ELSEVIER

Applied Numerical Mathematics 41 (2002) 219–233

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients

C. Vuik [a,*], A. Segal [a], L. el Yaakoubi [a,b], E. Dufour [b]

[a] *Delft University of Technology, Faculty of Information Technology and Systems,*
Department of Applied Mathematical Analysis, Mekelweg 4, 2628 CD Delft, The Netherlands
[b] *Shell International Exploration and Production, P.O. Box 60, 2280 AB Rijswijk, The Netherlands*

Dedicated to the memory of Rüdiger Weiss

Abstract

A mathematical model to predict excess fluid pressures in the earth's crust leads to a time-dependent diffusion equation for the pressure. Application of the finite element method to this equation results in a large system of linear equations. Due to the layered structure of the underground the permeability used in the diffusion equation has large jumps, so the coefficient matrix has a large condition number of order 10^8. This leads to bad convergence of the ICCG method and a wrong termination criterion. Combining ICCG with a deflation technique leads to a robust solution method. A difficulty is the construction of the deflation vectors. In this paper we present three different choices of the deflation vectors and compare them from a theoretical point of view and from numerical experiments. This comparison shows that the best deflation technique is based on algebraic deflation vectors. © 2001 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Deflation; IC preconditioned Conjugate Gradients; Poisson equation; Porous media; Discontinuous coefficients

1. Introduction

One of the issues an oil company is interested in is the fluid pressure history and the temperature history in the earth's surface. Knowledge of these quantities can be of help in predicting the presence of oil and gas in reservoirs. Moreover, the fluid pressure plays an important role in safety and environmental aspects of drilling a well.

Mathematical models to predict fluid pressures in porous media in a geological time scale are all based on Darcy's law [1] and the continuity equation.

[*] Corresponding author.
E-mail address: c.vuik@math.tudelft.nl (C. Vuik).

PII: S0168-9274(01)00118-0

The equations itself are three dimensional, time-dependent, nonlinear diffusion equations. One of the characteristics of these equations is the dependence of the diffusion parameter on the permeability. This permeability largely depends on the type of material and may vary orders of magnitude. For example the ratio in permeability between sandstone and shale may be of order 10^8.

To solve the diffusion equation we discretize in space by a finite element method. The time dependence is solved by applying some implicit scheme in combination with a suitable linearization. This all results in a large sparse system of equations to be solved. Due to the three dimensional nature of the problem, it is necessary to solve these equations by some iterative method.

Unfortunately the large jumps in permeabilities result in very ill conditioned matrices, and as a consequence a very slow convergence of classical iteration methods like conjugate gradients (CG). Another serious problem is that even if the reduction of the residual of the equation is large, still it may be possible that no convergence to the true solution has been achieved. This makes the choice of a suitable termination criterion a hard task.

In order to analyze this problem it is sufficient to study the following stationary linear diffusion equation:

$$-\operatorname{div}(\sigma \nabla p) = 0 \quad \text{on } \Omega, \tag{1}$$

with boundary conditions

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

where $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$. The fluid pressure and permeability are denoted by p and σ respectively. The domain Ω consists of a number of subdomains in which σ is constant. Two values for σ are considered: $\sigma^h = 1$ for high-permeability subdomains and $\sigma^l = \varepsilon$ for low-permeability subdomains (e.g., the permeabilities ratio for shale and sandstone: ε is of the order 10^{-7} see [8]). The subdomains are denoted by the disjoint sets Ω_i, $i \in \{1, \ldots, k\}$, which are such that: $\bigcup_{i=1}^k \overline{\Omega}_i = \overline{\Omega}$ and when $\overline{\Omega}_i \cap \overline{\Omega}_j \neq \emptyset$ then $\sigma_i \neq \sigma_j$. Note that in real life applications, the permeability σ is also varying in the subdomains.

In [8] this problem has been analyzed thoroughly. It is shown that the number of small eigenvalues of the large matrix is of the order of the number of nodes in the domains with low-permeability. However, using an Incomplete Cholesky (IC) preconditioner reduces this number considerably to the number of high-permeability regions, completely enclosed by low-permeability regions or boundaries with zero flux boundary conditions.

Based on this observation it is not difficult to create the eigenvectors corresponding to the small eigenvalues in the case of straight layers. It is then a logical choice to improve the condition of the matrix considerably by projecting the solution onto the space perpendicular to the space spanned by these eigenvectors. Effectively this means that all small eigenvalues will be replaced by 0. Since the CG method is well suited to solve singular problems, provided the right-hand side vector is in the correct space, effectively the condition of the matrix with respect to the CG method is considerably improved.

In [8] a method based on this projection, the Deflated ICCG method (DICCG), has been derived. For more general problems, however, the construction of the eigenvectors corresponding to the small eigenvalues is not so simple. Therefore an approximation of these “small” eigenvectors has been used to apply the DICCG method. These approximate eigenvectors are equal to 0 or 1 in regions with large permeabilities and more or less linear varying in regions with small permeabilities. Application of the DICCG method results in very fast convergence and also ensures that standard termination criteria can be applied without possible dangers.

In a later work [9] we have shown that even if the projection vectors are rather inaccurate in the small permeability region still the DICCG method behaves almost as good as in the case of “exact” projection vectors. This resulted in constructing the projection vectors by solving the original equation in each small permeability region separately with a very modest accuracy. This approach has already been successfully applied to other problems with a comparable jump in the coefficients [2].

However, still this method is quite complicated. In the present paper we try to further simplify the construction of the projection vectors. In first instance this is done by merely making all projection vectors equal to zero in the small permeability region. Hence the projection vectors consist of only numbers 0 and 1. Experiments show that this method sometimes introduces other small eigenvalues, and therefore is not suitable to get a well conditioned matrix.

In order to improve this behavior the set of projection vectors is extended in such a way that for each subdomain we have a separate projection vector. This vector is 1 in one subdomain and zero in all other subdomains. Of course the use of extra projection vectors makes the projection more expensive, but the simplicity of the projection vectors and the large number of zeros in it, make the projection relatively cheap.

In this paper the two new approaches are compared with the old one using approximate eigenvectors. It is shown that the first new approach leads to good results in some cases but is not robust. The second method with the larger number of projection vectors, appears to be very robust. Compared to the method of approximate eigenvectors, an important advantage of the new approach is that it can be applied, even if we have difficulty in deciding which part of the region is high permeable and which one is low permeable. This is especially the case if coefficients are varying in a somewhat continuous way.

Summary of the paper. In Section 2 the DICCG method and some of its properties are summarized. The various deflation strategies are specified in Section 3. A theoretical comparison is given in Section 4.1, together with a numerical comparison presented in Section 4.2. We end with some conclusions in Section 5.

2. The Deflated ICCG method

After a finite element discretization of (1) the linear system

$$Ax = b$$

with $A \in \mathbb{R}^{n\times n}$ has to be solved. The coefficient matrix A is large, sparse, symmetric, and positive definite. These properties motivates us to use the Conjugate Gradient method combined with an Incomplete Cholesky decomposition (ICCG). The Incomplete Cholesky factor is denoted by L. In [9], Theorem 2.2 it is shown that the spectrum of the IC preconditioned matrix $L^{-1}AL^{-\mathrm{T}}$ contains a small number of eigenvalues of order $\varepsilon = \sigma^l/\sigma^h$, whereas the largest eigenvalue is of order one. This leads to slow convergence and a termination criterion which is not reliable [8,9].

An error bound after k iterations of the ICCG method is given by [6, p. 187]:

$$\|x - x_k\|_{L^{-1}AL^{-\mathrm{T}}} \leqslant 2\|x - x_0\|_{L^{-1}AL^{-\mathrm{T}}}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \tag{2}$$

where $\kappa = \lambda_n/\lambda_1$ is the spectral condition number of $L^{-1}AL^{-\mathrm{T}}$ and $\|x\|_{L^{-1}AL^{-\mathrm{T}}} = (x^{\mathrm{T}}L^{-1}AL^{-\mathrm{T}}x)^{1/2}$. This bound is also slowly decreasing for increasing k when $\kappa \gg 1$.

From theory [3] and numerical experiments [8,9] it appears that Deflated ICCG can be used to speed up the convergence considerably and to obtain a reliable termination criterion. Below we summarize the DICCG method for ease of reference.

Let us define the projection P by

$$P = I - AZ\left(Z^{\mathrm{T}}AZ\right)^{-1}Z^{\mathrm{T}}, \quad Z \in \mathbb{R}^{n\times m},$$

where $Z = [z_1 \cdots z_m]$ has rank m. The space span$\{z_1, \ldots, z_m\}$ is denoted as the deflation subspace. Since $x = (I - P^{\mathrm{T}})x + P^{\mathrm{T}}x$ and because

$$\left(I - P^{\mathrm{T}}\right)x = Z\left(Z^{\mathrm{T}}AZ\right)^{-1}Z^{\mathrm{T}}Ax = Z\left(Z^{\mathrm{T}}AZ\right)^{-1}Z^{\mathrm{T}}b,$$

can easily be computed, only $P^{\mathrm{T}}x$ has to be computed by iterations. Using the identity $AP^{\mathrm{T}} = PA$ we can solve the deflated system

$$PA\tilde{x} = Pb, \tag{3}$$

for $\tilde{x}$ using ICCG and premultiply $\tilde{x}$ by P^{T} to obtain $P^{\mathrm{T}}x$. The algorithm to solve (3) is [8]:

DICCG

```
$j = 0$, $\hat{r}_0 = Pr_0$, $p_1 = z_1 = L^{-\mathrm{T}}L^{-1}\hat{r}_0$;
  while $\|\hat{r}_j\|_2 >$ accuracy do
    $j = j + 1$; $\alpha_j = \frac{(\hat{r}_{j-1}, z_{j-1})}{(p_j, PAp_j)}$;
    $x_j = x_{j-1} + \alpha_j p_j$;
    $\hat{r}_j = \hat{r}_{j-1} - \alpha_j PAp_j$;
    $z_j = L^{-\mathrm{T}}L^{-1}\hat{r}_j$; $\beta_j = \frac{(\hat{r}_j, z_j)}{(\hat{r}_{j-1}, z_{j-1})}$;
    $p_{j+1} = z_j + \beta_j p_j$;
  end while
```

When we choose $z_1, \ldots, z_m$ equal to the eigenvectors corresponding to the smallest eigenvalues it follows that the convergence of DICCG depends on the effective condition number

$$\kappa_{\mathit{eff}} = \frac{\lambda_n}{\lambda_{m+1}}.$$

A drawback of this approach is that it costs much work to compute these eigenvectors. Therefore an approximation is proposed in the literature [8,3,9], which is cheap to compute and leads to a comparable convergence behavior. In the following sections we present two other choices for the deflation vectors and compare them with respect to applicability, amount of work, and memory requirements.

3. Choice of deflation vectors

In this section three different choices for the deflation vectors are defined:

- deflation vectors, based on the system of equations,
- algebraic deflation vectors restricted to high-permeability domains, and
- algebraic deflation vectors.

To construct the deflation vectors the subdomains are ordered as follows:

Definition 1. The high-permeability subdomains are numbered first: $\Omega_i,\ i \in \{1,\dots,k^h\}$. Furthermore the first k^s high-permeability subdomains are such that $\overline{\Omega}_i \cap \partial\Omega^D = \emptyset,\ i \in \{1,\dots,k^s\}$.

In words: the first k^s subdomains are highly permeable and their boundaries do not contain a part of the Dirichlet boundary.

For all deflation methods the permeability may be variable. It can be a function of space and we may even replace the permeability by a full Cartesian permeability tensor that is a function of space.

3.1. Deflation vectors based on the system of equations

The minimum number of deflation vectors is equal to k^s. The deflation vectors based on the system of equations are defined as [9]:

Definition 2. The vectors z_i for $i \in \{1,\dots,k^s\}$ are such that:

- $z_i = 1$ on $\overline{\Omega}_i$ and $z_i = 0$ on $\overline{\Omega}_j,\ j \neq i,\ j \in \{1,\dots,k^h\}$,
- z_i satisfies the finite element discretization of the equation:
$$-\operatorname{div}(\sigma_j \nabla z_i) = 0 \quad \text{on } \Omega_j,\ j \in \left\{k^h+1,\dots,k\right\}, \tag{4}$$
where Dirichlet boundary conditions are used at the interfaces $(\partial\Omega_j \cap \Omega)$ and homogeneous Dirichlet and Neumann boundary conditions are used at outer boundaries ($z_i = 0$ on $\overline{\Omega}_j \cap \partial\Omega^D$ and $\partial z_i/\partial n = 0$ on $\overline{\Omega}_j \cap \partial\Omega^N$).

Note that the deflation vectors are independent of the permeabilities ratio ε. In the low-permeability domains the vectors z_i can also be interpreted as a solution of (1) with Dirichlet boundary conditions equal to 1 for $\partial\Omega_i$ and equal to 0 for $\partial\Omega_j,\ j \neq i,\ j \in \{1,\dots,k^h\}$.

In the remainder of this paper the matrix Z of this choice is denoted by Z_{phys}. From theory and numerical experiments [9] it appears that the convergence behavior of DICCG using the physical deflation vectors is independent of ε.

It appears that the DICCG method is not sensitive to large perturbations of the deflation vectors in the low-permeability domains. To explain this we summarize the perturbation results given in [9]. For a typical example the smallest eigenvalue of the preconditioned matrix is of the order 10^{-9}, whereas after deflation the smallest eigenvalue is equal to 10^{-1}. When the deflation vectors are changed with an order 1 perturbation in the low-permeability domains the smallest eigenvalue after deflation becomes 10^{-2}. This suggests that taking the deflation vectors as zero in the low-permeability domains leads to a suitable deflation operator. This choice is defined in the following section.

3.2. Algebraic deflation vectors restricted to high-permeability domains

The algebraic deflation vectors, which are only nonzero on high-permeability domains, are defined as:

Definition 3. The vectors z_i for $i \in \{1,\dots,k^s\}$ are such that: $z_i = 1$ on $\overline{\Omega}_i$ and $z_i = 0$ on $\Omega_j,\ j \neq i,\ j \in \{1,\dots,k\}$.

The matrix Z resulting from this choice is denoted as Z_{alg_min}.

3.3. Algebraic deflation vectors

In [3] deflation is used to accelerate a domain decomposition algorithm. The number of algebraic deflation vectors is equal to k. The vectors are defined as:

Definition 4. The vectors z_i for $i \in \{1, \ldots, k\}$ are such that: $z_i = 1$ on $\overline{\Omega_i}$ and $z_i = 0$ on Ω_j, $j \neq i$, $j \in \{1, \ldots, k\}$.

The matrix Z resulting from this choice is denoted as Z_{alg}. For simple problems discretized with finite volumes it appears that the convergence behavior of DICCG is also independent of ε [3]. In this paper this is confirmed for complicated problems using a finite element discretization.

In the following section we compare the various choices for the deflation vectors.

4. Comparison of the various choices

In this section we compare the choices for the deflation vectors as specified in Section 3. In Section 4.1 we give a theoretical comparison, and in Section 4.2 the deflation strategies are used to solve a finite element discretization of a number of Poisson problems.

4.1. A theoretical comparison

A comparison is given of the *phys*, *alg_min* and *alg* choices of the deflation vectors with respect to: construction, convergence, required memory and amount of work. At the end of this section we summarize our results.

Construction. The construction of Z_{alg_min} and Z_{alg} is straightforward, whereas for the construction of Z_{phys} a number of small subproblems has to be solved. An advantage of the latter choice is that the space spanned by $\{z_1, \ldots, z_{k_s}\}$ is close to the eigenspace corresponding to the small eigenvalues. This relation is not valid for the other deflation strategies.

An advantage of Z_{alg} is that all subdomains are treated in the same way. This is in contrast with both other choices, where the high- and low-permeability domains play a different role. This is a drawback for the *phys* and *alg_min* approach, because in practical applications with a variable permeability in the subdomains it is not always possible to distinguish the high- and low-permeability domains. The same holds if there are more permeability constants $\sigma_1, \sigma_2, \sigma_3, \ldots,$ instead of σ^l and σ^h.

Convergence. It is hard to give theoretical bounds on the convergence speed. Therefore we include a number of typical test problems in the next section to compare the convergence numerically.

For the choice Z_{phys} it has been proven in [9] that the span of the deflation vectors is close to the eigenspace corresponding to the small eigenvalues. Therefore one expects that the convergence is related to $\kappa_{eff} = \lambda_n / \lambda_{k_s}$, which is confirmed by numerical experiments.

In [3] it is shown that the convergence behavior of the deflated ICCG method with $\{z_1, \ldots, z_k\}_{alg}$ as deflation vectors is independent of the jumps in the permeability.

For the choice Z_{alg_min} the convergence is strongly related to the perturbation behavior of Z_{phys}. This is a point of current research.

Parallelization. All deflation strategies can be combined with parallel computing. However, since the deflation vectors of the *alg_min* and *alg* choice are restricted to one subdomain these choices are somewhat easier to parallelize than the *phys* deflation vectors. For a parallel implementation of Z_{alg} we refer to [7,3].

Work and memory. From the definitions given in Section 3 it appears that the deflation vectors are sparse, because they are zero everywhere, except on $\overline{\Omega}_i$. For the choice *phys* the vectors are also nonzero on its neighboring subdomains. From the definition of P:

$$P = I - AZE^{-1}Z^{\mathrm{T}}, \quad \text{with } E = Z^{\mathrm{T}}AZ, \tag{5}$$

it follows that the determination of Pv requires a number of inner products and vector updates with z_i and Az_i. Note that the vector Az_i is less sparse than z_i. Some fill-in occurs at the grid points connected to the domain where z_i is nonzero. In the following we define N_i as the number of nonzero elements of the vector Az_i, for $1 \leqslant i \leqslant m$. The values of N_i and m are different for the various deflation strategies. To illustrate this we give a typical example at the end of this section.

From (5) it follows that the vectors z_i and Az_i should be stored in memory This requires $2\sum_{i=1}^{m} N_i$ memory positions. Matrix element E_{ij} is only nonzero if the union of the support of z_i and Az_j is not empty, so the matrix E is sparse. Therefore we do not use E^{-1} but compute the Cholesky factor of E. Since $m \ll n$ we neglect the amount of memory to store E and its Cholesky factor.

With respect to the amount of work we distinguish two parts: the construction of the deflation vectors, computing Az_i and the Cholesky factor of E, and the application of P to a vector. The first part is done only once, whereas the second part is used in every iteration. To quantify the amount of work we use A_{row} which is equal to the average number of nonzero elements in the rows of A.

The amount of work for the construction of Z_{alg_min} and Z_{alg} is negligible. To construct Z_{phys} one has to solve a number of subproblems. In general it is hard to quantify this amount of work, which is denoted by $Cons_{phys}$. Furthermore an efficient implementation of this part can be complex. The evaluation of Az_i costs $2A_{row}\sum_{i=1}^{m} N_i$ flops. For the computation of E approximately $E_{row}\sum_{i=1}^{m} N_i$ flops are required, where E_{row} is equal to the average number of nonzero elements in the rows of E. The amount of work to factorize E is negligible.

In every iteration the deflation operator is used. The calculation of Pv is done in three steps. In the first step $w = Z^{\mathrm{T}}v$ is calculated, which requires $2\sum_{i=1}^{m} N_i$ flops in inner products. The amount of flops of the second step $s = E^{-1}w$ is insignificant. In the final step another $2\sum_{i=1}^{m} N_i$ flops are needed to form

$$Pv = \left(I - AZE^{-1}Z^{\mathrm{T}}\right)v = v - (AZ)s.$$

Sparseness. Before we illustrate these estimates with an example we note that there is another drawback of the *phys* deflation method. One can have applications where $N_i \approx n$ for the *phys* deflation vectors, which makes DICCG unattractive. An example of this: assume that the domain Ω consists of a low-permeability subdomain which contains k^s subdomains with a high permeability. In [9] it is shown that using a drop tolerance δ and ignoring all elements of the projection vectors less than δ makes DICCG feasible again.

Example. We consider the following characteristic configuration: Ω is a rectangular domain, which consists of $2k^s$ plain layers of equal thickness with alternating low- and high-permeability layers (see Fig. 1). We assume that every layer contains the same number of grid points: $n/(2k^s)$. Furthermore we

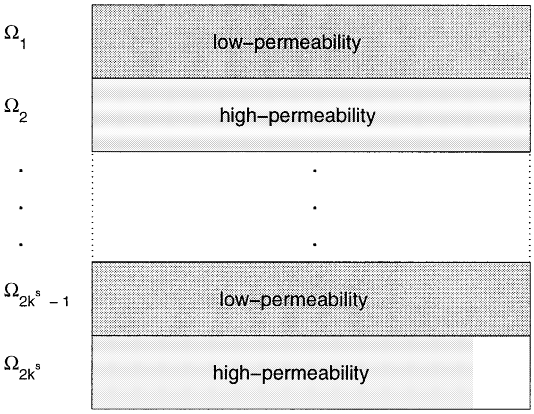

Fig. 1. A problem with $2k^s$ plain layers of equal thickness.

Table 1
The value of m, N_i, and the required memory for the various deflation strategies

Choice	m	N_i	Required memory
phys	k^s	$\frac{3n}{2k^s}$	$3n$
alg_min	k^s	$\frac{n}{2k^s}$	n
alg	$2k^s$	$\frac{n}{2k^s}$	$2n$

Table 2
Work to construct operator P

Choice	Work	Work with $A_{row} = 9$, $E_{row} = 3$
phys	$Cons_{phys} + 3n(A_{row} + \frac{1}{2}E_{row})$	$Cons_{phys} + 31.5n$
alg_min	$n(A_{row} + \frac{1}{2}E_{row})$	$10.5n$
alg	$2n(A_{row} + \frac{1}{2}E_{row})$	$21n$

assume $A_{row} = 9$ and $E_{row} = 3$. In Table 1 the number of deflation vectors m, the number of nonzero elements of Az_i: N_i, and the memory to store z_i and Az_i are given. It appears that the extra amount of memory is small with respect to the memory ($13n$) needed for the ICCG algorithm.

The amount of work needed to construct the projection operator P is given in Table 2. From this table we see that the work to construct P is comparable to the work to construct the IC preconditioner. Numerical experiments with Z_{phys} in [9] have shown that the construction time for large grids is small with respect to the solution time.

Table 3
Work to include deflation

Choice	*phys*	*alg_min*	*alg*
Work	$6n$	$2n$	$4n$

Finally the amount of work per ICCG iteration is equal to

$$\begin{array}{ccccccc} \text{matvec} & + & \text{prevec} & + & \text{vector updates} & = & \text{total}, \\ 18n & + & 18n & + & 10n & = & 46n. \end{array}$$

The extra amount of work per iteration to include deflation is given in Table 3. Comparing these numbers it follows that the extra amount of work to include deflation is small with respect to the costs of one ICCG iteration.

Summary. From the theoretical comparison we conclude that *alg* deflation is more flexible than *phys* and *alg_min*. The convergence of DICCG for *phys* and *alg* deflation is independent of the contrast of the permeability, so the methods are robust. The *alg_min* deflation is the cheapest method with respect to construction, memory and work per iteration.

To investigate the convergence behavior of the various methods we report numerical experiments in the next section.

4.2. A numerical comparison

In this section numerical experiments are presented using ICCG and DICCG with various deflation techniques. We compare the efficiency and the reliability of these methods starting with a simple layered problem and ending with a realistic three dimensional problem used in the oil industry.

4.2.1. A layered problem

A two dimensional Poisson equation,

$$-\operatorname{div}(\sigma \nabla p) = 0, \tag{6}$$

with p the pressure and σ the permeability is considered in a layered region. For the boundary conditions we make the following assumptions. At the earth's surface the excess pressure is prescribed. When the pressure field is required in some reservoir it is not practical to calculate the pressure in every position of the earth's crust. Therefore the domain of interest is restricted artificially. We assume that the lowest layer is bounded by an impermeable layer, so there is no flux through this boundary. The artificial vertical boundaries are taken at a sealing fault, or far away from the reservoir. Again a zero flux condition is a reasonable assumption at these boundaries. For the physical background of this problem we refer to Chapter 12 of [4].

For our model problem we assume that σ in sandstone is equal to 1 and σ in shale is equal to 10^{-7}. Furthermore the Dirichlet boundary condition at the earth's surface is set equal to 1. The solution of Eq. (6) with these boundary conditions is of course $p = 1$, but if we start with a random vector, our linear solver will not notice the difference with a real problem. It is also possible to use $p = 0$ as starting vector. For most of the methods this has only marginal effects, except for the *alg* deflation technique, where the solution is found in one or two iterations, since the initial error is in the span of the deflation vectors.

Eq. (6) is discretized by a standard finite element method. In each layer 10 elements in the horizontal and 5 elements in the vertical direction are used. This results in a system of linear equations to be solved, which will be denoted by $Ax = b$. In our first experiment we have solved this problem on a rectangular domain with 7 straight layers (Fig. 2), using ICCG and DICCG with the deflation strategies as introduced in Section 3.

Fig. 3 shows the norm of the true error. Note that all the versions of DICCG lead to a faster convergence than ICCG. We have not included CPU time measurements, since the *alg_min* and *alg* deflation methods are not yet implemented in an optimized way. In [9] a comparison of CPU times of ICCG and DICCG

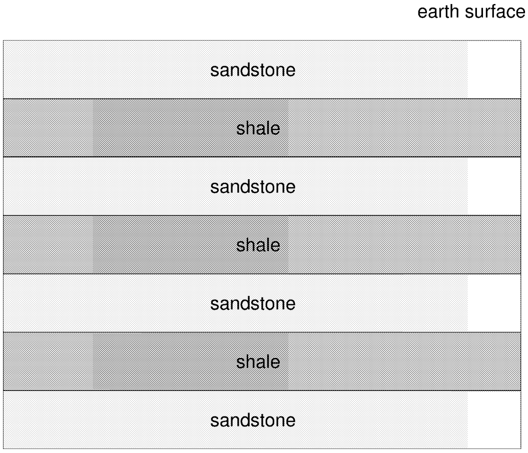

Fig. 2. Artificial configuration with 7 straight layers.

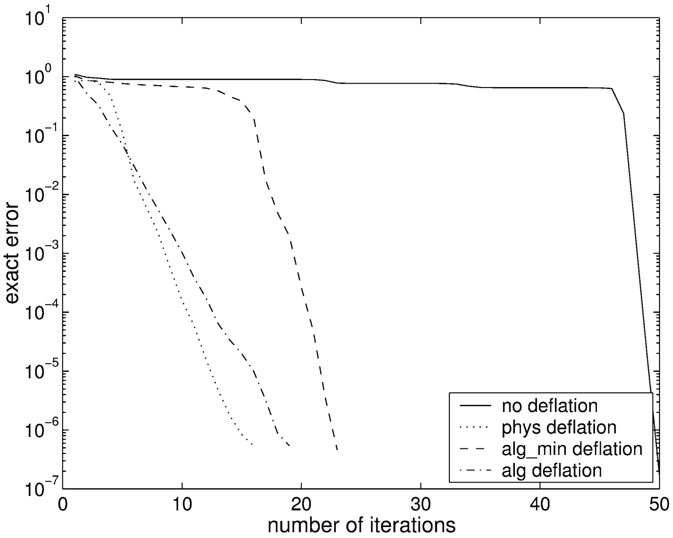

Fig. 3. The true error for a configuration with 7 straight layers.

(*phys*) has been given, which shows that DICCG is much faster than ICCG. Taking into account the results given in Table 3 we expect that the same holds for the other deflation vectors. The convergence behavior of *phys* and *alg* is more or less the same, whereas the convergence with the *alg_min* vectors is slower.

In Table 4 we also give the smallest nonzero eigenvalues of the deflated, preconditioned matrix. Since the effective condition number is a good indication of the initial convergence rate (compare Eq. (2)), these results correspond well with the convergence results as shown in Fig. 3.

4.2.2. *A skewed layered problem with variable permeabilities*

In this section we solve Eq. (6) on a geometry with 7 skewed layers. Furthermore the permeability is variable and depends on the depth. The configuration is given in Fig. 4, whereas an indication of the permeability is given in Fig. 5. Note that it is arbitrary for some of the layers to define them as shale or sandstone since the permeability varies orders of magnitude in some layers.

We use as stopping criterion:

$$\|M^{-1}r_k\|_2 \leqslant \lambda_{min}\|x_k\|_2\varepsilon, \tag{7}$$

where $M^{-1} = L^{\mathrm{T}}L$ is the Incomplete Cholesky decomposition and λ_{min} is estimated from the (D)ICCG parameters [5,8]. If the estimate of λ_{min} is accurate it is easy to show that if (7) holds the true error satisfies the inequality

$$\frac{\|x - x_k\|_2}{\|x_k\|_2} \leqslant \varepsilon.$$

Table 4
The smallest nonzero eigenvalue of the deflated, preconditioned matrix

Choice	ICCG	*phys*	*alg_min*	*alg*
λ_{min}	9.3×10^{-7}	1.6×10^{-1}	1.5×10^{-2}	3.8×10^{-1}

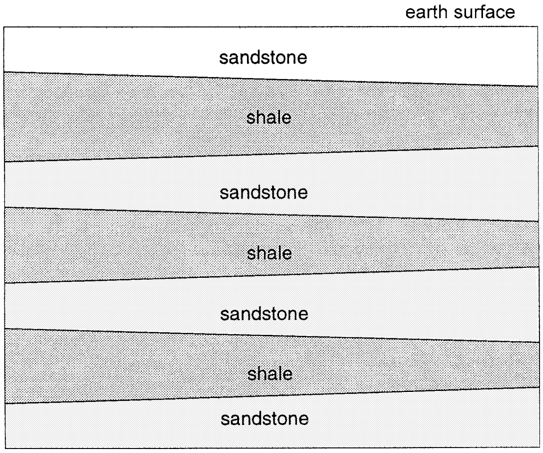

Fig. 4. The geometry of the skewed layers.

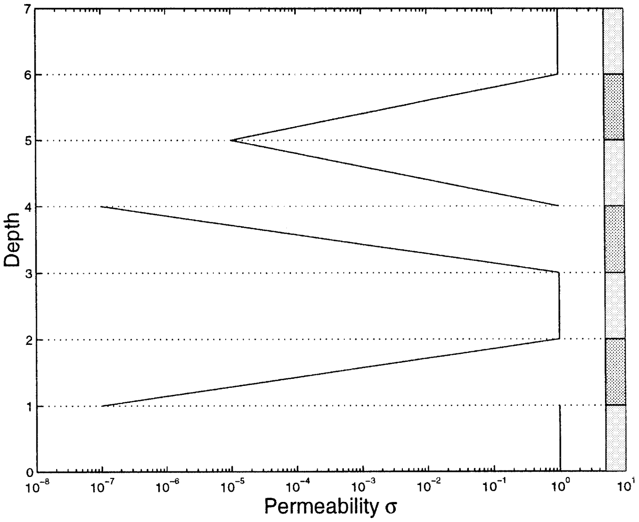

Fig. 5. The permeability.

Table 5
The number of iterations and in brackets the true relative error

Choice	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$
ICCG	171 (2.8×10^{-9})	183 (2.2×10^{-9})
phys	66 (1.2×10^{-3})	81 (1.5×10^{-7})
alg_min	116 (9.4×10^{-9})	118 (9.5×10^{-9})
alg	57 (1.1×10^{-5})	74 (2.0×10^{-7})

Using (7) as termination criterion we present the number of iterations and the true relative error in Table 5. From this table it appears that the termination criterion is reasonable for the *phys* and *alg* deflation, but it is not sharp for ICCG and the *alg_min* deflation. For this problem the *alg* deflation method leads to the smallest number of iterations.

A closer look at the estimated and true error shows that the termination criterion is not reliable for ICCG [8] and the *alg_min* deflation (see Fig. 6). Finally the smallest eigenvalue of $PM^{-1}A$ with *alg_min* deflation is as small as that of $M^{-1}A$. So for this problem the *alg_min* strategy has the same bad properties as ICCG.

4.2.3. A problem with many high-permeability inclusions

We consider a 3 layer problem, where the shale layer contains 8 sandstone inclusions (see Fig. 7). The resulting linear system is again solved with the ICCG and the DICCG iteration methods. In Fig. 8 the norm of the true error is given. The convergence behavior of the iterative methods is comparable to that observed in Section 4.2.1 for a straight layer problem. In the initial iterates the norm of the true error for the *alg* method is considerably smaller than that of the other deflation methods. For this problem the *alg* method is clearly the best method.

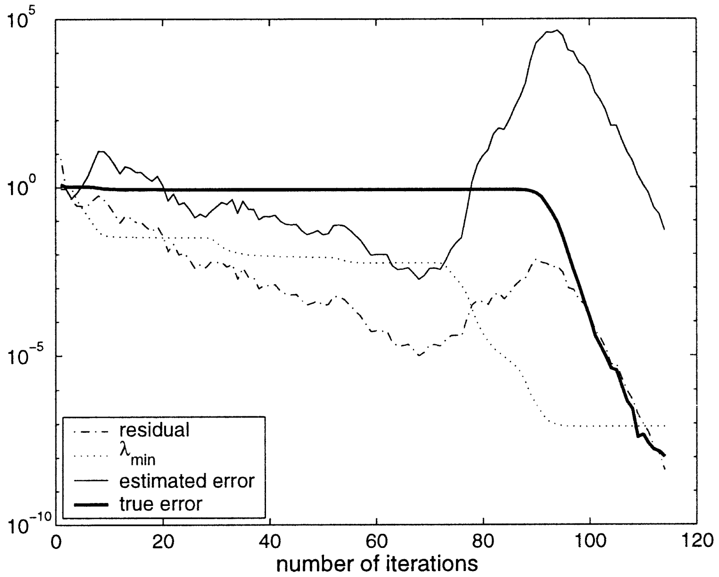

Fig. 6. The norm of the true and estimated error using the *alg_min* vectors.

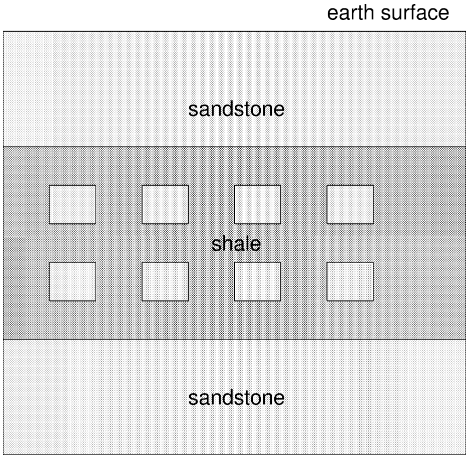

Fig. 7. A problem with 8 sandstone inclusions in the shale layer.

4.2.4. An oil flow problem

Finally we apply the iterative methods to a more realistic three dimensional problem which is used to simulate the flow of oil and gas in a reservoir. The geometry and the permeabilities are given in Fig. 9.

The same termination criterion is used as in Section 4.2.2. It appears that the termination criterion is reliable for all deflation strategies, but not for ICCG. The results given in Table 6 show that the number of iterations is the same for ICCG and *alg_min*, and *phys* and *alg*. The latter methods are clearly more efficient than the first ones, especially for a practical accuracy ($\varepsilon = 10^{-2}$).

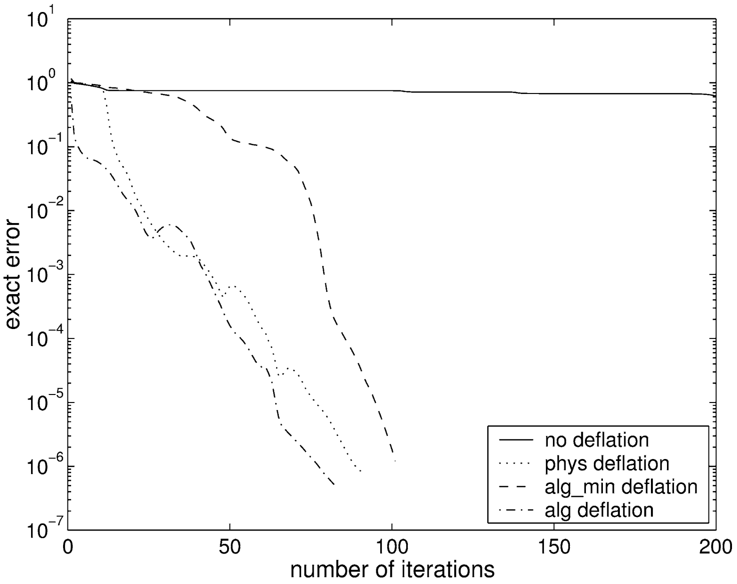

Fig. 8. The error for a problem with 8 sandstone inclusions in the shale layer.

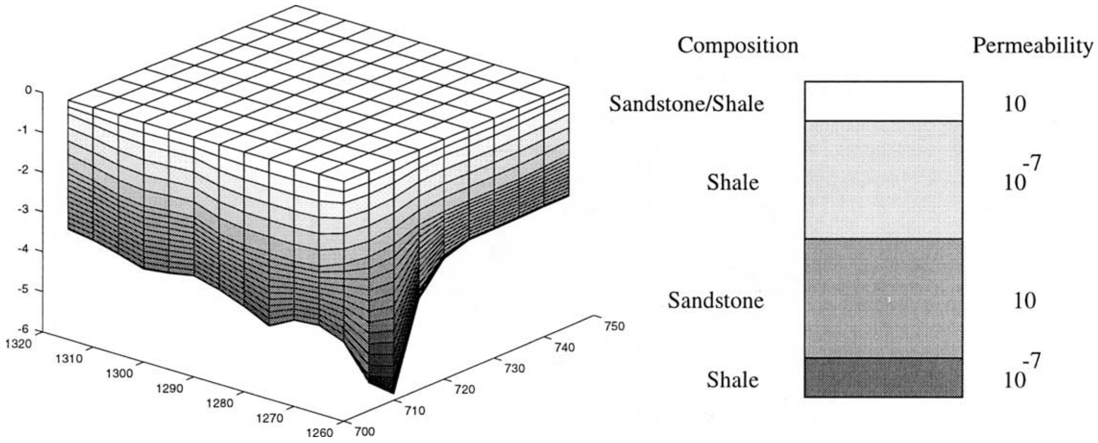

Fig. 9. The geometry of the oil flow problem.

Table 6
The number of iterations and in brackets the true relative error

Choice	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$
ICCG	46 (1.2×10^{-6})	55 (9.8×10^{-7})
phys	15 (1.2×10^{-3})	30 (2.9×10^{-6})
alg_min	45 (5.2×10^{-3})	55 (2.7×10^{-6})
alg	15 (1.4×10^{-3})	30 (5.5×10^{-6})

5. Conclusions

For elliptic problems with large jumps in the coefficients the ICCG method is not robust. Combining ICCG with a deflation technique can give an efficient and robust solver depending on the choice of the deflation vectors. In this paper we consider the *phys*, *alg_min* and *alg* deflation strategies. From the theoretical and numerical comparisons we made the following conclusions:

- The Deflated ICCG method with deflation vectors based on the system of equations (*phys*) is robust for all considered examples, also for problems which do not satisfy the assumption that high-permeability domains are contained in low-permeability domains.
- The *alg_min* strategy leads for some problems to good results, however for other problems the method fails. So this method is not robust.
- The *alg* deflation method is robust and in most of our applications it is the most efficient method.

Our final conclusion is that DICCG with *alg* deflation vectors is the method of choice for elliptic problems with discontinuous coefficients.

References

[1] J. Bear, Dynamics of Fluids in Porous Media, Elsevier, New York, 1972.

[2] H. De Gersem, K. Hameyer, A deflated iterative solver for magnetostatic finite element models with large differences in permeability, Eur. Phys. J. Appl. Phys. 13 (2000) 45–49.

[3] J. Frank, C. Vuik, On the construction of deflation-based preconditioners, SIAM J. Sci. Comput. 23 (2001) 442–462.

[4] M.R. Giles, Diagenesis: A Quantitative Perspective; Implications for Basin Modelling and Rock Property Prediction, Kluwer, Dordrecht, 1997.

[5] E.F. Kaasschieter, A practical termination criterion for the conjugate gradient method, BIT 28 (1988) 308–322.

[6] D.G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison Wesley, New York, 1973.

[7] C. Vuik, J. Frank, A parallel block preconditioner accelerated by coarse grid correction, in: M. Bubak, H. Afsarmanesh, R. Williams, B. Hertzberger (Eds.), High-Performance Computing and Networking, Proceeding of the 8th International Conference, HPCN Europe 2000, Amsterdam, May 8–10, 2000, Lecture Notes in Computer Science, Vol. 1823, Springer, Berlin, 2000, pp. 99–108.

[8] C. Vuik, A. Segal, 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 (1999) 385–403.

[9] C. Vuik, A. Segal, J.A. Meijerink, G.T. Wijma, The construction of projection vectors for a Deflated ICCG method applied to problems with extreme contrasts in the coefficients, J. Comput. Phys. 172 (2001) 426–450.