Deflation acceleration for domain decomposition preconditioners

C. Vuik[*] R. Nabben[†] J. Tang[‡]

January 13, 2006

Abstract

In this paper we compare various preconditioners for the numerical solution of partial differential equations. We start by investigating the influence of a so-called deflation preconditioner on the convergence of the preconditioned conjugate gradient method. Thereafter, we compare an additive coarse grid correction preconditioner and the balancing preconditioner, both used in domain decomposition methods, with the deflation preconditioner. We prove that the effective condition number of the deflated preconditioned system is always, i.e. for all deflation vectors and all restrictions and prolongations, below the condition number of the system preconditioned by the coarse grid correction or balancing preconditioner. This implies that the conjugate gradient method applied to the deflated preconditioned system is expected to converge always faster than the conjugate gradient method applied to the system preconditioned by the coarse grid correction or balancing. Numerical results for porous media flows emphasize the theoretical results.

Keywords. deflation, coarse grid correction, balancing, preconditioners, conjugate gradients, porous media flow, scalable parallel preconditioner

AMS subject classifications. 65F10, 65F50, 65N22

1 Introduction

It is well known that the convergence rate of the conjugate gradient method is bounded as a function of the condition number of the system matrix to which it is applied. Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite (SPD). We assume that the vector $b \in \mathbb{R}^n$ represents a discrete function on a grid Ω and that we are searching for the vector $x \in \mathbb{R}^n$ on Ω which solves the linear system

$$Ax = b.$$

Such systems are encountered, for example, when a finite volume/difference/element method is used to discretize an elliptic partial differential equation defined on the continuous analogy of Ω.

Let us denote the i-th eigenvalue in non-decreasing order by $\lambda_i(A)$ or simply by λ_i when it is clear to which matrix we are referring. After k iterations of the conjugate gradient method, the error is bounded by (cf. [8], Thm. 10.2.6)

$$\|x - x_k\|_A \leq 2\,\|x - x_0\|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \tag{1}$$

[*] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: 31 15 2785530, fax: 31 15 2787209 (c.vuik@tudelft.nl)

[†] TU Berlin Institut für Mathematik, MA 3-3 Strasse des 17. Juni 136 D-10623 Berlin Germany, phone 49 30-314-29291 (nabben@math.tu-berlin.de)

[‡] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: 31 15 2785530, fax: 31 15 2787209 (J.M.Tang@tudelft.nl)

where $\kappa = \kappa(A) = \lambda_n/\lambda_1$ is the spectral condition number of A and the A-norm of x is given by $\|x\|_A = (x^TAx)^{1/2}$. The convergence may be significantly faster if the eigenvalues of A are clustered [22].

If the condition number of A is large, it is advisable to solve, instead, a preconditioned system $M^{-1}Ax = M^{-1}b$, where the symmetric positive definite preconditioner M is chosen such that $M^{-1}A$ has a more clustered spectrum or a smaller condition number than that of A. Furthermore, M must be cheap to solve relative to the improvement it provides in convergence rate. A final desirable property in a preconditioner is that it should parallelize well, especially on distributed memory computers. Probably one of the most effective preconditioning strategy in common use is to take $M = LL^T$ to be an incomplete Cholesky (IC) factorization of A [13]. We denote the preconditioned conjugate gradient method by PCG.

With respect to the known preconditioners at least two problems remain:

- if there are large jumps in the coefficients of the discretized PDE the convergence of PCG becomes very slow,

- if a block preconditioner is used in a domain decomposition algorithm the condition number of the preconditioned matrix deteriorates if the number of blocks increases.

Both problems can be solved by a deflation technique [16, 23]. To describe the deflation method we define the projection P_D by

$$P_D = I - AZ(Z^TAZ)^{-1}Z^T, \quad Z \in \mathbb{R}^{n\times r}, \tag{2}$$

where the column space of Z is the deflation subspace, i.e. the space to be projected out of the residual, and I is the identity matrix of appropriate size. We assume that $r \ll n$ and that Z has rank r. Under this assumption $E := Z^TAZ$ may be easily computed and factored and moreover, E is symmetric positive definite. Since $x = (I - P_D^T)x + P_D^Tx$ and because

$$(I - P_D^T)x = Z(Z^TAZ)^{-1}Z^TAx = ZE^{-1}Z^Tb \tag{3}$$

can be immediately computed, we only need to compute P_D^Tx. In light of the identity $AP_D^T = P_DA$, we can solve the deflated system

$$P_DA\tilde{x} = P_Db \tag{4}$$

for $\tilde{x}$ using the conjugate gradient method, premultiply this by P_D^T and add it to (3). This is correct because $P_D^T\tilde{x} = P_D^Tx$.

Obviously (4) is singular. Kaasschieter [10] notes that a positive semidefinite system can be solved as long as the linear system is consistent (i.e. as long as $b = Ax$ for some x). This is certainly true for (4), where the same projection is applied to both sides of the nonsingular system. Furthermore, he notes (with reference to [22]) that because the null space never enters the iteration, the corresponding zero-eigenvalues do not influence the convergence. Motivated by this fact, we define the *effective condition number* of a positive semidefinite matrix $C \in \mathbb{R}^{n\times n}$ with r zero eigenvalues to be the ratio of its largest to smallest *positive* eigenvalues:

$$\kappa_{\text{eff}}(C) = \frac{\lambda_n}{\lambda_{r+1}}.$$

In this paper we start to investigate the convergence acceleration of deflation. Thereafter, we prove that the effective condition number of the deflated preconditioned system is always below the condition number of the system preconditioned by the coarse grid correction or balancing preconditioner. This implies that for all matrices $Z \in \mathbb{R}^{n\times r}$ and all positive definite preconditioners M the conjugate gradient method applied to the deflated preconditioned system is expected to converge always faster than the conjugate gradient method applied to the system preconditioned by the coarse grid correction or balancing preconditioner. Some numerical results are given to illustrate the theory.

2 Convergence acceleration of deflation

In this section, first we will prove that the effective condition number of P_DA is always lower than the condition number of A for all choices of Z, see Theorem 2.1.

Theorem 2.1 *Let A and P_D be as defined in Section 1. Then the following inequality holds:*

$$\kappa_{\mathit{eff}}(P_DA) < \kappa(A). \tag{5}$$

Thereafter we proof that this can be generalized in the case of using an SPD preconditioner M, see Theorem 2.2.

Theorem 2.2 *Let A and P_D be as defined in Section 1. Let M be an $n \times n$ SPD matrix. Then the following inequality holds:*

$$\kappa_{\mathit{eff}}(M^{-1}P_DA) < \kappa(M^{-1}A). \tag{6}$$

Both theorems imply that using deflation always leads to an asymptotically faster convergence.

This section is organized as follows. We start with some auxiliary results in Subsection 2.1, which are needed in the proofs of Theorem 2.1 and 2.2. Thereafter, in Subsection 2.2 the proof of Theorem 2.1 is given after showing that the inequalities $\lambda_{r+1}(P_DA) \geq \lambda_1(A)$ and $\lambda_n(P_DA) < \lambda_n(A)$ hold. Finally, we end up with the proof of Theorem 2.2 in the last subsection.

2.1 Auxiliary Results

A set of lemmata and a theorem, which are needed to prove Theorems 2.1 and 2.2, are given below.

Lemma 2.3 *Let Q be a projection matrix and let R be an SPD matrix with dimensions $n \times n$ such that QR is symmetric. Then QR is also SPD.*

Proof: By definition, $u^TRu > 0$ for all vectors u. In particular,

$$(Q^Tu)^TR(Q^Tu) > 0$$

leading to

$$(Q^Tu)^TR(Q^Tu) = u^TQRQ^Tu > 0.$$

In other words, $QRQ^T = Q(RQ^T)^T = Q^2R = QR$ is SPD. □

Lemma 2.4 *Matrix $I - P_D$ is a projector.*

Proof: By definition, $I - P_D = AZE^{-1}Z^T$ so that

$$(I - P_D)^2 = AZE^{-1}Z^TAZE^{-1}Z^T = AZE^{-1}EE^{-1}Z^T = AZE^{-1}Z^T = I - P_D.$$

□

Next, two lemmata are given about the symmetry of some matrices.

Lemma 2.5 *Define $T := (I - P_D)A$ with P_D as defined in Section 1. Then T is symmetric.*

Proof: Note first that $T = (I - P_D)A = -AZE^{-1}Z^TA$. Since

$$T^T = (-AZE^{-1}Z^TA)^T = -A^TZE^{-T}Z^TA = -AZE^{-1}Z^TA = T,$$

matrix T is symmetric. □

Let $M^{1/2}$ be the symmetric square root of M (see page 149 of [8]).

Lemma 2.6 *Let $\widehat{A} := M^{-1/2}AM^{-1/2}$ with M to be SPD. Then $\widehat{A}$ is SPD.*

Theorem 2.7 *(see [14] Theorem 2.10) Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Let $Z_1 \in \mathbb{R}^{n\times r}$ and $Z_2 \in \mathbb{R}^{n\times s}$ with rank $Z_1 = r$ and rank $Z_2 = s$. Let $E_1 := Z_1^T A Z_1$ and $E_2 := Z_2^T A Z_2$. If $ImZ_1 \subseteq ImZ_2$, then*

$$\begin{aligned} \lambda_n((I - AZ_1E_1^{-1}Z_1^T)A) &\geq \lambda_n((I - AZ_2E_2^{-1}Z_2^T)A), & (7)\\ \lambda_{r+1}((I - AZ_1E_1^{-1}Z_1^T)A) &\leq \lambda_{s+1}((I - AZ_2E_2^{-1}Z_2^T)A). & (8) \end{aligned}$$

This theorem states that the effective condition number decreases if we increase the number of deflation vectors.

2.2 Comparison of the (Effective) Condition Numbers of the Matrices A and P_DA

In this subsection, the proof of Theorem 2.1 is given.

Proof: [Proof of Theorem 2.1.] Note that

$$A - P_DA = VA, \quad V := AZE^{-1}Z^T = I - P_D.$$

$V = I - P_D$ is a projector due to Lemma 2.4. Obviously, applying the identity $P_DA = AP_D^T$, we have that VA is symmetric. Next, since A is SPD, we obtain that VA is also SPD, by using Lemma 2.3. Therefore, by definition, $A \prec P_DA$ so that

$$\lambda_i(A) > \lambda_i(P_DA),$$

by Theorem 4.3.1 of [9]. Thus in particular:

$$\lambda_n(A) > \lambda_n(P_DA).$$

Now we prove that the inequality $\lambda_1(A) \leq \lambda_{r+1}(P_DA)$ holds. We start to prove that $\lambda_1(A) \leq \lambda_2(P_{(1)}A)$, where $P_{(1)}$ is the deflation operator with $Z_{(1)} = z_1$. Note first that

$$P_{(1)}A = A - Az_1E_{(1)}^{-1}z_1^TA = A + T.$$

with $T = (I - P_{(1)})A = -Az_1E_{(1)}^{-1}z_1^TA$. Moreover, since $E_{(1)}^{-1}$ is a scalar, we write $\alpha := -E_{(1)}^{-1} \in \mathbb{R}$. Hence,

$$T = -Az_1E_{(1)}^{-1}z_1^TA = \alpha Az_1z_1^TA.$$

Obviously, rank $\alpha Az_1z_1^TA =$ rank $Az_1z_1^TA$. Furthermore, since A is invertible,

$$\text{rank } Az_1z_1^TA = \text{rank } z_1z_1^T.$$

Because of rank $z_1z_1^T = 1$, we obtain that rank $T = 1$. Moreover, T is symmetric by applying Lemma 2.5. Hence, the conditions of Theorem 4.3.6 of [9] have been satisfied, so we obtain immediately

$$\lambda_1(A) \leq \lambda_2(P_{(1)}A).$$

Application of Theorem 2.7 leads to the inequality

$$\lambda_1(A) \leq \lambda_2(P_{(1)}A) \leq \lambda_{r+1}(P_DA).$$

In the previous two steps it has been proven that

$$\lambda_1(A) \leq \lambda_{r+1}(P_DA), \quad \lambda_n(A) > \lambda_n(P_DA),$$

for all Z with rank $Z = r$. Hence, this leads to

$$\tilde{\kappa}(P_DA) < \kappa(A).$$

□

2.3 Comparison of the (Effective) Condition Numbers of the Matrices $M^{-1}A$ and $M^{-1}P_DA$

As mentioned in the beginning of this section, Theorem 2.1 can be generalized to deflated preconditioned systems $M^{-1}P_DA$ where M is an SPD matrix. This leads to Theorem 2.2 whose the proof can be found below.

Proof: [Proof of Theorem 2.2.] Let $\widehat{A} := M^{-1/2}AM^{-1/2}$. Then $\widehat{A}$ is SPD from Lemma 2.6.

Note that

$$\kappa_{\text{eff}}(M^{-1}P_DA) = \kappa_{\text{eff}}(M^{-1/2}P_DAM^{-1/2}) = \kappa_{\text{eff}}(M^{-1/2}P_DM^{1/2}\widehat{A}) \tag{9}$$

and

$$\kappa(M^{-1}A) = \kappa(M^{-1/2}AM^{-1/2}) = \kappa(\widehat{A}) \tag{10}$$

using the fact that $\kappa(B_1B_2) = \kappa(B_2B_1)$ (with the standard 2-norm) for two arbitrary invertible symmetric matrices B_1 and B_2.

Next, define $\widehat{P}_D$ as

$$\widehat{P}_D := I - \widehat{A}Y\widehat{E}^{-1}Y^T, \quad \widehat{E} := Y^T\widehat{A}Y$$

with $Y := M^{1/2}Z$. Since $M^{1/2}$ is invertible, Y is of rank r. Note further that

$$E = Z^TAZ = (M^{-1/2}Y)^TAM^{-1/2}Y = Y^T\widehat{A}Y = \widehat{E}.$$

Now we obtain

$$\begin{aligned} M^{-1/2}P_DM^{1/2} &= M^{-1/2}(I - AZE^{-1}Z^T)M^{1/2} \\ &= I - M^{-1/2}AZE^{-1}Z^TM^{1/2} \\ &= I - \widehat{A}M^{1/2}ZE^{-1}Z^TM^{1/2} \\ &= I - \widehat{A}Y\widehat{E}^{-1}Y^T \\ &= \widehat{P}_D. \end{aligned}$$

Hence, Equation (9) can now be rewritten as

$$\kappa_{\text{eff}}(M^{-1}P_DA) = \kappa_{\text{eff}}(M^{-1/2}P_DM^{1/2}\widehat{A}) = \kappa_{\text{eff}}(\widehat{P}_D\widehat{A}). \tag{11}$$

From Theorem 2.1 we know that $\kappa_{\text{eff}}(P_DA) < \kappa(A)$ for arbitrary Z with rank r and for arbitrary SPD matrix A. In particular we can take $P_D = \widehat{P}_D$ and $A = \widehat{A}$, since Y is also of rank r and $\widehat{A}$ is SPD. Therefore we obtain

$$\kappa_{\text{eff}}(\widehat{P}_D\widehat{A}) < \kappa(\widehat{A}),$$

which is equivalent to

$$\kappa_{\text{eff}}(M^{-1}P_DA) < \kappa(M^{-1}A).$$

□

3 Comparison of deflation and additive coarse grid correction

In this section we compare the deflation preconditioner with a well-known coarse grid correction preconditioner of the form

$$P_C = I + ZE^{-1}Z^T \tag{12}$$

and in the preconditioned case

$$P_{CM^{-1}} = M^{-1} + ZE^{-1}Z^T. \tag{13}$$

In the multigrid or domain decomposition language the matrices Z and Z^T are known as restriction and prolongation or interpolation operator. Moreover, the matrix $E = Z^TAZ$ is the Galerkin operator.

The above coarse grid correction preconditioner belongs to the class of additive Schwarz preconditioner. It is called the two level additive Schwarz preconditioner. If used in domain decomposition methods, typically, M^{-1} is the sum of the local (exact or inexact) solves in each domain. To speed up convergence a coarse grid correction $ZE^{-1}Z^T$ is added.

These methods are introduced by Bramble, Paschiak and Schatz [1] and Dryja and Widlund [2, 3, 4]. They show that under mild conditions the convergence rate of the PCG method is independent of the grid sizes.

For more details about this preconditioner we refer to the books of Quarteroni and Valli [19], and Smith, Bjørstad and Gropp [20]. A more abstract analysis of this preconditioner is given by Padiy, Axelsson and Polman [17], recently. To make the condition number of $P_{CM^{-1}}A$ smaller Padiy, Axelsson and Polman used a parameter $\sigma > 0$ and considered

$$P_{CM^{-1}}(\sigma) = M^{-1} + \sigma ZE^{-1}Z^T. \tag{14}$$

We compare this preconditioner to the corresponding deflated preconditioner

$$M^{-1}P_D. \tag{15}$$

Then, we obtain the following theorem.

Theorem 3.1 *Let $A \in \mathbb{R}^{n\times n}$ and $M \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Let $Z \in \mathbb{R}^{n\times r}$ with rank $Z = r$. Then*

$$\begin{aligned} \lambda_n(M^{-1}P_DA) &\leq \lambda_n(P_{CM^{-1}}(\sigma)A), & (16)\\ \lambda_{r+1}(M^{-1}P_DA) &\geq \lambda_1(P_{CM^{-1}}(\sigma)A). & (17) \end{aligned}$$

Proof: See the proof of Theorem 2.11 in [14]. □

In this theorem we prove that the effective condition number of the deflated preconditioned system $M^{-1}P_DA$ is always below the condition number of the system preconditioned by the coarse grid correction $P_{CM^{-1}}(\sigma)A$. This implies that for all matrices $Z \in \mathbb{R}^{n\times r}$ and all positive definite preconditioners M^{-1} the conjugate gradient method applied to the deflated preconditioned system converges always faster than the conjugate gradient method applied to the system preconditioned by the coarse grid correction. Finally, we note that the memory requirements and the amount of work per iteration of both preconditioners are comparable. This implies, that in practice it is better to use deflation instead of the additive coarse grid correction.

4 Comparison of deflation and balancing

In this section we compare the preconditioned deflation operator to the balancing preconditioner proposed by Mandel [11, 12] and analyzed by Widlund et al. [5, 18, 21]. As the FETI algorithm [6, 7] the balancing Neumann-Neumann preconditioner is one of the domain decomposition methods that have been most carefully implemented and severely tested on the very largest existing parallel computer systems.

Applied to some specific symmetric positive definite problems the balancing Neumann-Neumann preconditioner leads to moderately growing condition numbers if the size of the systems increases [20]. Moreover, if an appropriate scaling is used, the condition numbers are independent of jumps in the coefficients in the matrices [20].

In our notation the balancing preconditioner is given by

$$P_B = (I - ZE^{-1}Z^TA)M^{-1}(I - AZE^{-1}Z^T) + ZE^{-1}Z^T, \tag{18}$$

where $Z \in \mathbb{R}^{n\times r}$, $E = Z^TAZ$ and M is a symmetric positive definite matrix. Note that P_B is symmetric and positive definite. For more details we refer to [11] and the books [19, 20, 21].

As a first comparison of both preconditioners we observe that the balancing preconditioner needs per iteration 3 matrix vector products and the coarse grid operator is used 2 times. This makes the balancing preconditioner per iteration more expensive than the deflation approach. However, if an optimal implementation of the balancing preconditioner is used (see e.g. [21]), the amount of work per iteration is the same.

Theorem 4.1 *Suppose that the spectrum of $M^{-1}P_DA$ is given by:*

$$\textit{spectrum}(M^{-1}P_DA) = \{0,\ldots,0,\mu_{r+1},\ldots,\mu_n\},$$

then

$$\textit{spectrum}(P_BA) = \{1,\ldots,1,\mu_{r+1},\ldots,\mu_n\}.$$

Proof: See the proof of Theorem 2.8 in [15]. □

Thus both preconditioners lead to almost the same spectra with the same clustering. The zero eigenvalues of the deflation preconditioned system are replaced by eigenvalues which are one if the balancing preconditioner is used. It follows from Theorem 4.1 that

$$\kappa(P_BA) \geq \kappa_{eff}(M^{-1}P_DA)$$

so the convergence bound based on the effective condition number implies that preconditioned deflated CG never converges slower than CG preconditioned by the balancing preconditioner. This combined with memory requirements and work per iteration counts suggested that it is better to use the deflation preconditioner instead of the balancing preconditioner.

5 Numerical experiments

In all our numerical experiments, the multiplication $y = E^{-1}b$ is done by solving y directly from $Ey = b$, where E is decomposed in its Cholesky factor. In this section, coarse grid correction is abbreviated as CGC. The choice of the boundary conditions is such that all problems have as exact solution the vector with components equal to 1. In order to make the convergence behavior representative for general problems we chose a random vector as starting solution, in stead of the zero starting vector.

We simulate a porous media oil flow in a 3-dimensional layered geometry, where the layers vary in thickness and orientation (see Figure 1 and 2 for a 4 layer problem). The fluid pressure and permeability are denoted by p and σ_p, respectively. The pressure p satisfies the equation:

$$-\text{div}(\sigma_p \nabla p) = 0 \text{ on } \Omega, \tag{19}$$

with boundary conditions

$$p = 1 \text{ on } \partial\Omega^D \text{ (Dirichlet) and } \frac{\partial p}{\partial n} = 0 \text{ on } \partial\Omega^N \text{ (Neumann)},$$

where $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$. In this problem $\partial\Omega^D$ is the top boundary of the domain. Figure 1 shows a part of the earth's crust. The depth of this part varies between 3 and 6 kilometers, whereas horizontally its dimensions are 40×60 kilometers. The upper layer is a mixture of sandstone and shale and has a permeability of 10^{-4}. Below this layer, shale and sandstone layers are present with permeabilities of 10^{-7} and 10 respectively. An incomplete Cholesky factorization with no fill in is used as preconditioner [13]. We consider a problem with 9 layers: 5 sandstone layers are separated by 4 shale layers. Due to the Dirichlet boundary condition at the top the preconditioned matrix has 4 small eigenvalues. We use 4 *physical* projection vectors [24] and stop if $\|r_k\|_2 \leq 10^{-5}$. Trilinear hexahedral elements are used and the total number of gridpoints is equal to 148185.

The number of iterations and the CPU time of ICCG are given in Table 1. From this table it appears that the coarse grid correction preconditioner takes more iterations than the deflation preconditioner, which makes the required CPU time longer. It appears that the norm of the residuals of the deflation and balancing preconditioners are the same. Due to extra work per iteration, the balancing preconditioner costs more CPU time. The computations are performed on an AMD Athlon, 1.4 GHz processor with 256 Mb of RAM. The code is compiled with FORTRAN g77 on LINUX.

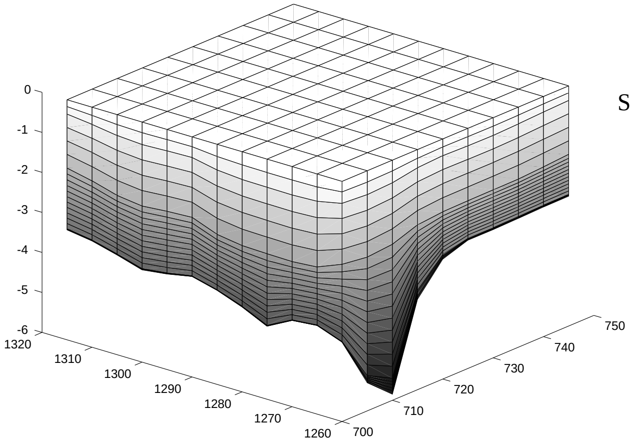

Figure 1: The geometry of an oil flow problem with 4 layers

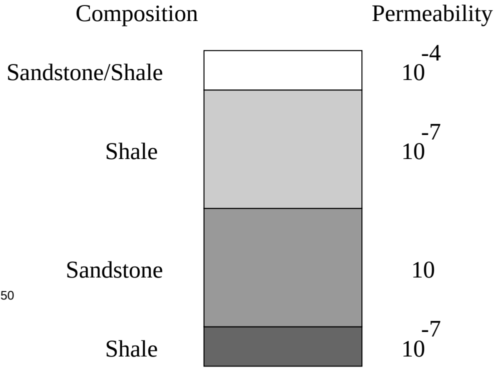

Figure 2: Permeabilities for each layer

method	deflation	CGC	balancing
iterations	36	47	36
CPU time in seconds	6.3	8.2	9.8

Table 1: The results for the ICCG method combined with various preconditioners applied to the oil flow problem

6 Conclusions

In this paper we have compared various preconditioners for the numerical solution of partial differential equations: deflation, coarse grid correction and balancing. It appears from theory and experiments that all these preconditioners lead to a (much) faster convergence of PCG. Furthermore, these preconditioners solve the problems concerning large jumps in the coefficients and the deterioration of the convergence for a block preconditioner if the number of blocks increases.

A comparison of the deflation preconditioner with the coarse grid correction preconditioner shows that the amount of work per iteration is the same, but the number of iterations with the coarse grid correction preconditioner is considerably higher than with the deflation preconditioner. Finally, it appears that the convergence of the balancing preconditioner is comparable to the convergence of the deflation preconditioner, but the amount of work per iteration is higher for the balancing preconditioner.

References

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

[2] M. Dryja. An additive Schwarz algorithm for two- and three dimensional finite element elliptic problems. In *Domain Decomposition methods*, pages 168–172. SIAM Philadelphia PA, 1989.

[3] M. Dryja and O.B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–21. SIAM Philadelphia PA, 1990.

[4] M. Dryja and O.B. Widlund. Multilevel additive methods for elliptic finite element problems in three dimensions. In *Parallel Algorithms for PDEs*, pages 58–69. Vieweg Braunschweig Germany, 1991.

[5] M. Dryja and O.B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48:121–155, 1995.

[6] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.*, 32:1205–1227, 1991.

[7] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. *Comput. Mech. Adv.*, 2:1–124, 1994.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[9] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

[10] E .F. Kaasschieter. Preconditioned Conjugate Gradients for solving singular systems. *J. Comput. Appl. Math.*, 24:265–275, 1988.

[11] J. Mandel. Balancing domain decomposition. *Commun. Appl. Numer. Meth.*, 9:233–241, 1993.

[12] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Mathematics of Computation*, 216:1387–1401, 1996.

[13] 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. *Mathematics of Computation*, 31:148–162, 1977.

[14] 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.

[15] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.*, 2006. to appear.

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

[17] A. Padiy, O. Axelsson, and B. Polman. Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems. *SIAM J. Matrix Anal. Appl.*, 22:793–818, 2000.

[18] L.F. Pavarino and O.B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55:302–335, 2002.

[19] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications, Oxford, 1999.

[20] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition.* Cambridge University Press, Cambridge, 1996.

[21] A. Toselli and W. Widlund. *Domain Decomposition Methods.* Computational Mathematics, Vol. 34. Springer, Berlin, 2005.

[22] A. van der Sluis and H. A. van der Vorst. The rate of convergence of Conjugate Gradients. *Numer. Math.*, 48:543–560, 1986.

[23] 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. Comput. Phys.*, 152:385–403, 1999.

[24] C. Vuik, A. Segal, L. El Yaakoubi, and E. Dufour. A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients. *Applied Numerical Mathematics*, 41:219–233, 2002.