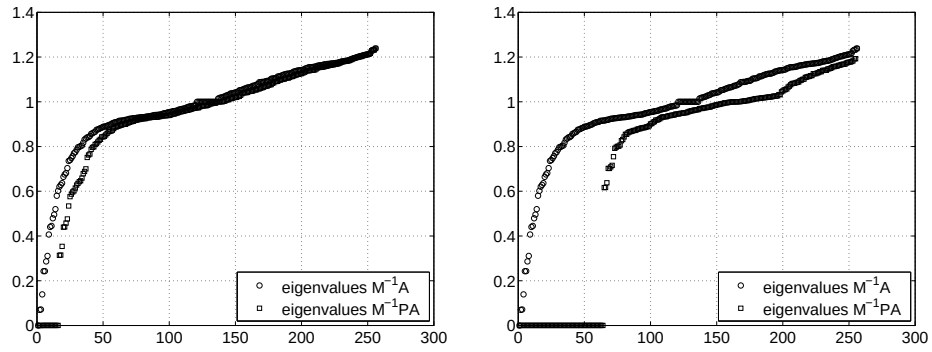
Recall that in our applications the row sum of A is zero for each row. Therefore, A is singular. As a consequence, M is also singular because the row sum of M is zero for each row due to the third rule. In this case, the MIC preconditioner can not be applied. It can be verified that the elements of L and D can be computed by

$$\left\{ \begin{array}{lcl} \tilde{d}_i & = & a_i - (b_{i-1} + c_{i-1})\frac{b_{i-1}}{d_{i-1}} - (b_{i-m} + c_{i-m})\frac{c_{i-m}}{d_{i-m}}; \\ \tilde{b}_i & = & b_i; \\ \tilde{c}_i & = & c_i, \end{array} \right. \tag{16}$$

for all $i = 1, \ldots, n$. The resulting CG method is denoted as MICCG and its deflated variant as DMICCG.

9.1.3. *RIC preconditioner.* In order to combine the advantages of both preconditioners, the relaxed IC (RIC) preconditioner has been proposed, which is an average of the IC and MIC preconditioners. We only give the algorithm to compute $M_{RIC} = LD^{-1}L^T$. For a given average parameter $\alpha \in [0, 1]$, the elements of L and D can be computed by

$$\left\{ \begin{array}{lcl} \tilde{d}_i & = & a_i - (b_{i-1} + \alpha c_{i-1})\frac{b_{i-1}}{d_{i-1}} - (\alpha b_{i-m} + c_{i-m})\frac{c_{i-m}}{d_{i-m}}; \\ \tilde{b}_i & = & b_i; \\ \tilde{c}_i & = & c_i, \end{array} \right. \tag{17}$$



(a) Case with $k = 16$ (15 iterations). (b) Case with $k = 64$ (10 iterations).

FIGURE 7. Eigenvalues of both $M^{-1}A$ and $M^{-1}PA$ with S-DICCG$-k$ for Test Problem P2 with $n_x = n_y = 16$.

for all $i = 1, \ldots, n$. Obviously, if $\alpha = 0$ then we obtain the original IC preconditioner, whereas if $\alpha = 1$ we have the MIC preconditioner. Moreover, we note from experiments that for $\alpha \neq 1$ the RIC preconditioner is invertible if A is singular.

It appears that the choice of $\alpha = 0.95$ is appropriate since it gives a good rate of convergence for a wide range of problems. Finally, the resulting CG method is denoted as RICCG and its deflated variant as DRICCG.

10. Numerical Experiments

In this section we perform some numerical experiments to test the levelset, subdomain en levelset-subdomain deflation in combination with ICCG, i.e., we apply

- L-DICCG$-k$: DICCG with k levelset deflation vectors;
- S-DICCG$-k$: DICCG with k subdomain deflation vectors;
- LS-DICCG$-k$: DICCG with k levelset-subdomain deflation vectors.

We do not consider eigenvector and perturbed eigenvector deflation since they are too expensive in use.

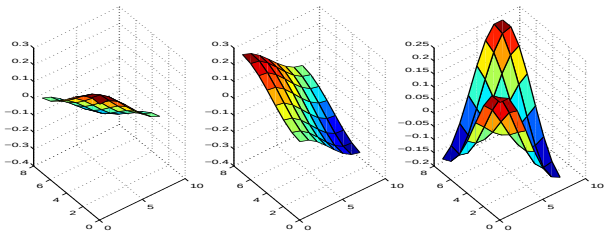
After discussing the results for the Poisson equation without bubbles (Test Problem P2), we consider the same equation with bubbles (Test Problem T2). Recall that the test problems used in this section have already been defined in Subsection 2.2.

During the experiments in the next subsection we take the jump $\epsilon = 10^{-6}$ and grid sizes $n_x = n_y = 16$. It appears that similar numerical results hold for other choices of jumps and grid sizes, see also the last subsection. In addition, similar results can be seen by using the diagonal instead of the IC preconditioner. Finally, the iterative processes converge if the relative update residuals are below the tolerance 10^{-7}.

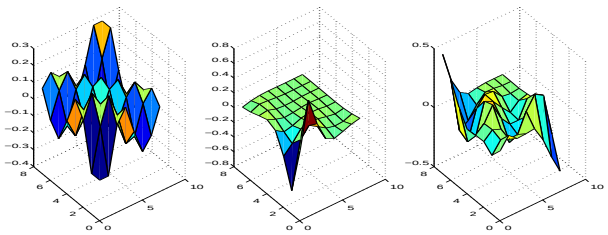
10.1. **Results Various Deflation Techniques for Test Problem P2.** In the case of Test Problem P2, only S-DICCG$-k$ can be applied while the other deflation variants are not defined since there are no bubbles in the domain.

It appears that ICCG converges within 23 iterations, whereas S-DICCG-16 and S-DICCG-64 require only 15 and 10 iterations, respectively. The corresponding eigenvalues of $M^{-1}A$ and $M^{-1}PA$ can be found in Figure 7.

From both subfigures, we notice that the smallest eigenvalues of $M^{-1}A$ are projected out of the spectrum of $M^{-1}PA$, while the largest eigenvalues remain in the spectrum. The more deflation vectors, the more small eigenvalues are eliminated. To explain the first observation, three eigenvectors associated to the small



(a) Three eigenvectors associated to small eigenvalues.



(b) Three eigenvectors associated to random non-small eigenvalues.

FIGURE 8. Eigenvectors associated to the eigenvalues of $M^{-1}A$ for Test Problem P2.

eigenvalues are depicted in Subfigure 8(a). From this subfigure we see that these eigenvectors are relatively smooth. In some sense, the subdomain deflation vectors approximate these smooth functions very well. Other eigenvectors corresponding to larger eigenvalues of $M^{-1}A$ do not have a smooth behavior, see also Subfigure 8(b). Therefore, these are difficult to approximate using the subdomain deflation vectors. In the past we have seen that S-DICCG$-k$ works very well in this case. This was suprising because there are no very small eigenvalues corresponding to bubbles. Now we know that the subdomain deflation vectors approach other eigenvectors of the preconditioned system $M^{-1}A$ very well, resulting in fast convergence.

10.2. **Results Various Deflation Techniques for Test Problem T2.** For Test Problem T2, the location of the bubbles can be found in Figure 9. In this case, ICCG converges within 39 iterations.

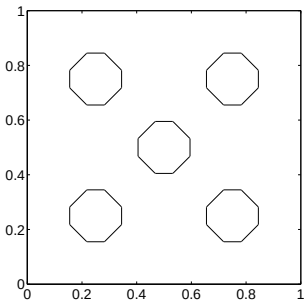
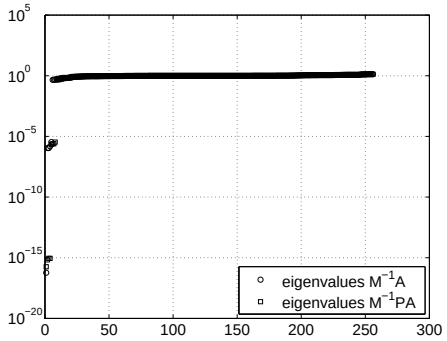
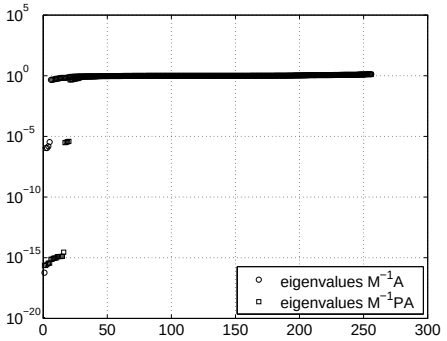


FIGURE 9. Geometry of Test Problem T2 for the case of $n_x = n_y = 16$.

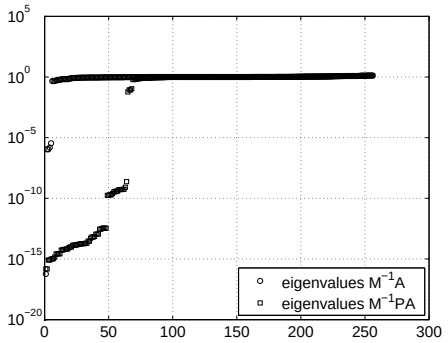
10.2.1. *Subdomain Deflation.* From the experiments, we see that S-DICCG$-k$ with $k = 4, 16$ converges within 36 and 37 iterations, respectively. There is almost no benefit compared to ICCG. To investigate this, we consider the spectra of the deflated preconditioned system, see Figure 10.



(a) Case with $k = 4$ (37 iterations).



(b) Case with $k = 16$ (36 iterations).



(c) Case with $k = 64$ (19 iterations).

FIGURE 10. Eigenvalues of $M^{-1}PA$ with S-DICCG$-k$ for Test Problem T2 with $n_x = n_y = 16$.

Obviously, none of the smallest eigenvalues of $M^{-1}A$ is eliminated after deflation for both $k = 4$ and $k = 16$. In Subfigures 10(a) and 10(b), we see that the smallest eigenvalues of order 10^{-6} remain in the spectrum after deflation. Only for larger k, each part of the bubbles will be captured by an independent subdomain and therefore, the smallest eigenvalues can then be eliminated. For example, S-DICCG-64 converges within 19 iterations since in this case the small eigenvalues are projected out, see Subfigure 10(c). Hence, in contrast to the cases of $k \leq 16$, the small eigenvalues of order 10^{-6} are eliminated for the case of $k = 64$.

Finally, notice from Figure 10 that in the cases $k = 4$ and $k = 16$ the zero eigenvalues are indicated as values of order 10^{-15} in MATLAB. However, in the case of $k = 64$, these zero eigenvalues are of order 10^{-10}, but fortunately these values are known as zeros during the iterative process since the stopping tolerance is of order 10^{-7}.

10.2.2. *Levelset Deflation.* After discussing S-DICCG$-k$, we consider L-DICCG$-k$ where we can note that it requires only $k = 5$ vectors in Test Problem T2.

L-DICCG-5 converges within 17 iterations. We refer to Figure 11 which gives the spectra of the corresponding deflated and undeflated systems. We see that all smallest eigenvalues are eliminated out of the spectrum after deflation, while all other eigenvalues are still of order 1. It appears that in the ideal case, using exact eigenvectors to eliminate the smallest eigenvalues of order 10^{-6}, the deflated ICCG (i.e., the eigenvector deflation variant) converges within 13 iterations. This means

that the eigenvalues not corresponding to the bubbles become relatively favorable after levelset deflation. [1]

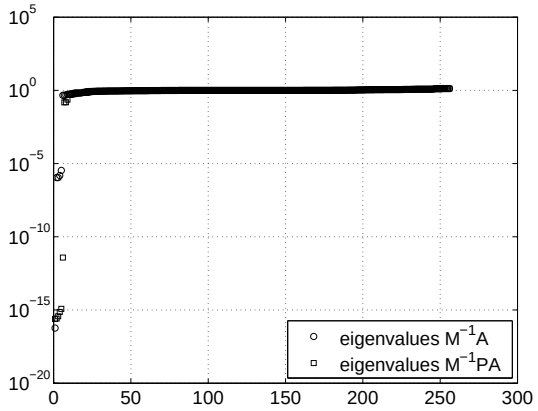
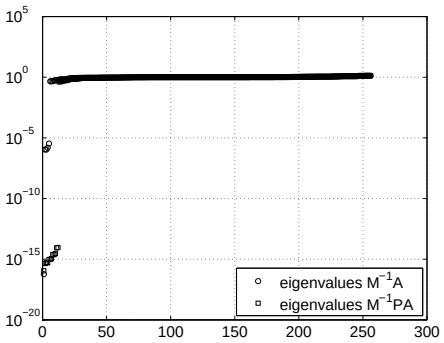


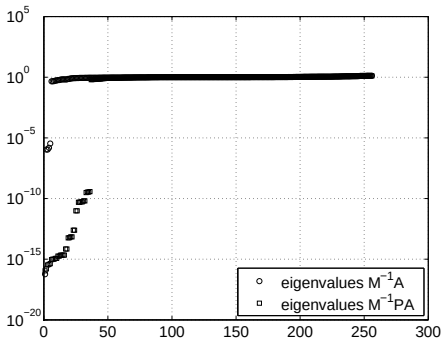
FIGURE 11. Eigenvalues of both $M^{-1}A$ and $M^{-1}PA$ with L-DICCG-5 for Test Problem T2 with $n_x = n_y = 16$.

10.2.3. *Levelset Subdomain Deflation.* After giving the results of S-DICCG$-k$ and L-DICCG$-k$, we apply the hybrid deflation method LS-DICCG$-k$ to Test Problem T2.

From experiments, we see that LS-DICCG-12 (cf. Figure 6(c)) converges within 14 iterations and LS-DICCG-36 converges within 10 iterations. The spectra can be found in Figure 12. Since the original S-DICCG-64 needs 19 iterations and L-DICCG-5 needs 17 iterations, it is clear that the LS-DICCG$-k$ can improve the speed of convergence, although the differences are relatively small.



(a) Case with 4 subdomains ($k = 12$; 14 iterations).



(b) Case with 16 subdomains ($k = 36$; 10 iterations).

FIGURE 12. Eigenvalues of both $M^{-1}A$ and $M^{-1}PA$ with LS-DICCG$-k$ for Test Problem T2 with $n_x = n_y = 16$.

[1] Note that if the neighbour points are not included in the deflation vectors then it requires 28 iteration, since it appears that it gives new small eigenvalues of a order between ϵ and 1. In this case, the smallest eigenvalues of order 10^{-6} are eliminated of $M^{-1}A$ but we get new eigenvalues of order 10^{-2} and 10^{-3}.

10.3. **Results Various Deflation Techniques for Varying Grid Sizes, Densities and Number of Bubbles.** The results of Test Problem T2 of the previous subsection, including the computational time in seconds are summarized in Table 4. Moreover, results with other grid sizes are also given.

		$n = 16^2$		$n = 32^2$		$n = 64^2$	
Deflation Method	k	# It.	CPU	# It.	CPU	# It.	CPU
ICCG	–	39	0.04	82	0.53	159	10.92
S-DICCG$-k$	4	37	0.12	80	0.67	194	14.01
	16	36	0.07	97	0.80	193	13.82
	64	19	0.11	16	0.20	26	2.14
L-DICCG$-k$	5	17	0.09	37	0.37	75	6.17
LS-DICCG$-k$	12	14	0.07	30	0.29	54	4.08
	36	10	0.08	21	0.32	40	3.05
	84	–	–	15	0.20	25	2.05

TABLE 4. Results of Test Problem T2 for various n.

From Table 4, one observes that S-DICCG$-k$ performs bad for $k = 4, 16$ in all test cases. However, for $k = 64$ the method is very efficient, since in this case, each subdomain consists of maximum one part of a bubble. Moreover, L-DICCG$-k$ reduces the number of iterations significantly. It is an efficient method since it requires only five deflation vectors. Finally, LS-DICCG$-k$ performs very well, but for sufficiently large k we notice that S-DICCG$-k$ and LS-DICCG$-k$ are comparable.

Notice further that in the case of $n = 64^2$, ICCG requires significantly less iterations compared to S-DICCG-4 and S-DICCG-16. To investigate this, we consider the exact residuals and the relative update preconditioned residuals of ICCG and the relative update preconditioned-deflated residuals of S-DICCG-16, see Figure 13. Note that all relative residuals start from 1, except for the relative update preconditioned-deflated residuals, since these are defined by

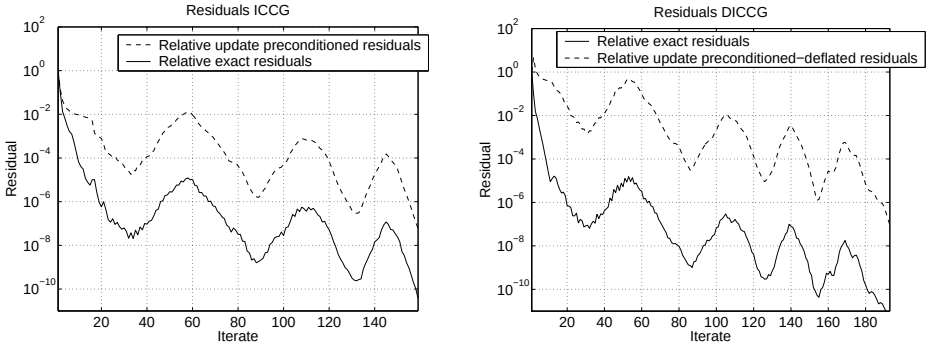
$$\frac{||M^{-1}P(b - A\tilde{x}_k)||_2}{||M^{-1}(b - Ax_0)||_2},$$

where one can observe that there is no deflation matrix P in the denominator.

Moreover, from Figure 13, it can be noted that when the relative update residuals are below 10^{-7}, the relative exact residuals of ICCG are already of order 10^{-10}. In the right subplot we see that the residuals require one extra 'bump' to converge. In addition, we see that the accuracy of order 10^{-10} has earlier been reached in the case of the exact residuals of S-ICCG-16 (around 155 iterations), but due to the bump S-DICCG-16 takes significantly more iterations to converge. The stopping criterion of S-DICCG-16 is apparently more severe compared to ICCG. The relative exact errors have also been compared and they are comparable and small (order between 10^{-7} and 10^{-8}), although the error in the case of S-DICCG-16 is somewhat smaller. Thus, the S-DICCG-16 requires more iterations but a somewhat higher accuracy of the real solution has been achieved.

Next, results with various jumps in the density and $n = 64^2$ are given in Table 5. From this table, we see that for larger contrasts between the density, ICCG requires more iterations and CPU time. This does not hold for L-DICCG$-k$ and LS-DICCG$-k$, which is a favorable property of these methods. S-DICCG$-k$ is also insensitive for the contrasts in the density for sufficiently large k.

Moreover, notice that S-DICCG$-k$ with $k \leq 16$ converges slower than the original ICCG, but for larger k it appears that S-DICCG$-k$ is very fast in convergence.



(a) ICCG (159 iterations). (b) S-DICCG−16 (190 iterations).

FIGURE 13. Relative update and relative exact residuals in the cases of ICCG and S-DICCG−16 for Test Problem T2 with $n_x = n_y = 64$.

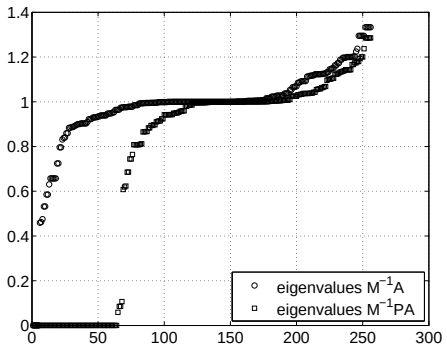
		$\epsilon = 10^{-3}$		$\epsilon = 10^{-6}$	
Deflation Method	k	# It.	CPU	# It.	CPU
ICCG	–	118	8.12	159	10.92
S-DICCG−k	4	134	9.79	194	14.01
	16	131	9.60	193	13.82
	64	26	2.31	26	2.14
L-DICCG−k	5	74	5.44	75	6.17
LS-DICCG−k	12	54	4.05	54	4.08
	36	40	3.08	40	3.05
	84	25	2.46	25	2.05

TABLE 5. Results of Test Problem T2 for varying jumps in the density with fixed $n_x = n_y = 64$.

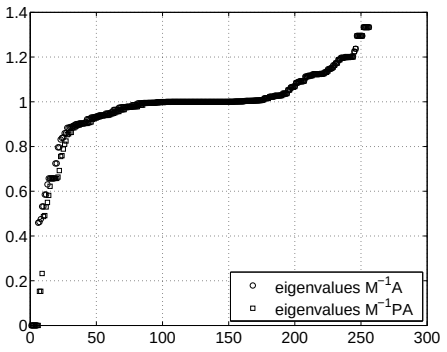
Finally, we consider results with various number of bubbles and $n = 64^2$, see Table 6. From this table, one observes that ICCG needs more iterations for problems with more bubbles. For the methods L-DICCG−k, LS-DICCG-k and S-DICCG−k with sufficiently large k, we see that the performance depends less on the number of bubbles, which is also a favorable property. Moreover, we observe that L-DICCG−k converges in less iterations for problems with more bubbles. We end with the remark that S-DICCG−k with $k \leq 16$ converges again slower than the original ICCG.

		1 bubble		2 bubbles		5 bubbles	
Deflation Method	k	# It.	CPU	# It.	CPU	# It.	CPU
ICCG	–	89	6.13	104	7.20	159	10.92
S-DICCG−k	4	96	7.39	69	5.13	194	14.01
	16	52	3.97	64	4.79	193	13.82
	64	26	2.14	27	2.16	26	2.14
L-DICCG−k	5	84	6.18	79	5.79	75	6.17
LS-DICCG−k	12	67	5.30	65	5.11	54	4.08
	36	41	3.14	42	3.22	40	3.05
	84	26	2.50	26	2.11	25	2.05

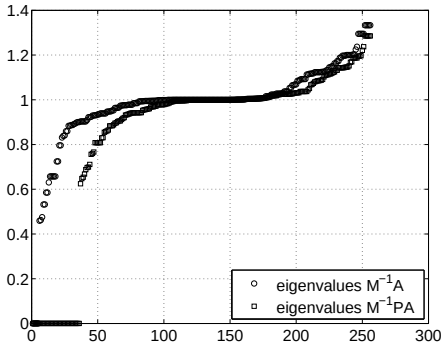
TABLE 6. Results of Test Problem T2 for various number of bubbles.



(a) S-DICCG-64 (19 iterations).



(b) L-DICCG-5 (17 iterations).



(c) LS-DICCG-36 (10 iterations).

FIGURE 14. Eigenvalues of $M^{-1}A$ and $M^{-1}PA$ with the three variants of deflation methods for Test Problem T2 with $n_x = n_y = 16$.

10.4. **Discussion of Small Eigenvalues for all Deflation Variants.** In the theory we have mentioned that approximating the smallest eigenvalues corresponding to the bubbles should be done carefully. In the case of Test Problem T2 with $n_x = n_y = 16$, it appears that the bubbles are very close to each other. As a result, some row sums of the levelset deflation subspace matrix are larger than zero in order to obtain good approximations of the eigenvectors. However, in S-DICCG$-k$ non-overlapping deflation vectors have been used. In this subsection, we investigate the consequences for all deflation variants for the smallest eigenvalues considering Test Problem T2, see Figure 14. In contrast to previous spectra plots, these plots are on normal scale so that the smallest eigenvalues can be observed more clearly.

From Subfigure 14(a) we notice that some small eigenvalues (around 0.1) remain in the spectrum for the case of S-DICCG-64. In fact, the smallest eigenvalues of order 10^{-6} are replaced by small eigenvalues of order 10^{-1}. This holds partly for L-DICCG-5, see Subfigure 14(b). The only difference is that the small eigenvalues of L-DICCG-5 are somewhat larger and are close to the tail of the remainder of the cluster than the small eigenvalues of S-DICCG-64.

In the case of LS-DICCG-36 we see that the smallest eigenvalues are completely eliminated of the spectrum, see Subfigure 14(c). This is the great advantage of this levelset-subdomain method compared to the subdomain and levelset deflation method. If the bubbles are very close to each other, than the LS-DICCG$-k$ appear to be the best method.

10.5. **Results Deflation with Different Preconditioners for Test Problem P2.** In the previous section we have introduced the MIC and RIC preconditioners as alternatives for the IC preconditoners. In this subsection, we will perform some 2-D numerical experiments using Test Problems P2 and T2 to test these preconditioners in combination with deflation. In these experiments we only use S-DICCG-64 (i.e., DICCG with 64 subdomain deflation vectors) and when there is no ambiguity we denote it simply as DICCG. In similar way, we introduce DMICCG and DRICCG.

10.5.1. *Invertible A.* We take grid sizes $n_x = n_y = 16$ in Test Problem P2. We force A to be invertible by modifying the last element of the matrix. Then, the results can be found in Table 7. We see that for all cases the deflated variant requires less

Method	α	Iterations
ICCG	–	30
DICCG	–	10
RICCG	0.5	27
DRICCG	0.5	11
RICCG	0.95	27
DRICCG	0.95	15
MICCG	–	30
DMICCG	–	22

TABLE 7. Results of Test Problem P2 and invertible A with $n = 16^2$. In the deflation variants, $k = 64$ has been used.

iterations than the original methods. The largest difference in iterations has been observed for ICCG and DICCG.

Next, we compare the spectra of $M_{IC}^{-1}A$ and $M_{IC}^{-1}PA$, $M_{MIC}^{-1}A$ and $M_{MIC}^{-1}PA$ and finally $M_{RIC}^{-1}A$ and $M_{RIC}^{-1}PA$, see Figure 15.

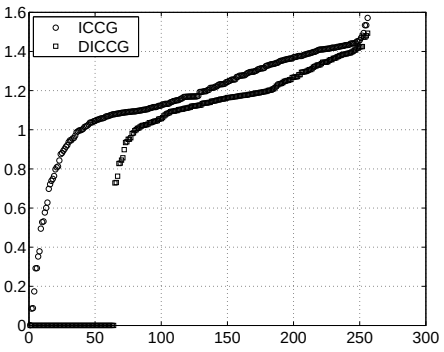
Obviously, in Subfigure 15(a) we observe that small eigenvalues corresponding to ICCG are eliminated by deflation, which is in contrast to Subfigure 15(b) where in MICCG the largest eigenvalues are eliminated by deflation. A more clear view of the spectra can be found in Figure 16. The fact that the largest eigenvalues have been eliminated looks contradictory. We know that small eigenvalues of ICCG correspond to slow eigenmodes. However, it appears that the large eigenvalues of MICCG also correspond to slow-varying eigenmodes and the other eigenvalues to relatively fast-varying eigenmodes, see also Figure 17.

Moreover, in both Subfigures 15(c) and 15(d) we see that both small and large eigenvalues are eliminated by deflation for ICCG and MICCG, respectively. Apparently, both small and large eigenvalues associate to slow eigenmodes.

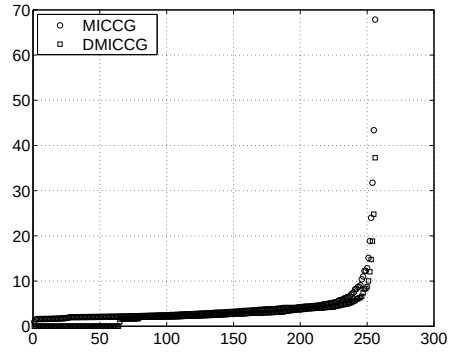
10.5.2. *Singular A.* Next we will do some experiments with the original singular A. In this case MICCG and DMICCG are not defined. We also vary the number of grid points. The results can be found in Table 8.

From Table 8 we see immediately that for all IC-variants the deflation method improves again the original method. Moreover, note that an invertible A leads to a slower original method ICCG and RICCG compared to the singular case (cf. Table 7), but the deflation remedies this, i.e., there are no differences between DICCG and DRICCG applied on singular and invertible A.

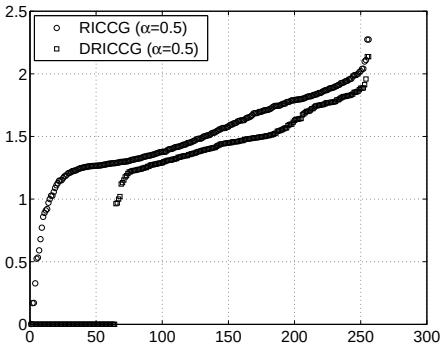
Additionally, for the cases of $n = 64^2$ and $n = 80^2$, DRICCG with $\alpha = 0.5$ converges most rapidly, although the differences with the other deflation variants are small.



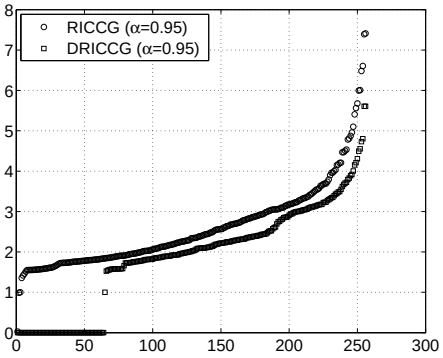
(a) IC preconditioner.



(b) MIC preconditioner.

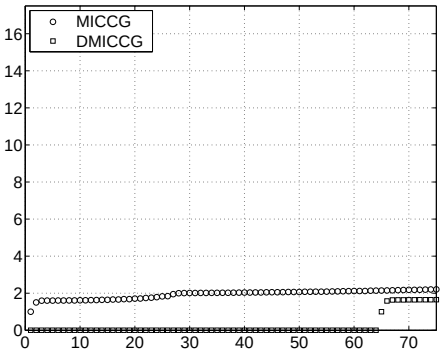


(c) RIC preconditioner with $\alpha = 0.5$.

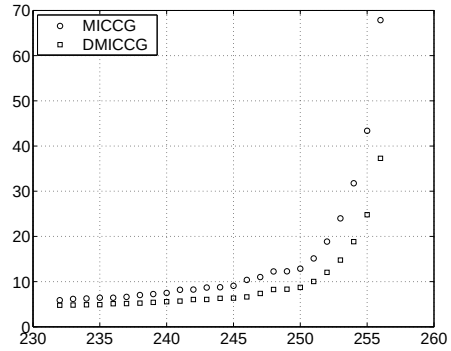


(d) RIC preconditioner with $\alpha = 0.95$.

FIGURE 15. Spectra of $M^{-1}A$ and $M^{-1}PA$ with an invertible A for three preconditioners in Test Problem P2 .



(a) First values of the spectra.



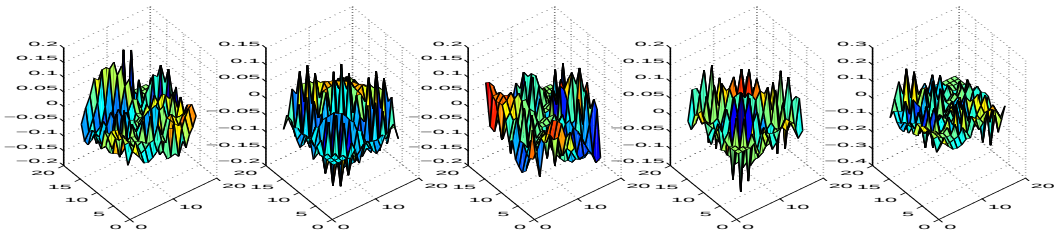
(b) Last values of the spectra.

FIGURE 16. Zoomed-in spectra of $M_{MIC}^{-1}A$ and $M_{MIC}^{-1}PA$ with an invertible A in Test Problem P2 (cf. Subfigure 15(b)).

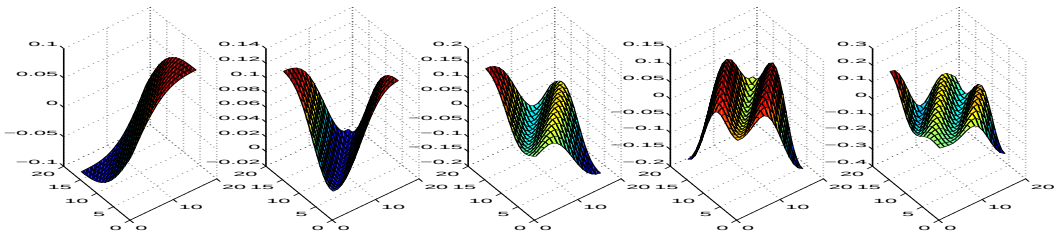
10.6. **Results Deflation with Different Preconditioners for Test Problem T2.** For Test Problem T2 with contrast $\epsilon = 10^{-3}$, the results are presented in the next subsections.

10.6.1. *Invertible A.* The results with invertible matrix A can be found in Table 9.

In contrast to Test Problem P2 and to the case of ICCG, we now see that deflation does not help significantly in the case of MICCG. This can be explained by



(a) Three eigenvectors associated to non-large eigenvalues.



(b) Three eigenvectors associated to large eigenvalues.

FIGURE 17. Eigenvectors associated to the eigenvalues of $M^{-1}A$ for MICCG.

Method	α	$n = 16^2$ Iterations	$n = 64^2$ Iterations	$n = 80^2$ Iterations
ICCG	–	24	77	95
DICCG	–	11	27	32
RICCG	0.5	21	66	81
DRICCG	0.5	11	26	30
RICCG	0.95	21	46	53
DRICCG	0.95	15	30	34

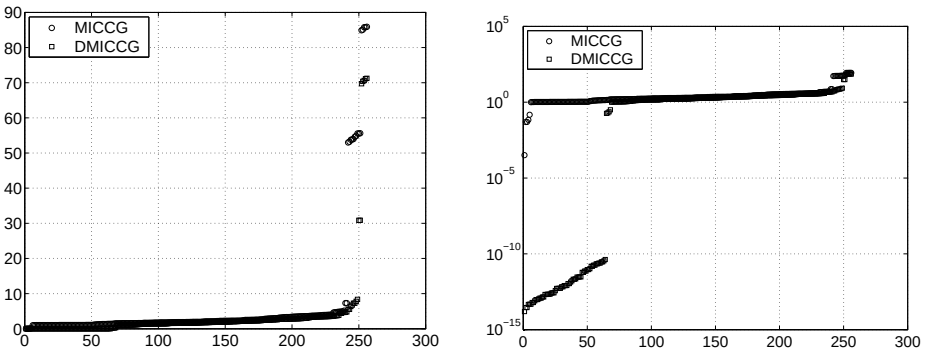
TABLE 8. Results with singular A in Test Problem P2. In the deflation variants, $k = 64$ has been used.

Method	α	Iterations
ICCG	–	44
DICCG	–	22
RICCG	0.5	47
DRICCG	0.5	24
RICCG	0.95	82
DRICCG	0.95	48
MICCG	–	132
DMICCG	–	112

TABLE 9. Results of Test Problem T2 using an invertible A with $n = 16^2$.

considering the spectrum of MICCG, see also Figure 18. It can be noticed that some large eigenvalues are eliminated, but small eigenvalues associated to the bubbles remain in the spectrum. That is the reason why DMICCG does not accelerate the convergence compared to MICCG.

Moreover since DICCG performs well, also RICCG with $\alpha < 1$ performs comparably. Finally, note that ICCG and its deflated variant are clearly the fastest



(a) Normal scale. (b) Log scale.

FIGURE 18. Spectra of $M^{-1}A$ and $M^{-1}PA$ with an invertible A for MICCG in Test Problem T2.

methods compared to MICCG and RICCG. This is caused by the fact that the spectrum of ICCG consists of small eigenvalues which are eliminated by deflation, while the largest eigenvalues in the spectrum of ICCG are of order 1 resulting in a relatively small effective condition number after deflation.

10.6.2. *Singular A.* The results with singular A can be found in Table 10. We see again that in all cases, the deflation variants requires significantly less iterations than the original methods. DICCG and DRICCG with small α perform the best. This is remarkable, since for the undeflated variants, RICCG with $\alpha = 0.95$ is obviously the fastest method.

In future, more experiments with larger grids will be done.

Method	α	$n = 16^2$ Iterations	$n = 64^2$ Iterations	$n = 80^2$ Iterations
ICCG	–	31	126	150
DICCG	–	23	28	33
RICCG	0.5	37	111	136
DRICCG	0.5	24	28	33
RICCG	0.95	66	93	115
DRICCG	0.95	49	44	51

TABLE 10. Results with a singular A in Test Problem T2. In the deflation variants, $k = 64$ has been used.

10.7. **Concluding Remarks.** After presenting the numerical results, the following conclusions can be drawn.

- Using subdomain deflation, the 'small' eigenvalues of Test Problem P2 are eliminated since the corresponding eigenvectors are smooth.
- For Test Problem T2, subdomain deflation is only effective for sufficiently large number of subdomains. In these cases, not only the smallest eigenvalues corresponding to the bubbles are eliminated but also other small eigenvalues.
- The levelset deflation eliminates the smallest eigenvalues in Test Problem T2. To eliminate more small eigenvalues the levelset-subdomain deflation method can be applied. Both methods have proven to be effective.

- All variants of the deflation methods are efficient for problems with varying grid sizes. LS-DICCG$-k$ and S-DICCG$-k$ appear to be comparable for sufficiently large k. For small k, L-DICCG$-k$ should be chosen as deflation method.
- All variants of the deflation method (except for S-DICCG$-k$ with relatively small k) are insensitive for the contrast in the densities, whereas the performance of ICCG strongly depends on this contrast. This means that the deflation method is not restricted to applications of bubbly flow problems, but it can also be applied easily on various other multi-phase problems.
- All variants of the deflation method (except for S-DICCG$-k$ with relatively small k) depend hardly on the number of bubbles, in contrast to ICCG. This means that in more complex problems, the deflation method can also be efficient.
- If the bubbles are close to each other in the bubbly-flow problem, S-DICCG$-k$ can show difficulties to approximate the smallest eigenvalues and therefore it can be slow in convergence. Fortunately, the LS-DICCG$-k$ does not show these difficulties. Hence, in these cases, LS-DICCG$-k$ is the most favorable method to apply.
- Adding the deflation technique to ICCG, RICCG and MICCG is in all cases efficient for Test Problem P2. For Test Problem T2 it is only beneficial for ICCG and RICCG. However, ICCG and its deflated variant DICCG remain the best methods in the experiments.

11. Conclusions

In previous papers we have considered eigenvector and subdomain deflation techniques. In this paper we have explained why subdomain deflation works well and we have introduced several new variants of deflation: perturbed eigenvector, levelset and levelset-subdomain deflation.

The perturbed eigenvector deflation has been introduced to get more feeling with the choice of deflation vectors. The main result is that eigenvectors corresponding to small eigenvalue can be perturbed in some sense while they remain very good approximations so that they can be used as deflation vectors to project the smallest eigenvalues to exactly zero.

In the 2-D numerical experiments we have compared the results of the subdomain, levelset and levelset-subdomain deflation techniques. The main conclusion is that they all perform very well compared to ICCG. It depends on the geometry of problem and on the number of deflation vectors, which variant of the deflation method is the most effective one.

Finally, replacing the original IC preconditioners by the MIC or RIC preconditioners in the experiments do not improve the convergence, although in some cases extending the resulting methods MICCG and RICCG with the deflation technique can accelerate the iterative process in simple test problems.

References

[1] O. Axelsson and G. Lindskog, *On the Eigenvalue Distribution of a Class of Preconditioned Methods*, Numer. Math., **48**, pp. 479–498, 1986.

[2] E.F. Kaasschieter, *Preconditioned conjugate gradients for solving singular systems*, J. Comp. Appl. Maths., **24**, pp. 265–275, 1988.

[3] G.H. Golub, C.F. van Loan, *Matrix Computations*, Third Edition, The John Hopkins University Press, Baltimore, 1996.

[4] W. Mulder, S. Osher, J.A. Sethian, *Computing interface motion in compressible gas dynamics*. Journal of Computational Physics, **100**, pp 209-228, 1992.

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

[6] R.A. Nicolaides, *Deflation of Conjugate Gradients with applications to boundary value problems*, SIAM J. Matrix Anal. Appl., **24**, pp. 355–365, 1987.
[7] S. Osher, R.P. Fedkiw, *Level set methods: an overview and some recent results*. Journal of Computational Physics, **169**, pp. 463–502, 2001.
[8] S.P. van der Pijl, *Computation of bubbly flows with a mass-conserving level-set method*, PhD thesis, Delft University of Technology, 2005.
[9] 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. Meth. Fluids , **47**, pp. 339–361, 2005.
[10] R. Scheichl and E. Vainikko, *Additive Schwarz and Aggregation-Based Coarsening for Elliptic Problems with Highly Variable Coefficients*, submitted (see also `http://www.bath.ac.uk/math-sci/BICS`), 2006.
[11] J.M. Tang, *Parallel Deflated CG Methods applied to Linear Systems from Moving Boundary Problems*, DUT Report 05-02, Delft University of Technology, 2005.
[12] J.M. Tang and C. Vuik, *On the Theory of Deflation and Singular Symmetric Positive Semi-Definite Matrices*, DUT Report 05-06, Delft University of Technology, 2005.
[13] J.M. Tang and C. Vuik, *On Deflation and Singular Symmetric Positive Semi-Definite Matrices*, *accepted for publication in* J. Comp. Appl. Math., 2006.
[14] J.M. Tang and C. Vuik, *An Efficient Deflation Method applied on 2-D and 3-D Bubbly Flow Problems*, DUT Report 06-01, Delft University of Technology, 2006.
[15] J.M. Tang and C. Vuik, *An Efficient Deflation Method applied on 2-D and 3-D Bubbly Flow Problems*, *submitted to* Electronic Transactions on Numerical Analysis, 2006.
[16] C. Vuik, A. Segal, J.A. Meijerink and G.T. Wijma, *The construction of projection vectors for a Deflated ICCG method applied to problems with extreme contrasts in the coefficients*, J. Comp. Physics, **172**, pp. 426–450, 2001.